Investigation of Supersonic Impulse Turbines for Application to Geothermal Binary Power Stations

Jason Paul Czapla

Bachelor of Science, Mechanical Engineering

A thesis submitted for the degree of Doctor of Philosophy at

The University of Queensland in 2015

School of Mechanical and Mining Engineering
Abstract

This thesis was conducted within the University of Queensland’s Geothermal Energy Centre of Excellence (QGECE). The centre had four major challenges:

1. Optimum energy extraction and sustainable resource management.
2. Efficient power conversion.
3. A cooling system for a desert zone in the world’s driest inhabited continent.
4. To resolve transmission issues inherent to a power plant which is located more than 500km from major load centres and the national grid.

This thesis is related to the second major challenge of improving power conversion efficiency. That challenge had an associated milestone to develop working laboratories for testing power conversion systems. This thesis aligns itself with the goals of the centre to develop working facilities and to investigate opportunities to improve power conversion efficiency.

The synergies of the centres goals and this thesis’s objectives are based on binary power plant technology for use with geothermal applications. This thesis postulates the question; what is the impact of incorporating a real turbine loss model into a binary cycle analysis?

A binary power plant test facility was designed and built to test turbines operating in organic Rankine cycles. For the operating conditions of the power plant test facility, a single-stage, supersonic, axial impulse turbine was designed, built and tested across a range of conditions and experimental performance data was collected for analysis.

The gathered data was used to calibrate a computer program written to calculate losses in the stator and rotor passages of a single stage axial impulse turbine. The calibrated loss model was then incorporated into another program written to calculate the performance of organic Rankine cycles. The incorporated loss model into the cycle analysis program allowed for calculating organic Rankine cycle performance based on calculated turbine efficiency.

Cycle analyses conducted over a range of pressures, working fluids, and temperatures showed a clear trend that each working fluid had a unique optimum evaporator pressure for each geothermal source temperature. High pressure supercritical cycles were shown to have good cycle performance as they tend to have a good thermal profile match between the thermal fluid and the working fluid in the evaporator, thus maximising the utilisation of the energy in the thermal fluid. However, in certain conditions, the implementation of a recuperator may achieve similar if not better
performance than supercritical cycles but at much lower pressures. This is achieved by initiating evaporation of the working fluid before the recuperator outlet.

The calibrated loss model showed that the losses in the single stage impulse turbine were dominated by the windage and the partial admission sector losses. At low admission rates the partial admission pumping losses became a dominant source of losses. Also at low admission rates the other losses (clearance, incidence, trailing edge and passage) all become a larger part of the overall losses on a percentage basis. This leads to low admission turbines having relatively low efficiencies and being more sensitive to operating conditions. Smaller power systems will generally have lower mass flow rates and will lead to lower admission turbines. The influence of an incorporated turbine loss model was more pronounced for lower power systems.

For a wide range of conditions, performance maps of optimum operating conditions were generated along with preliminary designs of single-stage, axial, impulse turbines. The incorporated loss model provided insight into a holistic design approach that optimises cycle and turbine performance concurrently.

The major achievements of this thesis are the successful design and build of an ORC test facility and a single stage axial impulse turbine. This test facility provides a new test platform for the University of Queensland’s ongoing research into power conversion systems. Another major achievement was the completion of a comprehensive computer code that may be used to analyse organic Rankine cycles, calculate turbine performance and produce geometry for stator and rotor passages for an impulse turbine. The last major achievement was to conduct experimentation on a single stage axial impulse turbine and use the experimental data to calibrate a loss model that could be incorporated into cycle analyses.
Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the General Award Rules of The University of Queensland, immediately made available for research and study in accordance with the Copyright Act 1968.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.
**Publications during candidature**

No publications

**Publications included in this thesis**

No publications included
Contributions by others to the thesis

The ORC test rig that was developed as part of this thesis was a joint effort by the members of the QGECE power conversion team. The data acquisition systems and software were developed by Andrew Rowlands and Rajinesh Singh. Hugh Russell played a major role in the design and build of the ORC test loop. Hal Gurgenci created the foundation for which the ORCCA code was written. Carlos Ventura contributed code to the AXIAL program that helped to calculate passage losses.

Statement of parts of the thesis submitted to qualify for the award of another degree

None
Acknowledgements

I thank the Queensland State Government for the funding that made this research possible. I would also like to thank the University of Queensland School of Engineering for the scholarship that has permitted me to carry out this thesis.

I would like to give special thanks to Halim Gurgenci for giving me the opportunity to be part of QGECE and allowing me the chance to pursue this research. I thank Peter Jacobs for being one of my supervisors and helping me to navigate the difficulties and problems encountered in the course of this research. I thank Paul Petrie-Repar, another one of my supervisors, for his experienced and knowledgeable advice on the topic of CFD and turbomachinery and Andrew Rowlands for his guidance in the PhD thesis process and helping me to maintain focus.

I thank Glenda Heyde for somehow making everything in the university seem to run a little smoother. I don’t think I would have been able to do nearly as much as I did without Glenda’s help to get items ordered and delivered, and just in general, always helping to point me in the right direction.

I thank Rajinesh Singh, Hugh Russell and Carlos Ventura. Building and commissioning the lab facilities was a group effort and I feel fortunate to have been able to work in such a talented group of engineers.

I also want to thank Examiners A and B wherever and whoever you are. Your questions and comments were insightful and thought provoking. They helped me to better my thesis and to draw better conclusions.

I certainly couldn’t have done any of the experimental work without the very talented and helpful mechanical engineering workshop. Particular thanks to Berto Di Pasqual who helped fabricate and install all the specialised equipment in the ORC Test Rig. I thank Gary for machining the rotor and stators of the impulse turbine on the CNC mill. The accuracy and balance of the finished parts was beyond my expectations.
Keywords

Geothermal Energy, Rankine Cycle, Binary Power Plant, Impulse Turbine, Turbomachinery, Loss Model

Australian and New Zealand Standard Research Classifications (ANZSRC)

091305 Energy Generation, Conversion and Storage Engineering 100%

Fields of Research (FoR) Classification

FoR code: 0913 Mechanical Engineering, 100%
Table of Contents

CHAPTER 1  INTRODUCTION........................................................................................................... 1

CHAPTER 2  LITERATURE REVIEW............................................................................................. 4
  2.1  Energy Supply.......................................................................................................................... 4
  2.2  Binary Power Plants................................................................................................................. 6
  2.3  Cycle Analysis.......................................................................................................................... 11
  2.4  Impulse Turbine ...................................................................................................................... 13

CHAPTER 3  THESIS OBJECTIVE ................................................................................................. 22

CHAPTER 4  THESIS METHODOLOGY ......................................................................................... 23

CHAPTER 5  CYCLE ANALYSIS ................................................................................................. 24
  Cycle Operating Conditions ........................................................................................................ 26
  Cycle Analysis Calculation Process Algorithm ......................................................................... 29
  Cycle State Points ....................................................................................................................... 30
  5.1  Cycle Calculations ............................................................................................................... 33

CHAPTER 6  APPLICATION OF CYCLE ANALYSIS ...................................................................... 43
  6.1  Cycle Analysis Results .......................................................................................................... 44
    6.1.1  Top Fluids vs. Benchmark Fluids (R245fa/Pentane) ......................................................... 45
    6.1.2  Influence of ETAT ........................................................................................................... 45
    6.1.3  Influence of Evaporator Pressure ..................................................................................... 46
    6.1.4  Influence of Recuperation .............................................................................................. 49
    6.1.5  Influence of $EHX_{\text{off}}$ and $CHX_{\text{off}}$ ........................................................................ 50
    6.1.6  Annual Considerations ................................................................................................... 52
  6.2  Operating Conditions Maps .................................................................................................. 55
CHAPTER 13  CONCLUSION................................................................................................................. 152

13.1  Future Work ................................................................................................................................. 157

REFERENCES .......................................................................................................................................... 160

Appendix A.  Cycle Analysis Code (ORCCA) ...................................................................................... 165
Appendix B.  Axial Code ......................................................................................................................... 218
Appendix C.  Small Impulse Turbine Experimental Results ................................................................. 304
Appendix D.  ORC Test Rig Operating Procedures ............................................................................. 353
Appendix E.  Original Impulse Turbine FEA Results .......................................................................... 364
List of Figures

Figure 1. Basic Layout of an Engineered Geothermal System with a Binary Power Station Using a Condensing Rankine Cycle with Recuperation ................................................................. 1
Figure 2. Temperature - Entropy diagram of a Rankine cycle with superheating ......................... 25
Figure 3. Rankine cycle component layout ................................................................................. 26
Figure 4. Heat exchanger temperature profiles (evaporator and condenser) .............................. 28
Figure 5. Overview of the cycle analysis calculation process (simplified algorithm) .................... 29
Figure 6. Heat exchanger temperature profiles (recuperator) .................................................... 32
Figure 7. State point fluid properties matrix. RP means value taken from REFPROP, G means given input for a particular cycle. .................................................................................. 33
Figure 8. Evaporator pinch point analysis diagram ................................................................... 36
Figure 9. Condenser pinch point analysis diagram .................................................................... 38
Figure 10. Evaporator (left) and condenser (right) pinch analysis process flow charts ................. 39
Figure 11. Recuperator pinch analysis process flow chart ............................................................ 40
Figure 12. Top fluids vs. benchmark fluids, R245fa and pentane .............................................. 45
Figure 13. ETAT verus $\beta$. This plot shows the relationship between ETAT and $\beta$ ............... 46
Figure 14. $\beta$ vs. evaporator pressure for RC318 for $T_h = 150^\circ$C (left) and $T_h = 170^\circ$C (right) .... 47
Figure 15. Comparison of RC318 cycles for $P_2$ of 1,000 kPa (left) and $P_2$ of 1,600 kPa (right) .... 47
Figure 16. An example of a cycle that has gas expanding through the vapour dome. The cycle is an RC318 at 6,000kPa with $T_h$ equal to 150°C and $T_c$ equal to 20°C. This figure relates to the missing optimum shown in Figure 14. .......................................................... 48
Figure 17. TS-diagram for non-recuperated (left) and recuperated (right) RC318 cycle with 1,350kPa evaporator pressure ......................................................................................... 49
Figure 18. $\beta$ versus $EHX_{dT}$ (left) and $CHX_{dT}$ (right) for RC318 at 1,000kPa ................................ 50
Figure 19. $\beta$ versus $EHX_{dT}$ (left) and $EHX_{dT}$ versus $P_2$ (right) for RC318 ............................... 51
Figure 20. $\beta$ versus $CHX_{dT}$ (left) and $CHX_{dT}$ versus $P_2$ (right) for RC318 ............................... 52
Figure 21. Annual power, energy, temperatures and optimised evaporator pressure for RC318 ..... 53
Figure 22. Annual optimum heat exchanger inlet temperature differential and recuperator for RC318 .......................................................................................................................... 54
Figure 23. Annual power, energy, temperatures and constant evaporator pressure for RC318 ..... 55
Figure 24. $n_{1st}$ plotted based on maximum $\beta$ (left) and $n_{1st}$ plotted based on maximum $n_{1st}$ (right) for pentane.................................................................56
Figure 25. $\beta$ versus $\eta_{1st}$ for $T_h = 170^\circ C$, $T_c = 20^\circ C$, $EHX_{dT} = 8^\circ C$, $CHX_{dT} = 14^\circ C$..........................57
Figure 26. Comparison a 1,000kPa cycle and a 1,900kPa cycle for Pentane .................................................................58
Figure 27. Optimum $\beta$ (left) and evaporator pressure (right) for pentane .................................................................59
Figure 28. $EHX_{dT}$ (left) and $CHX_{dT}$ (right) operating conditions map for optimum $\beta$ for pentane...59
Figure 29. $m_h$ (left) and $m_c$ (right) operating conditions map for optimum $\beta$ for pentane........60
Figure 30. Optimum usage of recuperation for RC318. Solid markers show where recuperation is advantageous.........................................................................................60
Figure 31. $\beta$ versus $P_2$ for pentane. Full plot (top left), zoomed in on $T_h = 170^\circ C$ (top right),
zoomed in $T_h = 141^\circ C$ (bottom left), zoomed in on $T_h = 120^\circ C$ (bottom right)................................61
Figure 32. Turbine Test Loop ..........................................................................................................................63
Figure 33. Photo of the QGECE ORC Test Loop..........................................................64
Figure 34. T-S diagram of average operating conditions for a R245fa cycle without recuperation
(left) and with recuperation (right) in ORC Test Loop.................................................................65
Figure 35. Example plot of rotational speed versus Time .................................................................68
Figure 36. Impulse Turbine Stator and Rotor Axial Pressure-Velocity Distribution ............70
Figure 37. Single Stage Stator-Rotor Velocity Triangle................................................72
Figure 38. 2D Stator and Rotor Passage Geometric Path Designation..........................74
Figure 39. Nozzle-blade passage geometry ..................................................................................75
Figure 40. Test impulse turbine 2D nozzle design (Top Left), Test impulse turbine work, torque,
and omega versus speed-work parameter (Top Right), Test impulse turbine velocity triangle
(Bottom Left), Test impulse turbine 2D paths (Bottom Right)..........................................................79
Figure 41. Test Impulse Turbine 3D Paths (left), Test Impulse Turbine 3D Surfaces (right)........80
Figure 42. Stator nozzle images. (Top Left) CAD rendering of stator nozzle passage piece, (Top
Right) CAD rendering of stator nozzle assembly, (Bottom Left) machined stator nozzle bottom
piece, (Bottom Right) machined stator nozzle assembly showing nozzle exit..........................83
Figure 43. Rotor images. (Left) CAD rendering of rotor, (Right) machined rotor.............83
Figure 44. Test impulse turbine assembly cut-away CAD rendering........................................84
Figure 45. Velocity patterns around rotating disks. Flow regimes I and III (left) and Flow regimes
II and IV (right) (Glassman 1994) .........................................................................................89
Figure 46. Illustration of trailing edge wake..................................................................................93
Figure 47. Rotor blade incidence nomenclature according to Roelke (Glassman 1994)........94
Figure 48. Speed and work data acquisition plot of R134A test case.................................101
Figure 49. Eilmer3 CFD results of a stator nozzle passage. (Top Left) Pressure, (Top Right) Temperature, (Bottom Left) Velocity Vector, (Bottom Right) Mach Number .........................102
Figure 50. R134a Case: Measured pressure (Top) and Eilmer3 calculated pressure (Bottom).....103
Figure 51. R134a Case: Measured Temperature (Top) and Eilmer3 Calculated Temperature (Bottom).........................................................................................................................104
Figure 52. R134a Case: Measured Mass Flow Rate (Top) and Eilmer3 Calculated Mass Flow Rate (Bottom).........................................................................................................................105
Figure 53. R134a Case: Eilmer3 Calculated Stator Exit Velocity (Top) and Mach Number (Bottom).........................................................................................................................106
Figure 54. Machine specific loss coefficient search algorithm.............................................107
Figure 55. R134a Case (1 Nozzle): Loss model coefficient search progress .......................108
Figure 56. R134a Case (1 Nozzle): Individual losses versus RPM for best fit.......................109
Figure 57. R134a Case (1 Nozzle): Measured work and calculated work versus RPM ........109
Figure 58. Gathered R134a case loss coefficient search: Measured work versus calculated work 110
Figure 59. Gathered R245fa single nozzle and dual nozzle loss coefficient search. Measured work versus calculated work .................................................................110
Figure 60. Comparison of measured work versus calculated work for no $N_{sector}$ modification (top left), $N_{sector}$ modification to sector loss only (top right) and $N_{sector}$ modification to sector and pumping losses (bottom).................................................................................................112
Figure 61. Incorporated loss model program (SSAL) flow chart .......................................117
Figure 62. Influence of Aspect Ratio ..................................................................................121
Figure 63. Losses as a percent of the turbine work for two scenarios. Aspect ratio 0.75 (left) and aspect ratio 4.00 (right). ........................................................................................................121
Figure 64. Zweifel coefficient influence on turbine efficiency ..............................................122
Figure 65. Losses as a percent of the turbine work for two scenarios. Zw of 0.75 (left) and Zw of 4.00 (right). ........................................................................................................122
Figure 66. Optimum T-S diagram of the R245fa cycle based on Table 21 ..........................124
Figure 67. Feasible turbines for 100kW R245fa cycle (work versus $\omega$). All results on the left and results for $\alpha_{II} = 80^\circ$ only on the right. ........................................................................................................124
Figure 68. Feasible turbines for 100kW R245fa cycle, work versus $\alpha_{II}$ (left) and $Z$ (right) ....125
Figure 69. Individual turbine losses for 100kW R245fa cycle of feasible turbines (left) and losses for maximum efficiency case (right).................................................................................126
Figure 70. Feasible turbines for 10kW R245fa cycle, work versus $\alpha_{II}$ (left) and $Z$ (right).......127
Figure 71. Individual turbine losses for 10kW R245fa cycle of feasible turbines (left) and losses for maximum efficiency case (right) ..................................................................................................................................................127
Figure 72. Losses for maximum efficiency 100kW case (left) and 10kW case (right) ............128
Figure 73. Feasible turbines for 2kW R245fa cycle, work versus $\alpha_{II}$ (left) and $Z$ (right)........129
Figure 74. Individual turbine losses for 2kW R245fa cycle of feasible turbines (left) and losses for maximum efficiency case (right) ..................................................................................................................................................129
Figure 75. Comparison of evaporator pressure influence of assumed efficiencies versus calculated efficiency for R245fa ..................................................................................................................................................131
Figure 76. Comparison of $EHX_{dT}$ influence of assumed efficiency versus calculated efficiency for R245fa ..................................................................................................................................................132
Figure 77. Comparison of $CHX_{dT}$ influence of assumed efficiency versus calculated efficiency for R245fa ..................................................................................................................................................133
Figure 78. $W_o$ influence of calculated efficiency for R245fa. $\beta$ versus Power (top left), $\beta$ versus $ETAT$ (top right) and $ETAT$ versus Power (bottom) ........................................................................................................................................134
Figure 79. Comparison of a 2kW (left) and a 100kW (right) turbine for R245fa at 2,500kPa .......135
Figure 80. Relationship between power and admission for a 2500kPa R245fa turbine ............136
Figure 81. Relationship between power and $U$ and $D_{r,m}$ for a 2500kPa R245fa turbine ........136
Figure 82. Comparison of fluid influence on $\beta$ of assumed efficiencies and calculated efficiency. Calculated efficiency (top left), assumed 60% efficiency (top right), assumed 70% efficiency (bottom left), assumed efficiency 80% (bottom right) ........................................................................................................................................138
Figure 83. Calculated ETAT for range of conditions listed in Table 29 ..................................138
Figure 84. Calculated ETAT versus $P_2$ and $T_h$ for pentane, R143a, R245fa and RC318 ......139
Figure 85. Comparison of fluid influence on evaporator pressure of assumed efficiency (top) versus calculated efficiency (bottom) ........................................................................................................................................140
Figure 86. Comparison of optimum fluid selection of assumed efficiencies and calculated efficiency........................................................................................................................................141
Figure 87. $\beta$ versus $P$ (left) and T-S diagram of maximum $\beta$ cycle (right) for ORC test rig based on cycle analysis with incorporated loss model ........................................................................................................................................145
Figure 88. $\beta$ versus ETAT (left) and ETAT versus Pressure (right) for ORC test rig based on cycle analysis with incorporated loss model. ........................................................................................................................................145
Figure 89. Relationship between turbine geometry and pressure for the ORC test rig turbine. ....146
Figure 90. Feasible turbines for optimum R245fa cycle in ORC test rig. $a_{II}$ versus power (left) and $Z$ versus power (right) ...................................................................................................................... 147

Figure 91. Individual turbine losses For optimum R245fa cycle in ORC test rig of feasible turbines .................................................................................................................................................. 148

Figure 92. Path and surface geometry for optimum R245fa turbine for ORC test rig. 2D paths of stator and rotor passage (upper left), 3D paths of stator and rotor passage (upper right), 3D surfaces of stator and rotor passage (lower left), divergent nozzle geometry (lower right) ................................. 149

Figure 93. Comparison of redesigned turbine cycle and the original turbine cycle .................. 150

Figure 94. Turbine Test Loop ........................................................................................................ 353

Figure 95. Example plot of RPM versus Time ........................................................................... 354

Figure 96. Javac XTR-Pro refrigerant recovery machine. ........................................................... 358

Figure 97. Refrigerant loop fill setup .......................................................................................... 358

Figure 98. Recovery machine valve settings for purge ............................................................... 359

Figure 99. Refrigerant withdrawal setup ...................................................................................... 360

Figure 100. Recovery machine valve settings for purge ............................................................. 361
List of Tables

Table 1. Geothermal Binary Power Plants Around The World (Franco A. 2009), (Gabriel 2009), ((NZGA) 2009), (Cuenot N. 2008), (Legmann 2003) ................................................................. 8
Table 2. Non-Exhaustive list of the main ORC manufacturers ((Quoilin 2009)) ........................................ 10
Table 3. Monthly mean maximum temperatures for Roma Airport from 1992–2011 .................................. 26
Table 4. Fluids analysed in cycle analysis .................................................................................................. 43
Table 5. Conditions of cycle analysis ....................................................................................................... 44
Table 6. Operating conditions for maximum $\beta$ for each fluid .............................................................. 44
Table 7. Monthly mean maximum temperatures for Roma Airport from 1992–2011 .............................. 53
Table 8. QGECE ORC Test Rig Specifications ......................................................................................... 66
Table 9. ORC Test Loop Component List ................................................................................................ 69
Table 10. Velocity Diagram Equations .................................................................................................... 73
Table 11. Test Impulse turbine design conditions investigated .............................................................. 78
Table 12. Test Impulse Turbine Design Conditions and Dimensions .................................................. 81
Table 13. Impulse Turbine Bolt Calculations .......................................................................................... 85
Table 14. Press fit calculations for the rotor to shaft fit .......................................................................... 85
Table 15. Housing inner diameter calculation based on rotor and housing displacement .................. 86
Table 16. Range of Conditions Tested ..................................................................................................... 100
Table 17. Comparison of turbine design conditions and test conditions .............................................. 110
Table 18. Average Loss Coefficients For All Test Cases ...................................................................... 111
Table 19. Tabulated comparison of loss coefficients for the different implementations $s$ of $N_{sector}$ ........................................................................................................................................ 112
Table 20. Experimental ORC Data versus ORCCA calculated values .................................................. 113
Table 21. Conditions for calculated turbine efficiency example ............................................................ 123
Table 22. Conditions for the influence of calculated turbine efficiency on evaporator pressure example .................................................................................................................. 130
Table 23. Comparison of optimum evaporator pressure for assumed efficiency and calculated efficiency ........................................................................................................................................... 130
Table 24. Conditions for the influence of calculated turbine efficiency on $EHX_{dT}$ ......................... 131
Table 25. Comparison of optimum $EHX_{dT}$ for assumed efficiency and calculated efficiency ........ 132
Table 26. Conditions for the influence of calculated turbine efficiency on $CHX_{dT}$ ......................... 133
Table 27. Comparison of optimum $CHX_{dT}$ for assumed efficiency and calculated efficiency......133
Table 28. Conditions for the influence of calculated turbine efficiency on power example ..........134
Table 29. Conditions for the influence of calculated turbine efficiency on working fluid selection example......................................................................................................................................137
Table 30. ORC Test Rig Operating Limits .................................................................................................................................144
Table 31. Dimensions of original turbine and the redesigned turbine for the small ORC test rig..151
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Sound speed</td>
<td>m/s</td>
</tr>
<tr>
<td>A</td>
<td>Area</td>
<td>m²</td>
</tr>
<tr>
<td>c</td>
<td>chord</td>
<td>m</td>
</tr>
<tr>
<td>C</td>
<td>Circumference</td>
<td>m</td>
</tr>
<tr>
<td>CHX</td>
<td>Condenser</td>
<td></td>
</tr>
<tr>
<td>Cp</td>
<td>specific heat</td>
<td>kJ/kg·C</td>
</tr>
<tr>
<td>D</td>
<td>Diameter</td>
<td>m</td>
</tr>
<tr>
<td>E</td>
<td>Power</td>
<td>kW</td>
</tr>
<tr>
<td>EHX</td>
<td>Evaporative Heat Exchanger</td>
<td></td>
</tr>
<tr>
<td>ETAC</td>
<td>Pump Isentropic Efficiency</td>
<td></td>
</tr>
<tr>
<td>ETAT</td>
<td>Turbine Isentropic Efficiency</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>Skin friction factor</td>
<td></td>
</tr>
<tr>
<td>Fs</td>
<td>Factor of Safety</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>Gap between rotor and stator</td>
<td>m</td>
</tr>
<tr>
<td>H</td>
<td>Enthalpy</td>
<td>kJ/kg</td>
</tr>
<tr>
<td>h</td>
<td>Passage height</td>
<td>m</td>
</tr>
<tr>
<td>j</td>
<td>Tip Clearance</td>
<td>m</td>
</tr>
<tr>
<td>K</td>
<td>Loss Coefficient</td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>Meridional Length</td>
<td>m</td>
</tr>
<tr>
<td>L</td>
<td>Power Loss, Pipe Length</td>
<td>W, m</td>
</tr>
<tr>
<td>m</td>
<td>Mass Flow Rate</td>
<td>kg/s</td>
</tr>
<tr>
<td>M</td>
<td>Moment, Mach Number</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>Number of nozzles or blades</td>
<td></td>
</tr>
<tr>
<td>o</td>
<td>Blade Throat Width</td>
<td>m</td>
</tr>
<tr>
<td>P</td>
<td>Pressure</td>
<td>kPa</td>
</tr>
<tr>
<td>p</td>
<td>Blade space/Pitch</td>
<td>m</td>
</tr>
<tr>
<td>Q</td>
<td>Heat, Volumetric Flow Rate</td>
<td>kJ, m³/s</td>
</tr>
<tr>
<td>q</td>
<td>Heat, subdivisions of Q</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Radius</td>
<td>m</td>
</tr>
<tr>
<td>Rc</td>
<td>Recuperation</td>
<td></td>
</tr>
<tr>
<td>Re</td>
<td>Reynolds Number</td>
<td></td>
</tr>
<tr>
<td>Reaction</td>
<td>Reaction</td>
<td></td>
</tr>
<tr>
<td>RHX</td>
<td>Recuperator</td>
<td></td>
</tr>
<tr>
<td>RP</td>
<td>Refprop Function</td>
<td></td>
</tr>
<tr>
<td>s</td>
<td>Solidity</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Entropy</td>
<td>kJ/kg·C</td>
</tr>
<tr>
<td>t</td>
<td>Blade Thickness, Pipe Wall Thickness</td>
<td>m</td>
</tr>
<tr>
<td>T</td>
<td>Temperature</td>
<td>deg C</td>
</tr>
<tr>
<td>te</td>
<td>Trailing Edge Thickness</td>
<td>m</td>
</tr>
<tr>
<td>U</td>
<td>Blade Tip Speed</td>
<td>m/s</td>
</tr>
<tr>
<td>V</td>
<td>Absolute Velocity, Fluid Velocity in Pipe</td>
<td>m/s</td>
</tr>
<tr>
<td>w</td>
<td>width</td>
<td>m</td>
</tr>
<tr>
<td>W</td>
<td>Work</td>
<td>kW</td>
</tr>
<tr>
<td>X</td>
<td>Fluid Quality</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>Relative Velocity</td>
<td>m/s</td>
</tr>
<tr>
<td>x</td>
<td>Cartesian x Coordinate of Turbine Geometry</td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>Cartesian y Coordinate of Turbine Geometry</td>
<td></td>
</tr>
</tbody>
</table>
Z = Speed-Work Parameter
z = Cartesian z Coordinate of Turbine Geometry
zw = Zwiefel coefficient

Subscript
I = Stator Inlet
1-11 = State Point Designation
1st = First Law
c = Cold Side Fluid
crit = critical point
f = Working Fluid
h = Hot Side Fluid, hydraulic
dT = Heat Exchanger Temperature Differential
i = in (i.e. Heat or Work)
II = Stator Outlet/Rotor Inlet
III = Rotor Outlet
iso = Isentropic
m = Refers to mean diameter
o = out (i.e. Heat or Work)
pinch = Pinch Point
r = Rotor / Blade
rt = Refers to root diameter
s = Stator / Nozzle
ss = Suction Surface
tp = Refers to tip diameter
tt = Throat
u = Tangential Direction
x = Axial Direction
xt = exit

Greek Symbol
γ = Machine specific exponential loss coefficient
β = Thermal Fluid Effectiveness, Relative Flow Angle kJ/kg, rad
Δ = Delta
ε = Active Fraction, Error
η = Efficiency
λ = Active Arc m
μ = Dynamic Viscosity, Local Mach Angle Pa-s
ξ = Energy Loss Coefficient
ρ = Density kg/m³
Φ = Machine specific linear loss coefficient
ω = Rotational Velocity rad/s
θ = Angle of Rotation rad
τM = Torque Coefficient Dimensionless
σ = Stress
ν = Kinematic Viscosity m²/s
α = Absolute Flow Angle rad

Superscript
' = Refers to metal angle
Chapter 1 Introduction

Geothermal energy is gaining more interest and support in Australia as the energy supply becomes a topic of debate on several fronts such as environment, security, and availability. The majority of the known geothermal resources in Australia are suited to engineered geothermal systems (EGS). EGS’s are man-made geothermal systems consisting of an injection well and production well. Figure 1 illustrates an example of an EGS.

![Diagram of an Engineered Geothermal System](image)

**Figure 1. Basic Layout of an Engineered Geothermal System with a Binary Power Station Using a Condensing Rankine Cycle with Recuperation**

EGS's are well suited to rock formations that produce heat but not water, Australia's heat producing granites are of this nature. They allow the heat from the formation to be extracted using brine that is circulated down the injection well, through a fracture network that connects to the production well and back to surface.

The heated brine can reach temperatures of 300°C from the production well. The heated brine at the surface can then be allowed to flow through a series of heat exchangers (labelled evaporator in Figure 1) where its thermal energy is transferred to a working fluid that is circulating through a power cycle. In cases where the heat from the geothermal fluid is exchanged to a working fluid, the power plant is termed a binary plant. Using a binary cycle as depicted in Figure 1 in conjunction with EGS's has several key advantages to traditional flash steam power plants.
In a binary system, brine is circulated thus minimizing the water requirements of the power plant. The brine can also be kept under pressure ensuring that it stays in the liquid phase which eliminates the requirement for a power plant to deal with non-condensable gases (NCG) as seen in flash plants. The equipment associated with NCGs can be expensive and ultimately the gases are vented to the atmosphere. In a binary system, a power plant can have zero emissions. Organic Rankine cycles (ORC), can also produce power more efficiently at lower temperatures than traditional steam flash plants because the working fluid in the ORC can be tailored to the temperatures available for a given geothermal source. A further advantage of a binary system is that the fluid passing through precision turbomachinery is clean fluid, whereas the geothermal brine will contain impurities acquired from when it passes through the formation from the injection to production well.

Traditionally, turbines have been developed and operated at temperatures higher than those available for geothermal binary power stations, because the majority of power plants in operation are steam power plants. A steam power plant will have significantly different operating conditions and properties compared with a geothermal binary power station. Therefore, the majority of commercially available turbomachinery technology isn't appropriate to EGS and there is a need to study designs specific to the expected EGS temperature ranges.

In this thesis detailed cycle analyses have been conducted over a wide range of fluids and conditions. The results indicate that cycles with high evaporator pressures yield more effective geothermal brine utilization. Also, the fluids that yield the best performance tend to be heavy refrigerants. These two initial findings, (high pressure and heavy molecular weight refrigerants as working fluids) are the key motivating factors for the research.

This thesis explores the incorporation of a turbine analysis with a cycle analysis. A supersonic impulse turbine potentially has the promise to deliver a power conversion unit that can operate at high pressure ratios using refrigerants as a working fluid and be relatively easy and inexpensive to manufacturer. This thesis will investigate the plausibility of using supersonic impulse turbines in geothermal binary power stations by examining cycle analysis that utilize an incorporated turbine loss model to more accurately predict the turbine efficiency and the overall cycle efficiency.

This research is being conducted through the University of Queensland's Queensland Geothermal Energy Centre of Excellence (QGECE). Therefore the focus of the work is for geothermal applications within Queensland itself.
The work is part of an integrated group of researchers whose individual work is collectively part of a greater effort. The scope of this work and its principle contributions have been:

1. The development of a cycle analysis program and an investigation of optimal operating conditions and working fluids and their relationship with thermal fluid temperature and coolant temperature as discussed in Chapter 5.
2. Development of a lab-scale ORC test facility and the experimentation of a single stage axial supersonic impulse turbine in an ORC using a refrigerant as the working fluid as discussed in Chapter 7.
3. Development of a flow-path design program and calibration of a single stage axial impulse turbine loss model based upon experimental results as discussed in Chapter 8, Chapter 9 and Chapter 10.
4. An examination of the influence and importance of a calculated turbine efficiency in ORC performance analysis as discussed in Chapter 11 using the calibrated loss model incorporated into ORCCA.
Chapter 2  Literature Review

To establish cause, the topic of energy supply and demand was first examined. The following pages in this chapter will examine the current energy situation, forecast future energy demands and the feasibility of geothermal energy being a significant contributor to the future energy supply.

2.1 Energy Supply

Supplying high quality energy is going to be an ever-increasing challenge as hydrocarbon supplies are strained while concurrently the global population and its energy demand continue to grow. Many analysts have predicted that the world's oil production will peak in the coming years. Some have even estimated that global oil production has already peaked in 2005 (Deffeyes 2005).

And transitioning from a global to a local perspective, in Australia, coal and liquefied natural gas (LNG) are major hydrocarbon energy resources. At current reserve-to-production ratios in Australia, the expected resource lifetimes for coal and LNG are estimated to be 110 years and 77 years respectively (Bartlett 2006). As these resources are produced and consumed, it will be important to find energy alternatives. In order to supersede current hydrocarbon based energy supplies (for various reasons i.e. supply, environment, economics, and security), more reliable and efficient alternatives will need to be developed to commercially competitive levels.

There are abundant energy sources available on earth in different forms of renewable energy sources; solar, geothermal, wind, tidal, hydroelectric and wave. Estimates of energy available on earth from renewable sources is listed as having a theoretical potential of ~174,000 TW (Abmann 2006). And focusing only on geothermal energy, this quantity of energy can be subdivided into a practical perspective based upon feasibility with regards to current technology and knowledge, the available energy becomes ~159 TW (Abmann 2006) for geothermal.

To compare the technically practical potential to global consumption, we can estimate the world’s average primary energy consumption as 2,100 watts per capita (Abmann 2006). For a global population of ~6 billion humans, the power used by humans worldwide equates to ~13 TW, only 8% of the technically available geothermal energy on earth. There is a vast supply of energy available from geothermal energy today with currently available technology, more than enough to meet global energy demands.

Australia is rich in geothermal resources. One cubic kilometre of hot granite at 240°C has the stored energy equivalent of 40 million barrels of oil when the heat is extracted to a temperature of
140°C. Australia is known to have several thousand cubic kilometres of identified high heat producing granites and these have the potential to meet the total electricity demand of the country for hundreds of years (Geodynamics 2009).

Another estimate of Australia's hot rock energy potential at 150°C at 5 km depth is ~190,000,000 PJ (Goldstein 2007). In 2006-2007 Australia's electricity consumption was reported as 941 PJ (ABARES 2009). Based on this annual energy consumption, the estimated reserves for hot rock energy would be enough to power Australia for more than 200,000 years.

These estimates on Australian geothermal energy reserves vary but they both agree that the available energy is enough to power Australia for many years into the future.

Currently the installed electricity generation capacity in Australia is ~50GWe (WNA 2010). To visualize what is really required to supersede fossil fuel based electricity generation, the following hypothetical scenario can be useful.

Assume that one-third of Australia's electricity will be generated from geothermal energy by 2100 (assume wind, solar, etc. will make up the remaining). To illustrate the plausibility of this scenario some reasonable assumptions can be made as follows:

a) The average mass flow rate per geothermal well is 20kg/s (equivalent to the stable production rate achieved at Soultz (Polsky Y. 2008).

b) The flowing temperature of the geothermal brine is 150°C and the exit temperature of the brine from a power station is 50°C.

c) An average conversion efficiency of 12% is realized by the power station.

d) By these assumptions, a single source well can be estimated to produce 1MW of power.

e) Energy consumption continues to grow at an average rate of 2% per year (annual average growth in energy consumption is around 2.3% (ABARES 2009). This would translate to a national installed capacity of ~300GWe by 2100 (current installed capacity is ~50GWe).

f) To produce 33% of national demand by 2100, 100GWe, from 1MW wells would require 100,000 production wells. This would require 1,100 wells per year over 90 years. 1,100 wells per year is a relatively small endeavour when you consider that in the U.S. 11,300 oil
and gas wells were drilled in only a three month period from July to September in 2010 (Landry 2010). Annually in the US alone, upwards of 44,000 wells are drilled per year.

g) Assume that each 1MW well would cost $17 million (inclusive of drilling, completion, stimulation, and relative portion of power station). The current cost now of such wells is $20 million (Polsky Y. 2008) and cost of associated power station equipment is $2 M (Sanyal 2007). It is reasonable to assume that if the volume of work increased to 1,100 wells per year the cost per well would decrease to $17 million per well if not less.

h) That would require a capital cost of $18 billion per year. For comparison, the coal seam methane industry in Australia has $18 billion invested in projects (Knights P. 2009).

i) In 2008 Australia’s Gross Domestic Product (GDP) was $1.25 trillion and the annual Australian Government spending amounted to $325 billion (Government 2010).

j) If all geothermal development is assumed to be government funded, this would mean that only 2% of GDP (6% of government revenue) each year for the next 90 years would be enough to install 100GWe of geothermal energy. More than likely though, with most projects, the cost would be shared between public and private entities. As Australia's economy continues to grow, the relative cost of the geothermal funding would continue to decrease in comparison to overall government revenue.

The intent of this hypothetical scenario was to demonstrate the economic feasibility of geothermal energy. Evidence of the economics being viable can be seen in how interest in Australia's geothermal technologies and capabilities has grown in recent years. Growing interest in geothermal energy is evidenced by political motivation to progress the energy industry towards cleaner and more sustainable sources. It is also exemplified by capital investments made by both the public and private sectors towards geothermal development. Since 2001, 30 companies applied for 208 licenses covering over 186,000 km² on a variety of projects in Australia (Goldstein 2007).

2.2 Binary Power Plants

In Australia there are both high and low temperature geothermal sources. There are the hot granites as previously mentioned but there are also low temperature sources readily available that could be utilized for small, rural homestead binary power generators. The Great Artesian Basin which is the best known of these artesian systems, underlies 1.7 million square kilometres of eastern central Australia (about 22% of the continent) (Habermehl 2002), and is currently producing around 1
million cubic meters of water per day from 3,000 bores, much of it at temperatures as high as 90-100°C (Swenson 2000).

These low temperature sites lend themselves to using binary power plant technology such as that in Birdsville, Queensland that utilises a 90°C source. Using binary power plant technology allows for closed-loop injector and producer well systems. By having a closed-loop system the brine can be circulated. If the brine is circulated it dramatically reduces the water requirements of a power station. This is important because in the majority of areas with large geothermal resources, there is not a large supply of water. Drawing large amounts of water from limited local water resources is not an acceptable practice and to import water would be cost prohibitive (Gurgenci H. 2008).

Binary power plants also provide the advantage of allowing the power cycle to be designed and operated with whatever fluid provides the optimum system efficiency for the given conditions of a site. With the option to select different working fluids a wider range of geothermal sources can be exploited. If the geothermal fluid temperature is 150°C or less, it becomes difficult to build a flash-steam plant that can efficiently and economically put such a resource to use (Dipippo 2005). However, if other working fluids are used then geothermal sources with temperatures as low as 90°C can be utilized. This greatly expands the definition of what is a usable geothermal resource.

Binary plants can also eliminate issues of scaling and NCGs in wells and surface equipment by producing the geothermal fluid under pressure (Dipippo 2005). In flash plants, steam undergoes a change in pressure and temperature that can allow for the precipitation of scale in both the well and in surface equipment. Dealing with scale can be expensive and timely. If scaling occurs in the well it requires an expensive well intervention that will likely consist of an acidizing operation. Often the acids used are extremely hazardous mixtures of hydrofluoric and hydrochloric acid. But when the geothermal fluid is kept pressurized and in a liquid state, NCGs are kept in solution and precipitates can more easily be managed, even eliminated.

Also, binary plants are emissions free whereas flash plants, though low compared to hydrocarbon based electricity production, do emit greenhouse gases such as CO₂, H₂S, CH₄, and NH₃ (Bloomfield K.K. 2003). These NCGs can be kept in solution if the geothermal fluid is kept under pressure. By eliminating the production of NCGs additional surface equipment can be eliminated and a power plants environmental impact can be reduced.

Today, binary plants are the most widely used type of geothermal power plant in operation. In 2005 it was reported that the total installed power worldwide for geothermal binary power plants
was about 700MWe, representing about 8% of the geothermal power installed worldwide (Franco A. 2009). The technology is growing as more companies are designing and building binary power plants. Binary power plants have become more and more cost competitive with flash plants on a per kW basis over the past 20 years (Farhar 2004) making the technology more attractive to investors and power suppliers. It has also been stated that for lower temperature sources (temperatures below 190°C), even though binary systems are more complex than steam flash plants, they are generally less expensive (Hance 2005). Table 1 lists some of the binary power plants in operation around the world.

Table 1. Geothermal Binary Power Plants Around The World (Franco A. 2009), (Gabriel 2009), ((NZGA) 2009), (Cuenot N. 2008), (Legmann 2003)

<table>
<thead>
<tr>
<th>Plant Site</th>
<th>T_h (°C)</th>
<th>Cycle Type</th>
<th>Fluid</th>
<th>Gross Power (kWe)</th>
<th>Cooling Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nigorikawa, Japan</td>
<td>140</td>
<td>Rankine</td>
<td>R114</td>
<td>1000</td>
<td>Wet</td>
</tr>
<tr>
<td>Otake, Japan</td>
<td>130</td>
<td>Rankine</td>
<td>Isobutane</td>
<td>1000</td>
<td>Wet/Dry</td>
</tr>
<tr>
<td>Husavik, Iceland</td>
<td>124</td>
<td>Kalina</td>
<td>NH3-H2O</td>
<td>1700</td>
<td>Wet</td>
</tr>
<tr>
<td>Nagqu, China</td>
<td>110</td>
<td>Rankine</td>
<td>Isopentane</td>
<td>1000</td>
<td>Dry</td>
</tr>
<tr>
<td>Altheim, Austria</td>
<td>106</td>
<td>Rankine</td>
<td>C5F12</td>
<td>1000</td>
<td>Dry</td>
</tr>
<tr>
<td>Wabuska, CA, USA</td>
<td>104</td>
<td>Rankine</td>
<td>Isopentane</td>
<td>1750</td>
<td>Wet</td>
</tr>
<tr>
<td>Chena Hot Spring, AK, USA</td>
<td>74</td>
<td>Rankine</td>
<td>R134a</td>
<td>400</td>
<td>Wet/Dry</td>
</tr>
<tr>
<td>Kutahya-Simav, Turkey</td>
<td>145</td>
<td>Rankine</td>
<td>R124</td>
<td>2900</td>
<td>Wet</td>
</tr>
<tr>
<td>Birdsville, QLD, Aus</td>
<td>98</td>
<td>Rankine</td>
<td>Isopentane</td>
<td>120</td>
<td>Wet</td>
</tr>
<tr>
<td>Wairakei, NZ</td>
<td>200</td>
<td>Rankine</td>
<td>Isopentane</td>
<td>1400</td>
<td>Dry</td>
</tr>
<tr>
<td>Red Rock, AZ, USA</td>
<td>240</td>
<td>Rankine</td>
<td>n-pentane</td>
<td>1350</td>
<td>Wet</td>
</tr>
</tbody>
</table>

Although binary technology is growing and being commercialized it is not yet capable of providing off-the-shelf machinery, meaning that each installation is designed for the conditions at a given location. Every system is tailored to specific geothermal site conditions (Franco A. 2009). For geothermal energy technologies to be more cost competitive with hydrocarbon power plants, they need to achieve a level of sophistication and development that allows for expedient design and high volume manufacturing. If this can be achieved the cost can be reduced and the benefits of an economy of scale can be realized.

Binary power plants can employ a range of cycle types; Rankine Cycle, Kalina Cycle, and Brayton Cycle. For each type of cycle there are further variations and derivatives such as those that include feed heating, dual pressure, recuperation and supercritical.

The Kalina cycle uses a water-ammonia binary working fluid. The binary fluid allows the working fluids evaporation curve to match closely to that of the geothermal brine cooling curve.
Herein lies the Kalina cycles main advantage, a reduction of irreversibilities in the evaporator (Dipippo 2005). Another advantage of the Kalina cycle lies in the fact that a Kalina power plant can use the same devices (turbine, pumps, valves, etc.) as a conventional steam power plant, since the molecular weight of the ammonia-water mixture only varies slightly from pure water (Lolos 2009). This makes it easier to design a power station from standard machinery.

However, it has been shown that there is little difference between the low-heat exchanger irreversibilities realised by a Kalina cycle in comparison with a supercritical Organic Rankine cycle (SORC). Further, Bliem (1991) et al. showed that SORCs produce 2.5% more net power per unit flow rate of geothermal fluid. Similarly, Guzovic et al. (Guzovic 2012) showed for medium temperature geothermal sources, SORCs will have higher thermal efficiencies than Kalina cycles. This translates directly to lower field development costs by using an SORC. If more power can be produced from a given mass flow rate of geothermal brine, then fewer wells need to be drilled, completed, stimulated and maintained.

Binary Rankine cycles with high pressure ratios appear to have several advantages for use with geothermal brine. They have the ability to operate at supercritical conditions and reduce irreversibilities in the evaporator given that supercritical fluids have no latent heat of evaporation. The temperature profile of working fluids in the evaporator at supercritical conditions can be designed to match closely the temperature profile of the geothermal brine. Supercritical cycles can be designed easily to have dry expansion of the working fluid through the turbine which reduces losses associated with liquid droplets flowing through the turbine. They can also operate over a wide range of pressures and temperatures and may be designed to accept recuperative heating. These advantages and flexibilities are possible because binary Rankine cycles can be designed to operate with many different working fluids. Having the ability to select the working fluid based on the operating temperatures, supercritical cycles can be designed from low to high operating temperatures.

Quoilin et.al. listed some of the major ORC manufacturers in operation as of 2009 as listed in Table 2. The list shows that a range of powers are covered, a range of applications are targeted, a range of temperatures can be exploited and that a range of technologies are being used. They also noted that R134a, R245fa, n-pentane and silicon oil are the four most common working fluids in ORC (Quoilin 2009). It’s been shown that positive displacement machines are preferable for small-scale applications but a “difficulty associated with the use of a positive displacement machine is its lubrication. An oil separator could be installed at the expander exhaust. Unlike with compressors, an oil
pump is necessary to drive the separated oil back to the expander suction” (Quoilin 2009). “Alternatively, oil-free machines could be used, but generally show lower volumetric performance due to larger tolerances between moving parts” (Quoilin 2009). They noted as well that most of the positive displacement machines in use are modified compressors and that turbomachines are designed for larger scale applications and show a higher degree of maturity.

### Table 2. Non-Exhaustive list of the main ORC manufacturers (Quoilin 2009)

<table>
<thead>
<tr>
<th>Manufacturer</th>
<th>Applications</th>
<th>Power Range</th>
<th>Heat Source Temperature</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORMAT US</td>
<td>Geothermal WHR Solar</td>
<td>200 kWe-72MWe</td>
<td>150-300°C</td>
<td>Fluid: n-pentane</td>
</tr>
<tr>
<td>Turboden Italy</td>
<td>Combined Heat and Power Geothermal CHP</td>
<td>200kWe-2MWe</td>
<td>100-300°C</td>
<td>Fluids: octamethyltrisilox-ane, Solkatherm Axial Turbines</td>
</tr>
<tr>
<td>Adoratec, Germany</td>
<td>Waste Heat Recovery Geothermal</td>
<td>315-1600kWe</td>
<td>300°C</td>
<td>Fluid: octamethyltrisilox-ane</td>
</tr>
<tr>
<td>GMK Germany</td>
<td>Waste Heat Recovery Geothermal</td>
<td>50kWe-2MWe</td>
<td>120-350°C</td>
<td>3,000RPM Multi-Stage Axial Turbines Fluid: GL160</td>
</tr>
<tr>
<td>Koehler-Ziegler Germany</td>
<td>Combined Heat and Power</td>
<td>70-200kWe</td>
<td>150-270°C</td>
<td>Fluid: Hydrocarbons Screw Expander</td>
</tr>
<tr>
<td>UTC US</td>
<td>Waste Heat Recovery Geothermal</td>
<td>280kWe</td>
<td>93°C</td>
<td>N/A</td>
</tr>
<tr>
<td>Cryostar</td>
<td>Waste Heat Recovery Geothermal</td>
<td>N/A</td>
<td>100-400°C</td>
<td>Radial Inflow Turbine Fluids: R245fa, R134a</td>
</tr>
<tr>
<td>Freepower UK</td>
<td>Waste Heat Recovery</td>
<td>6kWe-120kWe</td>
<td>180-225°C</td>
<td>N/A</td>
</tr>
<tr>
<td>Tri-O-Ge Netherlands</td>
<td>Waste Heat Recovery</td>
<td>160kWe</td>
<td>350°C</td>
<td>Turbo-expander</td>
</tr>
<tr>
<td>Electratherm US</td>
<td>Waste Heat Recovery</td>
<td>50kWe</td>
<td>&gt;93°C</td>
<td>Twin Screw Expander</td>
</tr>
<tr>
<td>Infinity Turbine</td>
<td>Waste Heat Recovery</td>
<td>250kWe</td>
<td>&gt;80°C</td>
<td>Fluid: R134a Radial Turboexpander</td>
</tr>
</tbody>
</table>

Turboden has installed a 1MW ORC in Lienz, Austria in operation with a biomass-fired combined heat and power cycle. The system uses an axial turbine and a silicon oil as the working fluid and employs recuperation. The facility has been reported to achieve electric efficiencies of 15% (Obernberger 2002).

Optimisation of ORC’s has been a popular research topic for many years and by many individuals and institutions. However it is an area of research that still hasn’t reached full maturity. With advances in computer modelling techniques and fluid property information more detailed analyses are being conducted. Shengjun (2011) et al. explored the issue of cycle optimisation and showed that there are optimum evaporator pressures and turbine inlet temperatures unique to each fluid and
set of site conditions. Roy (2011) et al. published similar findings that optimum pressures and temperatures exists but also went on to say that system mass flow rate is non-linearly related to gross power suggesting an optimum cycle mass flow rate as well. Baik (2011) et al. published findings on the topic of ORC performance that indicated that there are optimum operating pressures and temperatures as well as there being a benefit to operating an ORC with supercritical pressures in the evaporator. Kang (2012) showed experimentally, using an ORC test rig employing a low temperature, low pressure R245fa cycle, that there appears to be a correlation between evaporator temperature and cycle efficiency. Rahman (2011) stated that cycle efficiency is related to pressure ratio, however this was based on assumed turbine efficiency. Pan (2012) et al. showed that there is potential for considerable gains by operating ORC’s at supercritical pressures. Pan also stated that operation at near critical conditions could compare favourably to supercritical conditions due to abnormal phenomenon near the critical point. Wang (2011) et al. published contour maps relating pressure and temperature to thermal efficiency and noted that recuperation can improve a cycle’s efficiency. However, the recuperation model used an assumed efficiency and did not take into account pinch point effects in the recuperator. Franco A. (2009) et al. published a comprehensive paper on the topic of ORC design for geothermal applications that stated the importance of proper fluid selection, operating conditions and paying close attention to the utilisation of the thermal energy in the geothermal fluid as opposed to focusing solely on thermal efficiency. Saleh B. (2007) et al. paid particular attention to the importance of pinch point in the evaporator on the performance of a cycle. This thesis will continue the work and conduct experimentation on some aspects of the ORC in a small scale ORC test lab.

2.3 Cycle Analysis

Dipippo presented an evaluation metric for binary geothermal cycle analysis. He stated that “The Carnot cycle is the standard textbook ideal for power generation. The simplicity of the formula for the maximum thermal efficiency entices many to apply this in certain instances where an alternative is more appropriate. Such a case occurs with geothermal binary plants. There is in this case a lower upper bound on the thermal efficiency; that limit is found from the triangular cycle” (Dipippo 2007). He also noted that real binary power stations have demonstrated relative efficiencies of about 55% based on the maximum triangular cycle efficiency.

In EGS applications, subsurface costs (i.e. drilling, completions and stimulation) are substantial so minimising the required brine mass flow rate and maximising energy production are primary design objectives. Because of this, Franco used a merit parameter called specific brine consumption
which is the geothermal fluid mass flow rate required to generate a fixed power output. Franco stated “The parameter is often considered when the minimization of geothermal fluid flow rate (specific consumption) for a given power is suggested as an objective function for optimal design” (Franco A. 2009). Franco et. Al. noted that first law efficiencies for Rankine cycles with brine source temperature of 110°C are about 6% and increase to 12% for source temperatures of 160°C. They stated that “for each combination of geothermal fluid temperature and working fluids, there is a particular recovery cycle that permits maximization of the thermodynamic performance of the system. The important point is that the optimal design for each working fluid leads to a similar performance if one finds the best match between the working fluid, the recovery cycle and the geothermal brine” (Franco A. 2009). Another interesting finding from Franco et. al. was that the optimal condenser temperature was not necessarily the lowest possible. They noted that there is a range of “10–20°C above the average ambient temperature over which no beneficial effects are obtained by reducing the condensation temperature. This is because the higher thermodynamic performance of the recovery cycle is negated by the increase in fan power requirements” (Franco A. 2009). They commented as well on the effects of supercritical cycles noted that supercritical cycles can improve first law efficiency by about 6% and the use of supercritical cycles is appropriate for cycles with brine temperatures above 140°C. But they also note that the use of supercritical cycles should be considered carefully because the “efficiency increase of the heat recovery system is negated by an increase of the parasitic energy requirements and a reduction in the enthalpy drop” (Franco A. 2009).

Gawlik et. al. conducted a study on the influence of hydrocarbon mixtures as working fluids for geothermal binary plants. They used a figure of merit called geofluid effectiveness which is the work over the geothermal brine mass flow rate. It’s similar to the merit parameter suggested by Franco, just the inverse. Gawlik et. al. noted that supercritical cycles have lower levelised electricity costs. They noted that there is a potential for plant performance improvement by designing a plant to use a hydrocarbon mixture as the working fluid and that the lower the resource temperature the greater the advantage of hydrocarbon mixtures.

Saleh et.al. investigated working fluids for low-temperature organic Rankine cycles. They mention that for ORC’s where the working fluid is heated in a closed cycle as in solar applications, a thermal efficiency is desirable. However in the case of geothermal applications where the heat carrier fluid is discharged or reinjected into the ground after having gone through the heat exchanger it is desirable to maximise work output per unit mass flow rate of the heat carrier fluid
(i.e. geothermal brine). They also stated that “fluids with lower critical temperatures as R143a in a s2 [supercritical superheated cycle] or R152a in a b3 [bell-shaped TS diagram fluid in a subcritical cycle with superheating] process are favourable because they yield a more uniform increase of the temperature-enthalpy curve during heating” (Saleh B. 2007).

2.4 Impulse Turbine

Cycle analysis shows that ORC's theoretically have better performance at higher pressure ratios. To take advantage of cycles with high pressure ratios a turbine needs to be able to operate efficiently at high pressure ratios.

Axial flow impulse turbines can be designed to operate efficiently at both subsonic and supersonic velocities; additionally, they can accommodate extremely high stage pressure ratios. Impulse turbines have been designed to operate at pressure ratios as high as ninety to one (Barbers Nichols 2010). The impulse design lends itself to these high expansion ratios because essentially all expansion occurs in the stator nozzles. The rotor blades do not have to accept large changes in volume as the flow transits the rotor (Barbers Nichols 2010). They can also be designed to operate at both sub and supersonic velocities allowing them to suit a wide range of conditions.

If an impulse turbine is designed to operate at supersonic velocities it may lead to a more economic design. Supersonic turbines have the potential for large specific work outputs because of the high pressure ratio. For a given power level, this type of turbine would require a small amount of driving fluid and a small number of stages, it would therefore, be light-weight and relatively simple (Glassman 1994). Warner (1958) showed that there is a limiting efficiency for turbines and that theoretically a single stage turbine could achieve the same efficiency as a multistage turbine if the single stage is designed for a high speed-work parameter (high rotational speeds). A high-speed, supersonic, single-stage, impulse turbine has the potential to deliver high efficiencies and be compact and inexpensive and relatively simple to build.

A supersonic impulse turbine shows the promise of being able to deliver an efficient and economic turbine for ORC's. But to date, most the development of impulse turbines has been focused on using steam as the working fluid. The working fluids that are the primary candidates for geothermal binary ORC's tend to be from a class of refrigerants, hydrofluorocarbons (HFC). To better investigate the possibility of using supersonic impulse turbines in ORC’s, computer-aided cycle analyses need to incorporate real turbine efficiency models rather than assume turbine efficiency.
Another motivation for investigating impulse turbines relates to the economics of geothermal power station equipment. The cost of the surface installations, the power conversion and auxiliary systems, is over 60% of the total cost of a geothermal power plant (Gurgenci H. 2008). So if progress can be made in reducing the cost of surface equipment, geothermal power plants may become more competitive. There appears to be a window of opportunity with regards to the turbine design given the types of cycle conditions calculated to be optimum for geothermal sites, however, there needs to be a reliable predictive method for estimating machine performance in the specific regime relevant to geothermal energy production.

Li et. al. conducted experimental investigation of a single stage axial impulse turbine for use in a regenerative ORC. A single stage axial impulse turbine was designed, tested and analysed in CFD. They noted that the major source of losses in the tested turbine was shock losses and partial admission pumping and sector losses. They also commented on the influence of turbine inlet pressures on stator nozzle exit velocity. It was seen that “higher turbine inlet pressure leads to a higher vapour velocity at nozzle outlet and the higher vapour velocity leads to a higher turbine rotation speed. The turbine isentropic efficiency reaches its maximum when the turbine rotation speed reaches the design value (3000rpm). When the rotation speed exceeds 3000rpm the turbine isentropic efficiency decreases. Therefore, the turbine inlet pressure influences the turbine rotation speed and then influences the performance of the turbine” (Li 2012).

**Losses**

One of the first and most widely used loss models for turbine performance prediction is the model presented by Ainley and Mathieson in 1951. They developed a mean-line model based on experimental data on blades having a conventional profile shape for gas and steam turbines in England and the U.S.A. at that time. They compared their calculated results to a number of axial single and two stage turbines. In their comparison they found that the calculated efficiency near the designed operating conditions was within +/-2% of the test efficiency and at a given speed and pressure ratio the calculated flow was within +/-3% (Ainley 1957).

An improved method for loss prediction was presented by Dunham and Came in 1969. It was a modification of the Ainley-Mathieson method. The profile losses in the Ainley-Mathieson method were based on an “empirical function of blade inlet angle, maximum thickness, trailing edge thickness, pitch, chord, incidence and gas relative outlet angle, but not Mach number” (Dunham & Came 1970). Upon the profile losses were added secondary losses and tip clearance losses. Dunham and Came improvements were based on accounting for losses associated with supersonic
Mach numbers and a Reynolds number correction to the profile loss. They also suggested a revised secondary loss with a new blade loading parameter and numerical constant to account for wall boundary layer thickness and blade shape. And a further modification was to the clearance loss by making the clearance loss based on a power law relationship to tip clearance (Dunham & Came 1970). They made the statement that “The Ainley-Mathieson performance prediction method has been tested against experimental data from 25 turbines. Although it was satisfactory for typical aircraft turbines, it proved to be misleading for small turbines”. Their modifications made the model more applicable to smaller turbines by changing the loss correlations, particularly the secondary loss correlation.

Kacker and Okapuu published a meanline prediction method for axial flow turbine efficiency (Kacker 1982). Their method expanded upon AMDC method by separating out the trailing edge loss coefficient from the other terms. Aungier (Aungier 2006) published a system that expanded the AMDC-KO method to be more reliable in severe off-design conditions. Aungier stated that the AMDC-KO methods were based on one-dimensional mean-line analyses. The performance prediction method presented by Augnier was intended to be a more general hub-to-shroud performance analysis. A number of modifications were made to achieve a performance system that was reliable under sever off-design conditions and was applicable to modern high-pressure steam turbines. Aungiers system included total pressure losses, parasitic losses and leakage mass flow. The total pressure losses accounted for irreversible processes as the flow passes through the blade. These losses include profile, secondary flow, clearance, trailing edge, supersonic expansion, and shock losses. The parasitic losses accounted for lost work due to disk friction, shear forces, partial admission and leakage. The parasitic losses cause a reduction in efficiency but not a reduction in total pressure (Aungier 2006).

Roelke (1994) wrote on the topic of miscellaneous losses in the blade channel. The miscellaneous losses that Roelke suggested be considered to accurately determine a design point efficiency are tip clearance, disc-friction (windage), partial admission (sector and pumping), and incidence losses.

Many individual turbine design groups have produced their own models for individual families of turbines but they are usually only modifications of coefficients to the fore mentioned loss models (Moustapha 2003). Moustapha (2003) et al. noted that a mean-line loss cannot reproduce the full complexity of the flow in a real turbine and therefore machine specific loss coefficients obtained experimentally will be needed to accurately model the performance a turbine. Filling this need in
the geothermal regime, the material in Chapter 9 and Chapter 10 provide the details of a combined loss model calibrated with experimental data.

**Tip Clearance Loss**

A number of tests at the NASA Lewis Research Centre were conducted on axial flow impulse turbines to better understand tip-clearance losses. Experimentation on a 5-inch single stage turbine was conducted for various tip clearances. The tests showed that flow in the clearance space and at the tip was not fully turned and that under-turning of flow increased with clearance. The under-turning of the flow unloads the blade and reduces efficiency (Roelke 1994). It was also shown that reaction has a significant impact on clearance losses. The clearance losses for reaction turbines was about double that of an impulse turbine. The tip clearance loss increased with increased ratio of clearance gap to blade height. For small turbines this is an issue because the blade height is inherently small so ratio of clearance gap to blade height increases leading to increased tip clearance losses in smaller turbines.

The Ainley-Mathieson method coupled secondary losses and tip clearance losses together for convenience because they were both shown to be related to blade loading and pitch-chord ratio. Their model was stated to be applicable for a wide range of gas inlet angles (Ainley 1957). Dunham and Came improved upon the tip clearance model of Ainley and Mathieson by making the clearance loss based on a power law relationship to tip clearance (Dunham & Came 1970). Aungier suggested a more fundamental clearance loss based on seal leakage calculation. Aungier’s clearance loss “is simply the seal pressure drop times the ratio of leakage mass flow to the total mass flow” (Aungier 2006).

**Windage Losses**

Roelke presented a model for windage losses based on the work of Daily and Nece (Daily 1958) that were based on a fluid shear stress that acted on the rotor disc creating a resisting torque to disc rotation. Roelke’s model investigated the effects of chamber proportions on disc friction to yield a more accurate representation of the flow of the different flow regimes that may exist. The regimes were delineated as (i) Laminar, merged boundary layers (small clearance), (ii)Laminar, separate boundary layers (large clearance), (iii)Turbulent, merged boundary layers (small clearance) and (iv) Turbulent, separate boundary layers (large clearance). A torque coefficient was presented for each flow regime based on theoretical and experimental evaluation.
Aungier also used the model presented by Daly and Nece for estimating losses due to disc friction. However Aungier also added a model for estimating the losses due to shear forces on the rotating wall associated with the clearance gap (Aungier 2006). Daly and Nece model only accounted for friction associated with the side walls of the rotor disc.

**Partial Admission**

Full admission axial turbines will in general yield higher efficiencies, however circumstances arise that don’t permit full admission such as low mass flow rates. In cases were full admission is not possible, partial admission turbines are used. Partial admission turbines incur performance penalties that can be significant. The partial admission losses are divided into two categories, pumping losses and sector losses. The pumping losses refer to inactive blades rotating in a fluid filled casing (Roelke 1994) as well as flow from active blades re-entering inactive blade passages (Aungier 2006). The sector losses refer to the stagnant fluid in inactive blades having to be accelerated by the fluid from stator nozzles as the inactive blade enters the active arc. The sector loss encompasses the loss from high velocity fluid in active blades decelerating as the blade leaves the active arc. In partial admission machines if all stator nozzles are grouped together there will only be one sector but in cases where the stators nozzles are spaced apart there are multiple sectors and the sector losses proportional to the number of active stator sectors. More sectors induce greater losses.

Roelke showed that design point blade-jet speed ratio decreased with decreasing degrees of admission. Fridh (2004) et al. and Cho (2006) et al. also experimentally showed this relationship between optimum blade speed and admission rate.

**Sector Losses**

Roelke (1994) cites findings reported by Stenning (1953) which investigated the effects of partial admission on axial turbine performance across a range of admission rates for predicting sector losses. The model is based upon fitted data between 12% to 100% admission rates

However this model seemed to under predict the losses. This was reported by Cho (2006) et al in a paper on the performance prediction on partially admitted turbines. Similar findings were encountered in this study. Aungier (2006) reported that the model published by Suter and Traupel is about the best model available to describe the sector losses. Varma and Soundranayagam (2012) reported that the theory of Stenning’s sector loss combined with Suter and Traupel’s empirical correlation for the effect of multiple sectors provided reasonably accurate results of sector losses for
their experimental study on low aspect ratio axial turbines. They also showed experimentally that increasing the number of sectors did increase the losses as predicted by the Suter and Traupel model.

The implication of the multiplicative effect of the number of sectors on sector losses is that for partial admission machines all stator nozzles should be grouped together so that only one active sector exists. The greater the number of active sectors the greater the sector losses.

**Pumping Losses**
The pumping loss is the loss associated with inactive blades rotating in fluid. The pumping loss model is similar in form to the windage loss. It is exponentially dependent upon blade speed. Aungi (2006) and Roelke (1994) both published models for partial admission pumping losses that are very similar. The difference in the two models is that Augnier’s model does not place as much emphasis on blade height by raising blade height to the power of 1.5 as in Roelke’s model.

**Trailing Edge Loss**
The trailing edge thickness of the blade causes the flow to separate at two points. Between these two points there is a region where the pressure is significantly lower than the freestream pressure. Downstream of the trailing edge, this low pressure region merges with the boundary layers from the pressure and suction surfaces of the blade to form a single wake, which then dissipates into a single stream through shear (Baines 1997).

Okapuu and Kacker (Kacker 1982) generated a relationship for the trailing edge loss in terms of an energy coefficient versus trailing edge thickness to throat opening ratio. They published relationships for two distinct cases, one for an axial entry and one for an impulse blade.

**Incidence**
Incidence losses occur when the gas flow angle differs from the designed blade angle (absolute angle for stators and relative angles for rotors). The incidence angle may be positive or negative and the sign of the incidence is important as Roelke (1994) showed that variation of incidence loss differs for positive and negative incidence angles. Roelke also showed that minimum loss does not occur at zero incidence but an optimum incidence, usually between -4° to -8°.

Ainley and Mathieson presented a relationship between the ratio of profile loss to zero incidence profile loss and the ratio incidence to stall incidence. The stall incidence being defined as the incidence at which profile loss becomes twice the profile loss at zero incidence. For impulse rotors
this model is to be limited to rotors with maximum blade thickness to chord ratios between 0.15 and 0.25.

Aungier presented an empirical model for stalling incidence based on the data of Ainley and Mathieson. The empirical model is based on a reference value for incidence for pitch-chord ratio of 0.75 as presented by Ainley and Mathieson and a correction term. The extrapolation of the Ainley and Mathieson data by Aungier allows for wider range of application (Aungier 2006).

**Secondary Losses**
Moustapha (2003) describes secondary losses as

“Secondary flows are generated as the flow turns in the blade passage. In the endwall boundary layers the fluid velocities are lower than in the mainstream, and the cross-passage pressure gradient causes the fluid there to turn more sharply than in the centre of the passage. The boundary layer fluid therefore rolls up into the passage vortices.

Additionally, the endwall boundary layer rolls up in front of the blade and forms a vortex usually known as the horseshoe vortex because of the shape it forms as it distributes itself about the pressure and suction sides of the blade”.

Ainley and Mathieson used a relationship of an empirical secondary loss factor, lift coefficient and the pitch-chord ratio. Their secondary losses estimation was based on the derived total loss minus the estimated profile loss minus the estimated tip clearance loss. They stated that the “absolute error in this secondary loss is likely to be considerable since it will comprise the separate errors involved in estimating the total loss, the profile loss, and the tip clearance loss”. (Ainley 1955).

Dunham and Came built upon the secondary loss model of Ainley and Mathieson by suggesting a new blade loading parameter based on work they performed on fitting cascade test data. They also introduced a single numerical constant to represent the effects of wall boundary layer thickness and blade shape. “The numerical constant was chosen from comparisons with overall efficiency data, and it also compensates for the use of reference diameter blade loading values instead of values at the ends of the blade” (Dunham & Came 1970). They stated that the main improvement over the Ainley-Mathieson method was for low aspect ratio and low reaction turbines (Dunham & Came 1970). Small turbines will likely have low aspect ratios and therefore the modified model would produce more accurate results for smaller turbines.
Aungier built upon the AMDC model by introducing an aspect ratio correction factor for low aspect ratios less than 2. This was to help the model’s accuracy at low aspect ratios because the original was found to be slightly pessimistic at low aspect ratios (Aungier 2006). Aungier also noted that “Use of this mean-line model in a hub-to-shroud analysis can result in excessive values of this loss coefficient at extreme off-design conditions near the end-wall contours. Although this rarely occurs, it was found necessary to impose an asymptotic upper limit on the preliminary estimate of about 0.365 to permit an analysis under those extreme conditions” (Aungier 2006).

**Influence of Mach Number**

Ainley and Mathieson noted that at the time of their work, turbine designers had not employed supersonic Mach numbers in their designs due to limitations on running speed and fear of a loss of efficiency. But they did note that supersonic designs could be employed advantageously in low temperature refrigeration applications (Ainley 1955). In the Ainley-Mathieson model the gas outlet angle was related to outlet relative Mach number but the profile losses were not.

Dunham and Came stated that the Ainley-Mathieson model appeared to be reasonably good except at supersonic Mach numbers. They introduced a simple correction factor that penalizes supersonic outlet velocities but note that the penalty would not apply to blading especially designed for high Mach numbers (Dunham & Came 1970).

Kacker and Okapuu introduced a shock loss estimated from mean-line flow data. Aungier modified the Kacker and Okapuu model to be used in a hub to shroud analysis. Aungier stated that “Overall flow diffusion across the blade row is a significant loss source that has been ignored by all published loss systems. That is not too surprising since overall diffusion is rarely encountered in a performance analysis conducted along the mean stream surface. But it does occur in a more complete hub-to-shroud performance analysis, particularly near the hub stream surface at far off-design operating conditions” (Aungier 2006).

Aungier also included a loss to account for supersonic expansion. This is to account for when the flow at the discharge of a blade is over-expanded to supersonic speeds and shock waves form creating drag losses. Aungier notes that Dunham and Came imposed an arbitrary multiplier to the profile loss to account for the loss. Both Aungier and Kacker and Okapuu make mention of the weakness of the multiplier suggested by Dunham and Came because it was based on very little available information on the effects of supersonic expansion. Aungier’s suggested model is
independent of profile loss and notes that “there appears to be little reason to expect that the supersonic expansion loss should depend on the profile loss” (Aungier 2006). A study by Colonna et. Al on the real gas effects in Organic Rankine cycle turbine nozzles showed that “significant nonideal flow effects lead to rather unsatisfactory Mach numbers and pressure coefficient distribution along the blade which may substantially deteriorate turbine performance”(Colonna 2008). They also noted that in such flow conditions CFD would aid in improved designs.
Chapter 3 Thesis Objective

This thesis was conducted with the support of the University of Queensland’s Geothermal Energy Centre of Excellence (QGECE). The centre had four major challenges:

1. Optimum energy extraction and sustainable resource management;
2. Efficient power conversion. The Centre will explore radically new options based on synergies with other generation technologies, especially solar-thermal and natural gas augmentation. It will also review possibilities which have been proposed in earlier research;
3. A cooling system for a desert zone in the world’s driest inhabited continent. This will demand extreme efficiency at condensing the working fluid. As advances in cooling have benefits for conventional power plants, innovative platforms for cooling systems will be a significant focus of the Centre;
4. To resolve transmission issues inherent to a power plant which is located more than 500km from major load centres and the national grid.

This thesis is related to the second major challenge of improving power conversion efficiency. That challenge had an associated milestone to develop working laboratories for testing power conversion systems. A working organic Rankine cycle with a turbine was required to be designed, built, commissioned and tested. This thesis aligns itself with the goals of the centre to develop working facilities and to investigate opportunities to improve power conversion efficiency.

The synergies of the centres goals and this thesis’s objectives are based on binary power plant technology for use with geothermal applications. This thesis set out to develop infrastructure for QGECE and to also postulate a question that could be answered in part by the development of an organic Rankine cycle test facility.

This thesis postulates the question of what is the impact of incorporating a real turbine loss model into a binary ORC analysis. This thesis sets out to answer the question of how important is it to use calculated turbine efficiency in a cycle analysis as opposed to using assumed turbine efficiency.

The aim of this thesis is to develop a working test facility and to use that facility to conduct experimentation on a turbine. The data gathered from the experimentation is intended to be used to calibrate models of an ORC analysis and a turbine loss model that can be run as part of an incorporated cycle analysis.
Chapter 4 Thesis Methodology

This thesis has several major tasks in order to achieve its objective. The methodology for achieving the objective will be as follows:

1. Develop of a lab-scale ORC test facility. The test facility should be capable of operating at pressure and temperatures that are associated with geothermal power sites. The facility should be able to operate with a range of working fluids.

2. Test and commission the facility to ensure that all data acquisition systems and safety systems work properly. The facility must capable of recording critical parameters such as pressure and temperature at each major component in the loop. The facility must be able to measure and record the mass flow rate of the working fluid in the loop. The facility must also be able to record the performance of the turbine (power, rotation speed and torque).

3. Design and build a single stage axial impulse turbine. The turbine will be a test specimen for testing the loop and also for gathering data on turbine performance.

4. Conduct a range of tests on the turbine in the ORC test loop and collect data on turbine performance.

5. The data gathered from experimentation will be used to calibrate a turbine loss model for a single stage axial impulse turbine.

6. Write a program for conducting cycle analysis on organic Rankine cycles.

7. Write a program for calculating losses in a single stage axial impulse turbine.

8. Use the experimental data to validate and calibrate the cycle analysis program and the turbine loss model.

9. Incorporate the loss model into the cycle analysis model. Run analyses to investigate the impact of using a calculated loss model versus using assumed turbine efficiency in a cycle analysis.

The desired final outcome of this thesis is to have developed a working test facility that can be used for years to come to conduct research on organic Rankine cycles for power conversion of geothermal energy. This work also aims to leave a platform for developing further analytic tools for conducting cycle analysis and turbine analysis.
Chapter 5 Cycle Analysis

As part of this thesis, a computer program was written to analyse ORC’s. The program, named ORCCA (Organic Rankine Cycle Computational Analysis) analyses ORC’s over a wide range of conditions and compares the performance of cycles for different fluids and operating conditions.

Other cycle analysis programs have been created to analyse ORCs. Tchanche (2008) et al developed a cycle analysis program to aid in selecting the optimum fluid for low temperature solar cycles. Their program modelled each component in the cycle as ideal (ignoring irreversibilities and pressure drops) except for the turbine where an assumed efficiency was used to estimate irreversibilities. The program also includes a pinch analysis of the evaporator which is important for analysing effectiveness of heat transfer in the heat exchanger.

Saleh B. (2007) et al published findings for ORC working fluid selection. An important finding from their work that was incorporated into ORCCA was the ability to investigate a cycle with and without Recuperation.

ORCCA adopted techniques and built upon the lessons published by the creators of the cycle analysis programs described above. It also incorporates additional functionality. It includes pinch analyses for not only the evaporator but for the regenerator and condenser. It allows for calculating across a wider range of parameters such as thermal fluid temperature, coolant temperature, evaporator pressure, heat exchanger temperature differentials, power, and recuperation (on/off). ORCCA was built with the intent to analyse a very large number of cycles with logic built into the program to determine whether or not cycles were feasible and realistic. Because ORCCA was built with the intent of analysing very large data sets several post processing functions are built into ORCCA to allow the results of large data sets to be presented in a meaningful way.

However, the most unique feature of ORCCA is that it uses a loss model for calculating turbine efficiency for the conditions of a given cycle rather than using assumed turbine efficiencies. The loss model used in ORCAA is discussed in Chapter 8 and the incorporation of the loss model in ORCAA is discussed in Chapter 11. However, to simplify the discussion of ORCCA as a cycle analysis tool, estimated turbine efficiencies will be used in this chapter.

The computational methodology and a discussion of results obtained from the program are presented in this chapter. The program focuses on analysing ORC’s. Simplistically, the cycle
consists of four processes. Figure 2 graphically depicts the processes as seen on a temperature-entropy (T-S) diagram.

![Temperature-Entropy diagram of a Rankine cycle with superheating](image)

**Figure 2. Temperature - Entropy diagram of a Rankine cycle with superheating**

The processes in an idealised Rankine cycle are summarised as follows:

- Process 1-2s: Isentropic compression of the working fluid in the pump.
- Process 2s-5: Isobaric heating of the working fluid. 2-3 is heating to a saturated liquid state, 3-4 is evaporation, 4-5 is superheating (superheating is optional).
- Process 5-6s: Isentropic expansion of the working fluid through an expander (i.e. a turbine).
- Process 6s-1: Isobaric cooling of the working fluid. 6s-7 is cooling to a saturated vapour, 7-1 is condensation.

These are the main processes that take place in an ideal Rankine cycle and are the basis of the cycle analysis methodology. The main physical components that comprise a Rankine cycle are a pump, evaporator, turbine and condenser; a recuperator may be employed in some cycles if the conditions allow.

Figure 3 shows the layout of the main components and the state points in the figure corresponding to the state points on the T-S diagram in Figure 2. Work is added to the cycle at the pump \( W_i \), heat is added to the cycle in the evaporator \( Q_i \), work is extracted from the cycle at the turbine \( W_o \) and finally heat is rejected from the cycle at the condenser \( Q_o \). The energy balance at the end of the cycle should sum to zero as energy can neither be created nor destroyed. The desirable result for a cycle is to have a net work production (e.g. \( W_o > W_i \)). In this analysis all components are assumed to be perfectly insulated. Also the friction in the pipe work and heat exchangers is assumed to be negligible for the purposes of this analysis. The pressure drop in the heat exchangers
and in the pipe work in reality is of real concern, but those components can be designed appropriately to minimise pressure drop.

Figure 3. Rankine cycle component layout

**Cycle Operating Conditions**

Upon framing the type of thermodynamic cycle and defining the physical components that will make up the cycle, the cycle operating conditions can be determined. A Rankine cycle will have a set of conditions in which it operates. The conditions can be determined by a wide range of constraints such as environmental, economic, chemical and mechanical.

**Coolant Fluid Temperature \( (T_c) \)**

In this investigation, the focus is geothermal energy, particularly in Queensland, Australia. Accordingly, a constraint taken from the environment is the ambient air temperature of Central Queensland.

<table>
<thead>
<tr>
<th>Month</th>
<th>Monthly Mean Maximum Temperature (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>34.0</td>
</tr>
<tr>
<td>Feb</td>
<td>32.5</td>
</tr>
<tr>
<td>Mar</td>
<td>31.3</td>
</tr>
<tr>
<td>Apr</td>
<td>28.0</td>
</tr>
<tr>
<td>May</td>
<td>23.7</td>
</tr>
<tr>
<td>Jun</td>
<td>20.2</td>
</tr>
<tr>
<td>Jul</td>
<td>20.0</td>
</tr>
<tr>
<td>Aug</td>
<td>22.4</td>
</tr>
<tr>
<td>Sep</td>
<td>26.5</td>
</tr>
<tr>
<td>Oct</td>
<td>29.5</td>
</tr>
<tr>
<td>Nov</td>
<td>31.5</td>
</tr>
<tr>
<td>Dec</td>
<td>32.9</td>
</tr>
</tbody>
</table>

Because water is scarce in Central Queensland, it is assumed that the cooling method for the cycles will be fan-forced air cooling (fan power is calculated and considered a parasitic loss). Average
ambient temperatures in Central Queensland were taken from reported measurements by the Australian Bureau of Meteorology from 1992 to 2011 for the Roma Airport Weather Station (BOM 2012). The ambient temperatures will be used (later in Chapter 6) to investigate the influence of ambient temperature on cycle performance throughout the year.

**Geothermal Brine Temperature (T_h)**

The thermal fluid heat source is assumed to be hot, geothermal brine. EGS source temperatures may vary from 80°C for shallow artesian aquifers (Habermehl 2002) to 250°C in deep heat producing granites in Queensland (Voros 2007).

**Evaporator Pressure (P_2)**

The heating process through the evaporator is assumed to be isobaric. The pressure in the evaporator has a large effect on a power cycle’s performance. It affects the efficiency of heat transfer from the thermal fluid to the working fluid and has a significant effect on the design and performance of the turbomachinery used in the cycle. The influence of evaporator pressure will be discussed in more detail further in this chapter.

**Heat Exchanger Inlet Temperature Differential (EHX_dT for evaporator and CHX_dT for condenser)**

In order for a heat exchanger to operate well, a sufficiently large temperature differential must exist between the fluids. The magnitude of the difference affects the performance of the cycle. (Tchanche 2008) used 15°C, the ORC associated with the Soultz-Sous-Forêts EGS project used a differential of 47.5°C between the brine inlet and the working fluid outlet in the evaporator (Cuenot N. 2008), (Borsukiewicz-Gozdur 2010) assumed 5°C for an analysis of a supercritical ORC. The differential temperature can be a wide range is selected to best optimise a cycle for a given set of operating conditions. The heat exchangers modelled in this analysis are assumed to be counterflow heat exchangers. Figure 4 illustrates the relationship between the fluid temperature profiles in the heat exchangers and T_h, T_c, EHX_dT and CHX_dT.

**Isentropic Pump Efficiency (ETAC)**

The pump efficiency needs to either be calculated or specified. In this analysis, ETAC was specified based on reported average values. A typical value for ETAC used is 75% for pumping saturated liquids (Landgraf 1973), (Dunham & Came 1970).
Isentropic Turbine Efficiency \( (ETAT) \)

The turbine efficiency can either be calculated using loss models or specified based on reported averages. In this chapter of the analysis, the efficiency will be based on 85%. This value was shown to be a characteristic efficiency for a single stage impulse turbine by (Glassman 1994). Landgraf (1973) also used 85% as an assumed turbine efficiency for a study on the choice of working fluid and operating conditions for energy conversion with geothermal heat sources. This is relatively consistent with what (Dipippo 2007) noted as the average efficiencies for turbines in geothermal binary plants 77-82%. In later chapters the efficiency will be calculated based upon an experimentally calibrated turbine loss model.

Power, Gross Work Out \( (W_o) \)

In this analysis, the gross power desired is specified with the ability to sensitise across a range of powers.

Recuperation \( (Rc) \)

In a Rankine cycle, a recuperative heat exchanger may be employed to transfer residual heat in the working fluid at the turbine exit (state 6) to the working fluid at the pump outlet (state 2). The
decision to employ recuperation depends on several conditions; temperature, pressure, and working fluid. The relationship between the recuperator and other components can be seen in Figure 3. In this cycle analysis, the option whether or not to use recuperation is made available. This allows for ascertaining the relative benefit of incorporating a recuperator into a given system.

**Cycle Analysis Calculation Process Algorithm**

Figure 5 shows a simplified algorithm of the calculation process employed in this cycle analysis program. The objective of the model is to enable very large searches over a wide range of conditions to find what set of conditions (working fluid, operating pressures, heat exchanger temperature differentials, recuperation) will yield the best performing cycle for a given set of $T_h$, $T_c$, $ETAC$, $ETAT$ and $W_o$.

![Diagram](image)

Figure 5. Overview of the cycle analysis calculation process (simplified algorithm)
The program begins by specifying a set of inputs to frame the conditions for cycle analyses to be iterated across. Each parameter is specified as an array of minimum to maximum incremented by a specified step size. The inputs are structured as

\[
\text{Inputs} = \begin{bmatrix}
\text{Fluid}_1, \ldots, \text{Fluid}_n \\
T_{h\text{ min}}, \ldots, T_{h\text{ max}} \\
T_{c\text{ min}}, \ldots, T_{c\text{ max}} \\
P_{2\text{ min}}, \ldots, P_{2\text{ max}} \\
EH_{XdT\text{ min}}, \ldots, EH_{XdT\text{ max}} \\
CH_{XdT\text{ min}}, \ldots, CH_{XdT\text{ max}} \\
ETAC_{\text{min}}, \ldots, ETAC_{\text{max}} \\
ETAT_{\text{min}}, \ldots, ETAT_{\text{max}} \\
W_o, \ldots, W_o \\
Re (\text{on} / \text{off})
\end{bmatrix}
\]

From the specified set of inputs, a list of all possible combinations is created for the cycle analysis program to read. Looking at all possible combinations in this manner is computationally expensive, but it allows for a comprehensive investigation of the relationships between many different parameters.

**Cycle State Points**

By specifying the previously mentioned conditions \((T_c, T_h, P_2, EH_{XdT}, CH_{XdT}, ETAC, ETAT, W_o, Re)\), the state points for a specific cycle can be fully described. Here a state point is defined as being a set of thermodynamic properties (pressure \(P\), temperature \(T\), density \(\rho\), enthalpy \(H\), entropy \(S\), and quality \(X\)).

If any two of these properties are known, pressure and temperature for example, then all the remaining properties can be found by using equations of state. In this analysis, REFPROP (NIST) was used to determine fluid properties. REFPROP is a program that uses published equations to calculate thermodynamic and transport properties for many pure fluids (Lemmon E.W. 2007). In this work when REFPROP is used to calculate properties it will be denoted as RP().

**State 1, Pump Inlet**

State 1 is fixed by assuming a saturated liquid \((X_1 = 0)\) at \(T_1\) where

\[
T_1 = T_c + CH_{XdT}
\]

(1)
State 2, Pump Outlet
State 2 is fixed by assuming isentropic compression with a pump efficiency of $ETAC$ to calculate the enthalpy at state 2 ($H_2$) and by the pump outlet pressure being equal to $P_2$. The enthalpy at state 2 is calculated as follows:

$$H_2 = H_1 + (H_{2s} - H_1)/ETAC$$  \hspace{1cm} (2)

where $H_{2s}$ is the enthalpy for ideal isentropic compression, $H_1$ is the enthalpy at the compressor inlet and $ETAC$ is the isentropic pump efficiency.

State 3, Evaporator (Saturated Liquid)
State 3 is fixed by the user-specified evaporator pressure ($P_2$) and a saturated liquid quality ($X_3 = 0$). Heat addition is assumed to be isobaric from the compressor outlet to the turbine inlet (States 2-5).

State 4, Evaporator (Saturated Vapour)
State 4 is fixed by the evaporator pressure ($P_2$) and a saturated vapour quality ($X_4 = 1$). In supercritical cycles however, states 3 and 4 are not present because a supercritical fluid does not have a latent heat of evaporation.

State 5, Turbine Inlet
State point 5 is fixed by $P_2$ and $T_5$, where

$$T_5 = T_h - EHX_{dT}$$  \hspace{1cm} (3)

In the case where there is no superheating, state 5 will be equal to state point 4.

State 6, Turbine Outlet
State 6 is fixed by assuming isentropic expansion with a turbine efficiency of $ETAT$ to calculate the enthalpy at state 6 ($H_6$) and by the turbine outlet pressure being equal to the pressure in state 1. Isobaric condensation is assumed from states 6-1. The enthalpy at state 6 is calculated as follows:

$$H_6 = H_5 - ETAT \left( H_5 - H_{6s} \right)$$  \hspace{1cm} (4)

where $H_6$ is the actual enthalpy and $H_{6s}$ is the enthalpy for ideal isentropic expansion.

State 7, Condenser Inlet
State 7 is fixed by $P_1$ and by assuming that the fluid is a saturated vapour ($X_7 = 1$).
In the case where recuperation is implemented, there are four additional states determined to allow for calculating heat transfer inside the recuperator. Heat transfer inside the recuperator will be discussed later.

**State 8, High-Temperature Fluid Recuperation Inlet**

State 8 is equal to state 6 as there is no process occurring between these two points. The only change to the fluid properties between these two points would be a result of pipe friction. But in this analysis the pipe friction is ignored for simplification. However these points have been separated in the code for the intent of incorporating pipe loss modules into ORCCA in future works. Recuperation is possible as long as the recuperation inlet condition satisfies the recuperation criterion expressed as

\[ T_6 > T_2 + 2 \text{RHX}_{dT} \]  

(5)

where \(\text{RHX}_{dT}\) is the user specified minimum temperature differential between the hot working fluid (states 8-9) and cold working fluid (states 10-11). Figure 6 graphically shows the temperature profiles of the cold and hot working fluid in the recuperator.

So at a minimum, \(T_6\) must be greater than twice \(\text{RHX}_{dT}\) in order to maintain a minimum difference of \(\text{RHX}_{dT}\) between \(T_8\) and \(T_{11}\), and \(T_9\) and \(T_{10}\).

**State 9, High-Temperature Fluid Recuperation Outlet**

State 9 is fixed by \(P_1\) and by a pinch point analysis between the two fluid streams to find \(T_9\). The pinch point will be described in more detail later.
Figure 7. State point fluid properties matrix. RP means value taken from REFPROP, G means given input for a particular cycle.

State 10, Low-Temperature Fluid Recuperation Inlet
State 10 is equal to state 2 so long as the recuperation criterion is satisfied. As with state point 8, pipe losses are being ignored.

State 11, Low-Temperature Fluid Recuperation Outlet
State 11 is fixed by $P_1$ and by $T_{11}$, which are found from the recuperator pinch point analysis.

Figure 7 shows how all the state properties are specified in a matrix format. States 8-11 are only calculated for cycles incorporating recuperation.

5.1 Cycle Calculations
Once all the states are determined for a given cycle, the analysis can be performed. Principle values of interest are heat addition, heat rejection, mass flow rates, thermal efficiencies, work output, work input and thermal fluid effectiveness.
Cycle calculations begin with determining the working fluid mass flow rate. For a given set of cycle operating conditions, \( W_o \) is specified so the working fluid mass flow rate can be found by

\[
m_f = \frac{W_o}{(H_5 - H_6)}
\]  

(6)

where \( m_f \) is the working fluid mass flow rate. After \( m_f \) is defined, then energy accounting can be performed for both mechanical and thermal energy streams; Work In, \( W_i \), at the pump, Heat In, \( Q_i \), at the evaporator, Work Out, \( W_o \), at the turbine, and Heat Out, \( Q_o \), at the condenser.

**Pump**

Corresponding to the state point numbering on Figure 3, the cycle begins at the inlet of the pump. Saturated liquid enters the pump (state 1) and is compressed to a higher pressure (state 2). The work done by the pump is calculated as

\[
W_i = m_f (H_2 - H_1)
\]  

(7)

where \( W_i \) is the pump work input.

In the actual process there are irreversibilities and losses. The actual process through the pump is not isentropic. Figure 2 shows the actual process represented by the line 1-2. The work input from 1-2 is greater than the idealised isentropic process 1-2s. The isentropic pump efficiency is expressed as

\[
ETAC = \frac{H_{2s} - H_1}{H_2 - H_1}
\]  

(8)

**Evaporator**

After the fluid leaves the pump, it enters the evaporator. The total amount of heat added to the working fluid is calculated as

\[
Q_i = m_f (H_5 - H_2)
\]  

(9)

**Turbine**

After the fluid has been heated in the heat exchangers it then enters the turbine. It can either enter as a saturated vapour, superheated vapour, or supercritical fluid. There are cases where the fluid can have a quality less than 1, but this study will only focus on turbine inlet conditions where the fluid quality is greater than or equal to 1.

The work output of the turbine is calculated as
\[ W_o = m_f (H_5 - H_6) \]  

where \( W_o \) is the gross turbine work output. As with the pump, the turbine experiences irreversibilities and losses and the actual process is not isentropic. The actual process is represented by the line 5-6 in Figure 2. The work output for 5-6 is less than the idealised isentropic process 5-6s. The isentropic turbine efficiency, ETAT, is expressed as

\[ ETAT = \frac{H_5 - H_6}{H_5 - H_{6s}} \]  

Condenser

For the pump to operate efficiently, the exhausted fluid from the turbine, usually in the form of a superheated vapour or a saturated vapour, needs to be cooled and condensed to a saturated liquid. The fluid cools from a superheated vapour to a saturated liquid (6-7) and then condenses from a saturated vapour to a saturated liquid (7-1). The total amount of heat rejected, \( Q_o \), is expressed as

\[ Q_o = m_f (H_6 - H_1) \]  

In this study, we consider the cooling method to be fan-forced convection, which carries a work penalty. In large industrial fans, gas density through the fan does not usually exceed 7% so a simplified incompressible flow analysis is used to calculate fan work (Munson 1998). The cooling work, \( W_c \), is the work required to fan cool the cycle and is defined as

\[ W_c = \left( \frac{m_c}{\rho_c} \right) P_{fan} \eta_c \]  

Where \( m_c \) is the coolant mass flow rate, \( \rho_c \) is the cooling fluid average density, \( P_{fan} \) is the pressure increase across the fan, \( \eta_c \) is cooling fluid pump efficiency (i.e. fan for air-cooled system).

Net Work

\( W_{net} \) is the net work available from the cycle (kW) and defined as

\[ W_{net} = W_o - W_i - W_c \]  

Evaporator Pinch Analysis

To determine the thermal fluid mass flow rate (\( m_h \)), a pinch point analysis is performed. A standard pinch analysis will look at temperature differences between two fluid streams through a heat exchanger. It allows for determining the inlet and outlet temperatures of the two fluid streams such that their temperature profiles do not intersect. At any point in the evaporator, the hot thermal fluid stream must always have a higher temperature than the cold working fluid stream to ensure heat is
being transferred from hot to cold fluid. If these two temperature profiles intersect or coincide, then no further heat transfer occurs.

To begin a pinch analysis in the evaporator, the total amount of heat to be transferred is determined by

\[ Q_i = m_f (H_5 - H_2), \text{ w/o Recuperation} \]

\[ Q_i = m_f (H_5 - H_{11}), \text{ w/ Recuperation} \]

The total amount of heat is divided into \( N \) number of segments, representing finite amounts of heat \( (q_N) \)

\[ q_N = \frac{Q_i}{N} \]

that are transferred over regular intervals. Figure 8 represents the relationship between the two temperature profiles and the heat exchanger length.

The pinch analysis is an iterative process. A general pinch algorithm, described below, is employed that works for both subcritical and supercritical cases. The iterative term in this case is the outlet temperature of the thermal fluid \( (T_{ho}) \). The initial guess for \( T_{ho} \) is

\[ T_{ho}(i = 1) = T_2 \]

And, based on an initial guess of \( T_{ho} \), an initial \( m_h \) is calculated as

\[ m_h = \frac{Q_i}{c_h (T_{hi} - T_{ho}(i = 1))} \]

where \( c_h \) is the heat capacity of the hot thermal fluid and \( T_{hi} \) is its inlet temperature. In this case, \( T_{hi} \) is equal to the temperature of the geothermal fluid. Then an iteration of \( n \) equal to 1 to \( N \) begins by calculating
\[ T_h(n) = T_{hi} - (n - 1) \frac{q_N}{m_h c_h} \]  
where \( T_h(n) \) is the temperature of the hot fluid at point \( n \) along the length of the heat exchanger and \( T_f(n) \) is the temperature of the working fluid at point \( n \) found by

\[ T_f(n) = RP(P_2, H_f(n)) \]

The enthalpy \( H_f(n) \) is found by

\[ H_f(n) = H_5 - (n - 1)q_N \text{ mf} \]

The iteration continues until

\[ T_h(n) - T_f(n) < \Delta T_{\text{pinch}} \]

where \( \Delta T_{\text{pinch}} \) is the minimum allowable temperature differential at any point in the heat exchanger. If the above criterion in Eq. 22 is true, then the iteration is started over with a new \( T_{ho} \) (i.e. \( T_{ho} = T_{ho} + i \)). The iterative process continues until \( n \) is equal to \( N \). When \( n \) reaches \( N \) without the above criterion being true, then a successful pinch analysis has been completed. In a successful pinch analysis, the outlet temperature of the geothermal brine from the heat exchanger is then

\[ T_{ho} = T_h(n = N) \]

And \( m_h \) for \( T_{ho} \) in the successful pinch analysis is calculated as

\[ m_h = \frac{Q_i}{c_h (T_{hi} - T_{ho})} \]

**Condenser Pinch Analysis**

Similarly, the coolant mass flow rate and outlet temperature need to be determined. Another pinch analysis is performed for the condenser.

To begin a pinch analysis in the condenser, the total amount of heat to be transferred is determined from

\[ Q_o = m_f(H_6 - H_1) \text{, w/o Recuperation} \]

\[ Q_o = m_f(H_9 - H_1) \text{, w/ Recuperation} \]

Then the total amount of heat is again subdivided into \( N \) number of segments.

\[ q_N = \frac{Q_o}{N} \]

Figure 9 represents the relationship between the two fluid temperature profiles and the heat exchanger length.
Again, the process is iterative. The initial guess for $T_{co}$ is

$$T_{co}(i = 1) = T_6$$ (27)

And based on an initial guess from $T_{co}$, $m_c$ is calculated as

$$m_c = \frac{Q_o}{c_c (T_{ci} - T_{co}(i = 1))}$$ (28)

where $c_c$ is the heat capacity of the coolant and $T_{ci}$ is the inlet temperature of the coolant. In this case, $T_{ci}$ is equal to the ambient air temperature. Then, an iteration ranging from 0 to $N$ begins by calculating the following values

$$T_c(n) = T_{ci} + (n - 1) \frac{q_N}{m_c c_c}$$ (29)

where $T_c$ is the temperature of the coolant at point $n$ and $T_f$ is the temperature of the working fluid at point $n$ found by

$$T_f(n) = RP(P_1, H_f(n))$$ (30)

The enthalpy $H_f$ is found by

$$H_f(n) = H_1 + (n - 1)q_N m_f$$ (31)

The iteration continues until

$$T_f(n) - T_c(n) < \Delta T_{pinch}$$ (32)

then restarts with a new $T_{co}$ (i.e. $T_{co} = T_{co} - i$). The process continues until $n$ is equal to $N$. When $n$ reaches $N$ without Eq. 32 being satisfied, a successful pinch analysis has been completed. The outlet temperature of the coolant from the condenser is
\[ T_{co} = T_c (n = N) \]  
(33)

and \( m_c \) for \( T_{co} \) in a successful pinch analysis is

\[ m_c = \frac{Q_o}{c_c (T_{ci} - T_{co})} \]  
(34)

Figure 10 Shows the flow charts for both the evaporator pinch analysis and condenser pinch analysis processes.

**Recuperator Pinch Analysis**

When recuperation is employed, a pinch analysis is performed on the recuperator to determine the working fluid exit temperatures (\( T_9 \) and \( T_{11} \)). The pinch analysis is similar to that of the evaporator and condenser with the difference of the heat transfer being calculated in each iteration of the outlet temperature loop (\( T_9 \) Loop). In the evaporator and condenser pinch analysis loops the heat transfer
is calculated outside the temperature iteration loop based upon the inputs of the cycle state points and working fluid mass flow rate. Figure 11 shows the flow chart for the recuperator pinch analysis process.

![Figure 11. Recuperator pinch analysis process flow chart](image)

**Efficiencies**

To evaluate and rank cycles, metrics need to be calculated. Two useful metrics are the first law of thermal efficiency and the thermal fluid effectiveness.

The first law of thermal efficiency is a useful expression that quantifies the extent to which the heat added, $Q_i$, is converted into mechanical energy, $W_o$. The first law of efficiency is expressed as

$$n_{1st} = \frac{W_{net}}{Q_i}$$  \hspace{1cm} (35)
Bearing in mind that this thesis is focused on EGS, an even more useful metric is the “thermal fluid effectiveness”. Thermal fluid effectiveness is useful in quantifying how effectively a cycle utilises the energy in a thermal fluid stream. Thermal fluid effectiveness, $\beta$, for the cycle is expressed as

$$\beta = \frac{W_{\text{net}}}{m_h}$$

(36)

where $m_h$ is the thermal fluid mass flow rate. $\beta$ can be related to $n_{1st}$ by substituting Eq. 24 into Eq. 36 and then substituting this into Eq. 35. The relationship between $\beta$ and $n_{1st}$ can be expressed as

$$\beta = n_{1st} c_h (T_{hi} - T_{ho})$$

(37)

This shows that high values of $\beta$ can be achieved even at low values of $n_{1st}$ if the temperature drop in the geothermal fluid is high enough.

When the primary objective is to make the most efficient use of the energy in the thermal fluid stream, the largest possible $\beta$ value is the objective. Large $\beta$ values can be interpreted as a large energy output for a small thermal fluid mass flow input. When capital costs (such as those for wells) dominate, this parameter is a very good measure of cycle’s performance. It’s true that situations where a high $\beta$ may exist with a low $n_{1st}$ value will likely have a high working fluid mass flow rates. High working fluid mass flow rates can lead to large and expensive surface equipment but if the cost of subsurface development is much greater than surface facilities then a larger $\beta$ may be desirable at the expense of a lower $n_{1st}$.

$\beta$ is useful for assessing cycles for use with EGS because it readily relates a wells brine production rates to power generation. In EGS applications, subsurface costs (i.e. drilling, completions and stimulation) are substantial so minimising the required brine mass flow rate and maximising energy production are primary design objectives. $\beta$ is useful because it shows a direct relationship between energy and mass flow rate. $\beta$ was adopted from (Franco A. 2009) who stated that “The parameter is often considered when the minimization of geothermal fluid flow rate (specific consumption) for a given power is suggested as an objective function for optimal design”.

Pipe and Heat Exchangers

Losses due to friction in the pipe work and the heat exchangers can be a major source of losses in an ORC. A pipe friction module and a heat exchanger friction module were written to be incorporated into ORCCA. After running the model using these modules for some time it was decided that the pipe and heat exchanger losses detracted from the objective of investigating the incorporation of a turbine loss model into a cycle analysis. The main limitation for pipe and heat
exchanger friction is pipe and heat exchanger size, and size is in practical terms, limited by a monetary constraints. Because monetary constraints cannot be easily defined and generalised, pipe and heat exchanger losses were omitted from this analysis.

However, in the practical design of an ORC, pipe and heat exchanger design are a crucial part of the design process.
Chapter 6 Application of Cycle Analysis

Many simulations were run using the cycle analysis software developed in this thesis, ORCCA. Some of the conditions investigated are listed in Table 4. The fluids analysed are a compilation of fluids listed by other authors’ published papers on the topic of ORC design and optimisation (Tchanche 2008), (Saleh B. 2007), (Franco A. 2009). The majority of fluids that receive the most attention for low temperature ORC’s are typically Hydrochlorofluorocarbons (HCFC), halocarbons, or hydrocarbons. These fluids tend to have critical points that lie in between the heat source temperature and the cooling source temperature.

<table>
<thead>
<tr>
<th>Fluid</th>
<th>T(_{\text{critical}}) (°C)</th>
<th>P(_{\text{critical}}) (kPa)</th>
<th>CAS No</th>
<th>Molecular Weight (kg/mol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4F10</td>
<td>113.18</td>
<td>2323.40</td>
<td>355-25-9</td>
<td>238.03</td>
</tr>
<tr>
<td>C5F12</td>
<td>147.41</td>
<td>2045.00</td>
<td>678-26-2</td>
<td>288.03</td>
</tr>
<tr>
<td>CF3I</td>
<td>123.29</td>
<td>3953.00</td>
<td>2314-97-8</td>
<td>195.91</td>
</tr>
<tr>
<td>Pentane</td>
<td>196.55</td>
<td>3370.00</td>
<td>109-66-0</td>
<td>72.15</td>
</tr>
<tr>
<td>Propane</td>
<td>96.74</td>
<td>4251.20</td>
<td>74-98-6</td>
<td>44.10</td>
</tr>
<tr>
<td>R124</td>
<td>122.28</td>
<td>3624.30</td>
<td>2837-89-0</td>
<td>136.48</td>
</tr>
<tr>
<td>CF3I</td>
<td>66.02</td>
<td>3617.70</td>
<td>2314-97-8</td>
<td>120.02</td>
</tr>
<tr>
<td>C4F10</td>
<td>101.06</td>
<td>4059.28</td>
<td>355-25-9</td>
<td>102.03</td>
</tr>
<tr>
<td>R143a</td>
<td>72.71</td>
<td>3761.00</td>
<td>420-46-2</td>
<td>84.04</td>
</tr>
<tr>
<td>R152a</td>
<td>113.26</td>
<td>4516.75</td>
<td>75-37-6</td>
<td>66.05</td>
</tr>
<tr>
<td>R227ea</td>
<td>101.75</td>
<td>2925.00</td>
<td>431-89-0</td>
<td>170.03</td>
</tr>
<tr>
<td>R236ea</td>
<td>139.29</td>
<td>3501.98</td>
<td>431-63-0</td>
<td>152.04</td>
</tr>
<tr>
<td>R245fa</td>
<td>154.01</td>
<td>3651.00</td>
<td>460-73-1</td>
<td>134.05</td>
</tr>
<tr>
<td>RC318</td>
<td>115.23</td>
<td>2777.50</td>
<td>678-26-2</td>
<td>200.03</td>
</tr>
</tbody>
</table>

Table 5 lists the range of input conditions for which the analysis presented in this chapter was conducted. The temperature ranges are representative of the potential geothermal conditions in Queensland. The assumed efficiencies are representative of average reported values of pumps and turbines. The power is selected so as to be representative of small rural geothermal power stations. Small rural power stations were focused on because they are a promising geothermal development area in Queensland. There are thousands of producing water bores in rural Queensland that are producing water upwards of 100°C (Swenson 2000) and with simple solar boosting mechanisms or drilling deeper wells higher temperatures can be achieved.

The range of values listed above was divided into evenly spaced arrays of the stated number of increments. For the list of conditions above and the fluids investigated this resulted in 6,720,000 unique scenarios to investigate. Parallel processing was employed across 8 CPU’s to speed up the
investigation. At an average calculation time of 0.002 seconds the simulation can be run in less than an hour.

<table>
<thead>
<tr>
<th>Cycle Scenario</th>
<th>Value Range</th>
<th>No. Increment</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_h$</td>
<td>110-170</td>
<td>30</td>
<td>°C</td>
</tr>
<tr>
<td>$T_c$</td>
<td>20-40</td>
<td>10</td>
<td>°C</td>
</tr>
<tr>
<td>$P_2$</td>
<td>200 - 2 x $P_c$</td>
<td>50</td>
<td>kPa</td>
</tr>
<tr>
<td>$EHX_{dr}$</td>
<td>5-20</td>
<td>4</td>
<td>°C</td>
</tr>
<tr>
<td>$CHX_{dr}$</td>
<td>5-20</td>
<td>4</td>
<td>°C</td>
</tr>
<tr>
<td>$ETAC$</td>
<td>75%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$ETAT$</td>
<td>85%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$W_o$</td>
<td>100</td>
<td>1</td>
<td>kW</td>
</tr>
</tbody>
</table>

### 6.1 Cycle Analysis Results

For each of the 6,720,000 scenarios analysed the critical cycle parameters were calculated as described in Chapter 5. The results were processed to extract the best operating conditions ($P_2$, $EHX_{dr}$, $CHX_{dr}$, $m_f$, $m_{fh}$, $m_{fc}$, whether or not to employ recuperation) based on producing the highest $\beta$ possible. The results of the best performing cycles for each fluid are summarised in Table 6.

<table>
<thead>
<tr>
<th>Fluid</th>
<th>$T_h$ ($°C$)</th>
<th>$T_c$ ($°C$)</th>
<th>Max $\beta$ (kJ/kg)</th>
<th>$\eta_{1st}$</th>
<th>$P_2$ (kPa)</th>
<th>$m_f$ (kg/s)</th>
<th>$m_{fh}$ (kg/s)</th>
<th>$m_{fc}$ (kg/s)</th>
<th>$EHX_{dr}$ ($°C$)</th>
<th>$CHX_{dr}$ ($°C$)</th>
<th>Recuperation</th>
</tr>
</thead>
<tbody>
<tr>
<td>R134a</td>
<td>170</td>
<td>20</td>
<td>48</td>
<td>10%</td>
<td>7,676</td>
<td>2.74</td>
<td>1.27</td>
<td>22.78</td>
<td>10</td>
<td>20</td>
<td>Off</td>
</tr>
<tr>
<td>R124</td>
<td>170</td>
<td>20</td>
<td>48</td>
<td>11%</td>
<td>4,846</td>
<td>3.55</td>
<td>1.35</td>
<td>25.29</td>
<td>20</td>
<td>20</td>
<td>On</td>
</tr>
<tr>
<td>RC318</td>
<td>170</td>
<td>20</td>
<td>47</td>
<td>12%</td>
<td>6,027</td>
<td>4.29</td>
<td>1.28</td>
<td>22.33</td>
<td>10</td>
<td>20</td>
<td>Off</td>
</tr>
<tr>
<td>C5F12</td>
<td>170</td>
<td>20</td>
<td>47</td>
<td>14%</td>
<td>2,297</td>
<td>4.86</td>
<td>1.55</td>
<td>22.35</td>
<td>15</td>
<td>20</td>
<td>On</td>
</tr>
<tr>
<td>R227ea</td>
<td>170</td>
<td>20</td>
<td>47</td>
<td>9%</td>
<td>5,865</td>
<td>3.67</td>
<td>1.27</td>
<td>26.70</td>
<td>10</td>
<td>15</td>
<td>Off</td>
</tr>
<tr>
<td>R143a</td>
<td>170</td>
<td>20</td>
<td>47</td>
<td>9%</td>
<td>8,166</td>
<td>2.50</td>
<td>1.21</td>
<td>24.54</td>
<td>5</td>
<td>15</td>
<td>Off</td>
</tr>
<tr>
<td>Propane</td>
<td>170</td>
<td>20</td>
<td>46</td>
<td>12%</td>
<td>6,915</td>
<td>1.21</td>
<td>1.27</td>
<td>28.03</td>
<td>5</td>
<td>15</td>
<td>On</td>
</tr>
<tr>
<td>R152a</td>
<td>170</td>
<td>20</td>
<td>46</td>
<td>11%</td>
<td>5,301</td>
<td>1.87</td>
<td>1.47</td>
<td>24.21</td>
<td>15</td>
<td>20</td>
<td>Off</td>
</tr>
<tr>
<td>R236ea</td>
<td>170</td>
<td>20</td>
<td>46</td>
<td>11%</td>
<td>4,179</td>
<td>3.16</td>
<td>1.51</td>
<td>23.99</td>
<td>15</td>
<td>20</td>
<td>Off</td>
</tr>
<tr>
<td>C4F10</td>
<td>170</td>
<td>20</td>
<td>45</td>
<td>9%</td>
<td>4,493</td>
<td>4.52</td>
<td>1.39</td>
<td>25.82</td>
<td>10</td>
<td>15</td>
<td>Off</td>
</tr>
<tr>
<td>R125</td>
<td>170</td>
<td>20</td>
<td>44</td>
<td>12%</td>
<td>9,671</td>
<td>3.33</td>
<td>1.18</td>
<td>23.31</td>
<td>5</td>
<td>15</td>
<td>On</td>
</tr>
<tr>
<td>R245fa</td>
<td>170</td>
<td>20</td>
<td>40</td>
<td>11%</td>
<td>2,111</td>
<td>2.60</td>
<td>1.92</td>
<td>23.56</td>
<td>20</td>
<td>20</td>
<td>Off</td>
</tr>
<tr>
<td>CF3I</td>
<td>166</td>
<td>20</td>
<td>39</td>
<td>11%</td>
<td>3,692</td>
<td>6.03</td>
<td>1.69</td>
<td>26.78</td>
<td>20</td>
<td>20</td>
<td>On</td>
</tr>
<tr>
<td>Pentane</td>
<td>170</td>
<td>20</td>
<td>37</td>
<td>11%</td>
<td>1,082</td>
<td>1.18</td>
<td>1.99</td>
<td>30.08</td>
<td>20</td>
<td>15</td>
<td>Off</td>
</tr>
</tbody>
</table>

Table 6 shows the operating conditions that produce the cycle which yields the highest $\beta$ for each fluid investigated. There are a range of pressures, heat exchanger temperature differentials, and recuperation employment, but all the fluids produce their best cycle when the temperature differential is the greatest. That largest temperature differential is intuitive based on the Carnot principle, but the operating conditions are not as easily predicted.
6.1.1 Top Fluids vs. Benchmark Fluids (R245fa/Pentane)

In practice, there are a few fluids commonly utilised as working fluids (i.e. R245fa, Pentane, R134a) (Quoilin 2009). There could be many reasons why other fluids are not selected as working fluids. They may be too expensive, there could be compatibility issues between equipment and fluids or there may be environmental or regulatory constraints that prohibit a fluids use.

A closer look at the results of this analysis are presented as the top performing fluids (based on maximum $\beta$) for four selected $T_h$ values in a comparison with benchmark fluids. For a specific $T_h$, there will be an associated optimum fluid and set of operating conditions.

![Figure 12. Top fluids vs. benchmark fluids, R245fa and pentane](image)

The potential gains to be realised for tailoring a cycle to the specific conditions of a site are more evident when comparing the top performing fluid, for a given set of $T_h$ and $T_c$, against the performance of benchmark fluids like R245fa and pentane. Figure 12 shows the top performing fluids ranked by $\beta$ at several source temperatures as well as the corresponding $\beta$ for both R245fa and pentane for comparison.

It is clear there is a significant potential for gains in performance by tailoring a cycle to the specific conditions for a given site. By selecting the fluid and operating conditions appropriately, the thermal resource can be utilised much more effectively.

6.1.2 Influence of ETAT

Figure 13 shows an example of the relationship between $\beta$ and ETAT for a set of cycles run for R245fa with $T_{ha}$ equal to 145°C and $T_{ca}$ equal to 22°C and the evaporator pressure equal to 1800kPa but with a varied ETAT.

As ETAT is increased incrementally, $\beta$ increases incrementally in a linear relationship. At large ETAT values the incremental change in $\beta$ is relatively small as a percentage to the magnitude of $\beta$. 45
Where at smaller value of $ETAT$ the incremental change in $\beta$ is large as a percentage of the magnitude of $\beta$.

Figure 13. $ETAT$ versus $\beta$. This plot shows the relationship between $ETAT$ and $\beta$.

This would suggest that the importance of accurate $ETAT$ values are more and more important when dealing with operating conditions that will yield lower turbine efficiencies. In smaller power systems that have lower mass flow rates a partial admission turbine may be required and the efficiencies are likely to be lower, thus more accurate ETAT values would be desirable for conducting a more accurate cycle analysis.

6.1.3 Influence of Evaporator Pressure

The evaporator pressure is one of the most influential parameters on the performance of a cycle. It has a major influence on the performance of the pump, evaporator and turbine. The evaporator pressure’s influence is most prominent on $\beta$ as a result of the pinch point. The effects of evaporator pressure on the system performance has been published by other authors, however, the conditions for which it was shown were limited and did not include supercritical cases and the influence of recuperation (Tchanche 2008) (Chen 2012).

For each cycle there is an optimum evaporator pressure. Figure 14 shows a plot of $\beta$ versus evaporator pressure (ranging from subcritical to supercritical) for $T_{hi}$ equal to 150°C (left) and 170°C (right) and $T_{ci}$ equal to 20°C for a cycle using RC318 as the working fluid.

This figure shows that there is a unique optimum evaporator pressure. This relationship between $\beta$ and evaporator pressure was seen for all the fluids investigated in this thesis. This has major implications on the selection of a working fluid and the design of plant equipment.
Certain pressures allow more heat to be extracted from the thermal fluid by matching more closely the temperature glide of the thermal fluid to the working fluid. When the temperature profiles of the working fluid and the geothermal brine match closely in the evaporator the effects of the pinch point are minimised and the temperature of the geothermal brine is able to be drawn down further (i.e. $Q_i = m_h c_h \Delta T_h$). By drawing down the geothermal brine temperature more, $\Delta T$ is maximised and therefore $Q$ is maximised.

As an example of how matching the temperature profiles helps to maximise draw down of the geothermal brine, Figure 15 shows the T-S diagrams of two RC318 cycles. The cycle on the left has an evaporator pressure of 1,000 kPa and the cycle on the right has an evaporator pressure of 1,600 kPa.

The lower pressure cycle is able to extract more heat from the thermal source as seen by the lower value of $T_{ho}$, nearly 20°C less. The lower pressure cycle is able to draw down the geothermal brine...
to nearly 60°C whereas the higher pressure cycle is only able to draw down the geothermal brine to 80°C. In the higher pressure cycle state point 3 is the pinch point and greatly limits the degree to which heat can be extracted from the brine. The optimum pressures shown in Figure 14 are a result of this difference in heat extraction capability which is a consequence of the pinch point in the evaporator.

Also note in Figure 14 that there are no pressures beyond ~5,500kPa for the curve representing $T_{ha}$ of 150°C. This is because as the evaporator pressure increases at this temperature the turbine inlet condition is moved closer and closer towards the vapour dome. At around 5,500kPa, the turbine inlet conditions are moved so close to the vapour dome that gas expanding through the turbine expands through vapour dome (on a T-S diagram) and has the potential to create droplets in the turbine. This can be detrimental to turbine performance and machinery operating life.

Figure 16. An example of a cycle that has gas expanding through the vapour dome. The cycle is an RC318 at 6,000kPa with $T_h$ equal to 150°C and $T_c$ equal to 20°C. This figure relates to the missing optimum shown in Figure 14.

Figure 16 shows an RC318 cycle at 6,000kPa for $T_{ha}$ equal to 147°C. The expansion that takes place from points 5 to 6 through the turbine passes through the vapour dome. It is technically possible to have these types of cycles, but it was decided to filter out cycles like this because of the unknown effects droplets may have on turbine performance. ORCCA had been programed to not allow cycles like these in its analysis. However it should be noted that these cycles are of interest and should be investigated. Work on expanders that can operate efficiently in wet expansion could be very beneficial to ORC technology.
6.1.4 Influence of Recuperation

There is also a significant effect on the optimum pressure by the employment of recuperation. Recuperation is often only thought of as a means to minimise heat transfer requirements in a cycle. But it can also have a profound effect on the design of the components in the entire system. By incorporating a recuperator into the system, improved \( \beta \) values can be achieved for certain conditions. This is shown in Figure 14 and is particularly prominent in the curve for \( T_h \) equal to 170°C. It also shows that by incorporating recuperation a source site with a source temperature of 150°C (\( \beta = 18 \text{kJ/kg} \)) would out perform a site with a source temperature of 170°C (\( \beta = 17 \text{kJ/kg} \)) that doesn’t employ recuperation for evaporator pressures below 1,000 kPa. In a situation where the capital cost of a power station is paramount, it may be an attractive solution to drill shallower wells (lower temperature source) and employ a low pressure, lower temperature recuperated cycle.

The influence of recuperation on \( \beta \) can be seen analytically by looking at what constitutes \( \beta \) in equation 36 and 38, where \( m_h \) is expressed as

\[
m_h = \frac{(Q_i - Q_{hx})}{c_h(T_{hi} - T_{ho})}
\]

Here \( Q_{hx} \) is the amount of heat exchanged in the recuperator. In cycles with sufficient \( Q_{hx} \) to initiate evaporation of the working fluid at the recuperator exit, higher values of \( \beta \) can be achieved. In these cycles, the ratio of heat addition to the change in thermal fluid temperature (\( T_{hi} - T_{ho} \)) is decreased thereby decreasing \( m_h \). For equal \( W_{net} \) values, a lower \( m_h \) will result in a larger \( \beta \) value.

Figure 17. TS-diagram for non-recuperated (left) and recuperated (right) RC318 cycle with 1,350kPa evaporator pressure

Figure 17 shows the T-S diagrams for two scenarios, one without recuperation and one with recuperation for RC318 cycles with \( T_h \) equal to 170°C and evaporator pressure of 1,350 kPa.
In the recuperated cycle the “x’s” indicate the portion of the heat that is exchanged in the recuperator from the hot fluid at the turbine exit to the cold fluid at the pump outlet. In this example, the fluid leaving the recuperator has begun to evaporate. This allows for a steeper thermal-fluid cooling temperature profile in the evaporator and results in a lower $m_h$, which in turn results in a larger $\beta$.

This has significant implications on cycle analysis and system design for geothermal applications. When the key objective is to utilise the thermal fluid as effectively as possible there are three options; binary fluid, supercritical cycle or low pressure recuperation. The option of a binary fluid can be employed as a Kalina cycle, but this makes the system more complicated. Supercritical cycles offer good performance but at the cost of having to use higher pressure equipment which can dramatically increase the cost of equipment. A lower pressure recuperated cycle has the advantage of a simple working fluid, lower pressure equipment, and good thermal fluid effectiveness.

### 6.1.5 Influence of $EHX_{dT}$ and $CHX_{dT}$

The designed inlet temperature differential between fluids in the heat exchangers is critical. The results from an analysis of RC318 over a range of $T_c$, $T_h$, $P_2$, $EHX_{dT}$ and $CHX_{dT}$ are presented to illustrate the variations in $EHX_{dT}$ and $CHX_{dT}$ for optimum cycle performance for varying temperature conditions.

![Figure 18. $\beta$ versus $EHX_{dT}$ (left) and $CHX_{dT}$ (right) for RC318 at 1,000kPa](image)

Figure 18 shows the relationship between $EHX_{dT}$ and $\beta$ for various $T_h$ (using 1,000kPa as the evaporator pressure). This plot shows that the smallest value of $EHX_{dT}$ is best and that the employment of recuperation can have a significant effect on the $EHX_{dT}$ selected for a cycle.
Similarly, $CHX_{dt}$ has an impact on $\beta$ and is related to recuperation. The right hand plot shows the influence of $CHX_{dt}$ on $\beta$ for several different $T_h$ ranges for RC318 cycles (using 1,000kPa as the evaporator pressure).

These two examples are for a single pressure. What’s interesting is to look at $EHX_{dt}$ and $CHX_{dt}$ over a range of pressures. When the simulation is run over a range of pressures (rather than a single pressure) there is a comparable solution (in terms of the achievable $\beta$) for the range of $EHX_{dt}$. Figure 19 shows this relationship. The left hand plot showing that for a range of $EHX_{dt}$ a comparable value of $\beta$ can be achieved. To maintain the value of $\beta$ though with a changing $EHX_{dt}$ the pressure changes. The right hand plot shows that relationship between $EHX_{dt}$ and $P_2$.

Recall the $\beta$ versus $P_2$ plot in Figure 14. The portion of the plot from ~4,000kPa to ~7,000kPa remains relatively constant as $P_2$ varies. Notice in the $EHX_{dt}$ versus $P_2$ plot how $EHX_{dt}$ is changing significantly while $\beta$ remains constant. As the pressure is increasing the pinch point is diminishing and that allows $EHX_{dt}$ to decrease. The critical pressure of RC318 is 2,776kPa and it is above the critical pressure where $EHX_{dt}$ can begin to be reduced as temperature profiles between the thermal fluid and the working fluid begin to match more closely.

The relationship between $CHX_{dt}$ shows a very different behaviour (Figure 20). There is a clear optimum for $CHX_{dt}$ with respect to $\beta$. The large change in $CHX_{dt}$ with respect to $P_2$ also begins at the same pressure the $EHX_{dt}$ begins to change dramatically, but $CHX_{dt}$ is increasing rather than decreasing. As $EHX_{dt}$ decreases there is more available enthalpy across the turbine because the turbine inlet temperature is higher. However there is an overall net benefit to sacrifice some of that available enthalpy across the turbine by increasing $CHX_{dt}$ which reduces the coolant mass flow rate and thus the cooling work. The increase in $CHX_{dt}$ does increase the turbine outlet pressure and thus
reduces the available enthalpy that could be converted into work at the turbine, but as an overall system, the increased $CHX_{dT}$ maximises the net work produced by unit mass of geothermal fluid produced.

![Figure 20. $\beta$ versus $CHX_{dT}$ (left) and $CHX_{dT}$ versus $P_2$ (right) for RC318](image)

The effects of heat exchanger temperature differential are more pronounced in the condenser because it will influence the pressure differential across the turbine and will also dictate the mass flow rate of the cooling air and therefore the parasitic losses of cooling fans. There is a much clearer optimum $CHX_{dT}$ for each $T_h$ value and the influence of recuperation on $CHX_{dT}$ increases with $T_h$.

### 6.1.6 Annual Considerations

In an air-cooled cycle, $\beta$ is influenced by ambient temperature therefore $\beta$ will not be constant throughout the year. It is important to consider the influence of annual fluctuations in $T_c$ on a cycle’s performance. Average ambient temperatures in Central Queensland were taken from reported measurements by the Australian Bureau of Meteorology from 1992 to 2011 for the Roma Airport Weather Station (BOM 2012). The analysis was conducted by setting $T_c$ equal to the temperatures for each month listed in the Table 7. For each month’s $T_c$ throughout the year, the optimum power and operating conditions were determined. And for each month, based on the relevant $T_c$ and operating conditions, the total power was found assuming 24 hour power plant operation.

Figure 21 and Figure 22 show the power and optimum operating conditions throughout the year for a cycle utilising RC318 with a geothermal source temperature of 120°C, thermal fluid mass flow rate of 25kg/s, and assuming fan forced cooling.
Intuitively in hot months, the cycles’ power output is at a minimum and in cold months, the power output is at a maximum. In hot months there are increased parasitic losses for the cooling fans and also the pressure ratio across the turbine decreases as the condenser pressure increases with increasing air temperature.

<table>
<thead>
<tr>
<th>Month</th>
<th>Monthly Mean Maximum Temperature (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>34.0</td>
</tr>
<tr>
<td>Feb</td>
<td>32.5</td>
</tr>
<tr>
<td>Mar</td>
<td>31.3</td>
</tr>
<tr>
<td>Apr</td>
<td>28.0</td>
</tr>
<tr>
<td>May</td>
<td>23.7</td>
</tr>
<tr>
<td>Jun</td>
<td>20.2</td>
</tr>
<tr>
<td>Jul</td>
<td>20.0</td>
</tr>
<tr>
<td>Aug</td>
<td>22.4</td>
</tr>
<tr>
<td>Sep</td>
<td>26.5</td>
</tr>
<tr>
<td>Oct</td>
<td>29.5</td>
</tr>
<tr>
<td>Nov</td>
<td>31.5</td>
</tr>
<tr>
<td>Dec</td>
<td>32.9</td>
</tr>
</tbody>
</table>

The trickle-down effect of increased coolant temperatures is symbolically expressed as follows;

\[ T_c \uparrow, P_6 \uparrow, \Delta P_{5-6} \downarrow, \Delta H_{5-6} \downarrow, m_f \uparrow, m_h \uparrow, \beta \downarrow \]

As the coolant temperature increases so does the temperature of the working fluid in the condenser. Increased temperature of the working fluid in the condenser will raise the pressure of the working fluid in the condenser thereby raise the pressure at the outlet of the turbine. By increasing the pressure at the turbine’s outlet, the pressure differential across the turbine decreases and thus the available enthalpy across the turbine decreases. In order to maintain the same power with less available enthalpy, the working fluid mass flow rate must increase. When the working fluid mass flow rate increases the geothermal brine mass flow rate will need to increase in order to maintain
the same working fluid temperature at the evaporator outlet/turbine inlet. If the geothermal brine mass flow rate didn’t increase along with the working fluid mass flow rate, the turbine inlet temperature would decrease and further reduce the available enthalpy. All of these changes occur in order to maintain the same power output while compensating for the increased coolant temperature. In doing so $\beta$ has decreased because power has remained constant but the geothermal brine mass flow rate has increased. Ultimately this leads to a lower utilisation of the geothermal brines energy.

![Figure 22. Annual optimum heat exchanger inlet temperature differential and recuperator for RC318](image)

To continue operating a cycle at maximum potential output, operating conditions (i.e. evaporator pressure) should be adjusted with changing ambient temperature. A relatively small change in coolant temperature can have a significant impact on the produced power as seen by the difference in monthly power production between cold and hot months.

Figure 22 shows that along with changes in evaporator pressure there are associated changes to the heat exchanger inlet temperature differentials. The condenser inlet temperature differential stays fairly constant throughout the year. This means that the outlet temperature of the turbine rises and falls with the ambient temperature. So in cooler months the outlet pressure of the turbine can be lower providing more power across the turbine. Which allows the temperature differential at the evaporator inlet to be greater. A higher evaporator temperature differential will reduce pinch point effects in the evaporator and allow more heat to be drawn out of the geothermal brine. That means that in cooler months it is more efficient with respect to $\beta$, to have a lower turbine inlet temperature to increase the effectiveness of the heat exchange in the evaporator and give an overall higher $\beta$. 
The lower plot in Figure 22 shows the months that a recuperator is most beneficial. In the coldest months (June, July and August in Australia) that recuperation will add a benefit with respect to $\beta$. It shows that there is a theoretical advantage, but this type of decision would ultimately be decided based on economics. The question would have to be answered if a recuperator operating for only three months of the year makes financial sense.

The impact of adjusting operating conditions according to changing ambient temperature is more evident when comparing a cycle that adjusts to changes to one that doesn’t. Figure 23 shows the annual power and operating conditions for a cycle that does not adjust to changing ambient temperature.

![Monthly Power and Cumulative Energy](image)

**Figure 23. Annual power, energy, temperatures and constant evaporator pressure for RC318**

There are relatively small differences in the performance between the cycle that adjusts and the one that does not. But when these differences are accumulated throughout the year there is a large difference in annual power production. A gain in annual energy production in this case of more than 13% can be realised by making small adjustments to the operating conditions in response the changing ambient conditions.

### 6.2 Operating Conditions Maps

As annual ambient temperatures change, the performance of a given cycle changes accordingly. To maintain the optimum performance of a cycle with varying ambient temperature, the cycles operating conditions need to change accordingly.

Similar to $T_c$, $T_h$ may vary as well. In some reservoirs, the temperature of the geothermal fluid may decrease as the reservoir cools over its production life. Or, in more extreme cases, the $T_h$ can vary
hourly if solar boosting is incorporated into the heat addition system. A power station will likely find that it will need to operate over a range of both $T_c$ and $T_h$. To ensure that the plant is designed optimally for the full range of $T_c$ and $T_h$ expected to be encountered during the lifetime of the plant, optimum operating conditions maps can be useful.

Optimum operating conditions maps show the optimum operating conditions (evaporator pressure, heat exchanger inlet temperature differential, mass flow rates, and where to employ recuperation) to achieve the highest $\beta$ for all possible combinations of $T_c$ and $T_h$ that a cycle may be expected to operate within.

For each fluid analysed in section 6.1, optimum operating conditions maps were generated to produce the highest $\beta$ for each combination of $T_h$ and $T_c$. To illustrate how the maps are used and interpreted, the maps for pentane are presented in the following section.

**$\beta$ versus $\eta_{1st}$**

To show the importance of using $\beta$ instead of $\eta_{1st}$ for geothermal binary systems a comparison of $\eta_{1st}$ versus $T_h$ and $T_c$ based on maximum $\beta$ versus being plotted based on maximum $\eta_{1st}$ is shown in Figure 24. The plot on the left is the associated $\eta_{1st}$ for the maximum $\beta$ at each combination of $T_c$ and $T_h$. Whereas the plot on the right is the maximum $\eta_{1st}$ for each $T_c$ and $T_h$.

Comparing the two plots it can be seen that the plot on the right shows higher values of $\eta_{1st}$ compared to the plot on the left which is sorted based on maximum $\beta$ for each combination of $T_c$ and $T_h$. As mentioned in the section on Cycle Calculations, there are conditions where the maximum $\beta$ doesn’t necessarily correlate to the maximum $\eta_{1st}$. Take the point where $T_h = 170^\circ C$ and $T_c = 20^\circ C$ in the plots of Figure 24. The maximum achievable $\eta_{1st}$ is 14% but the highest $\beta$ is
based on a \( \eta_{1st} \) value of 10\%. Figure 25 shows the relationship between \( \beta \) and \( \eta_{1st} \). There is a maximum of \( \beta \) with respect to \( \eta_{1st} \) at \( \sim 10\% \).

Figure 25 shows the relationship between \( \beta \) and \( \eta_{1st} \). There is a maximum of \( \beta \) with respect to \( \eta_{1st} \) at \( \sim 10\% \).

Recall the equations for \( \beta \) and \( \eta_{1st} \) are

\[
\begin{align*}
n_{1st} &= \frac{W_{net}}{Q_i} \\
\beta &= \frac{W_{net}}{m_h} \\
\beta &= n_{1st} c_R (T_{hi} - T_{ho})
\end{align*}
\]

In Figure 25 it can be seen that as \( \eta_{1st} \) increases \( m_h \) begins to rapidly increase after 10\%. Because \( m_h \) is increasing at a more rapid rate than \( W_{net} \), \( \beta \) starts to decrease. But \( \eta_{1st} \) is increasing because \( W_{net} \) continues to increase and \( Q_i \) continues to decrease, but at the expense of a very large \( m_h \). This relates back to the efficiencies in the evaporator.

Figure 26 shows two cycles from the conditions modelled in Figure 25. The variant in the two cycles is the evaporator pressure. One cycle has an evaporator pressure of 1,000 kPa and the other is 1,900kPa. The higher pressure cycle has a larger \( \eta_{1st} \) but a lower \( \beta \). The higher pressure cycle is not as effective at drawing heat from the geothermal brine (shown by small difference in the brines inlet and outlet temperature) thereby requiring that the mass flow rate of the brine be large. Between 1,000kPa and 1,900kPa there is a 251\% increase in \( m_h \) and an 18\% decrease in \( Q_i \) where \( W_{net} \) only changes 4\%. \( \beta \) is inversely related to \( m_h \) which is why \( \beta \) decreases with the increase in pressure. \( \eta_{1st} \) is inversely related to \( Q_i \) which is why \( \eta_{1st} \) increases with the increase in pressure but because \( m_h \) changes significantly more than \( Q_i \), \( \beta \) decreases while \( \eta_{1st} \) increases.
**Evaporator Pressure**

The operating conditions maps can be viewed together to determine what are some of the optimum operating conditions for a given set of operating temperatures. Figure 27 shows the operating
condition maps for $\beta$ and $P_2$ for pentane. Assume that a geothermal well is producing brine at 140°C ($T_h$) and that the ambient air temperature is 35°C ($T_c$).

Figure 27. Optimum $\beta$ (left) and evaporator pressure (right) for pentane

Looking at the left hand plot for $\beta$ in Figure 27 and moving along the x-axis ($T_h$) to 140°C and then and moving along the y-axis ($T_c$) to 35°C the optimum $\beta$ calculated for those operating temperature conditions can be seen to be $\sim 15$kJ/kg. On the right hand plot for $P_2$ in Figure 27 the optimum $P_2$ can be seen to be $\sim 700$kPa (again moving along the x-axis to 140°C and the y-axis on the lower contour plot to 35°C and finding the corresponding pressure in the pressure colour bar).

$EHX_{dT}$ and $CHX_{dT}$

The same can be done for the contour plots in Figure 28 to identify the optimum $EHX_{dT}$ and $CHX_{dT}$ for $T_h = 170$°C and $T_c = 35$°C as 11°C and 17°C respectively.

Figure 28. $EHX_{dT}$ (left) and $CHX_{dT}$ (right) operating conditions map for optimum $\beta$ for pentane

Because of the penalty of fan cooling, $CHX_{dT}$ remains high across the range of $T_h$ and $T_c$. Where as $EHX_{dT}$ is more closely related to $T_h$ to minimise $m_h$. When there is a greater $T_h$ there is room to
have increased $EHX_{dt}$ at the expense of the available enthalpy across the turbine but at the gain of smaller a $m_h$.

**Mass Flow Rate**

Figure 29 shows the optimum mass flow rates for the geothermal fluid, the working fluid and the coolant to be $m_h$ as 5kg/s and $m_c$ as 40kg/s.

**Recuperation**

Figure 30 shows the operating temperatures where recuperation should and should not be used to achieve the optimum $\beta$. The markers in this figure are coloured as black and white. Black markers indicate that recuperation is beneficial and white (hollow) markers indicate that no recuperation is best to optimise $\beta$. So again moving along the x-axis to 140°C and along the y-axis to 35°C a black marker is present and indicates that recuperation is beneficial to optimise $\beta$.

**Figure 30.** Optimum usage of recuperation for RC318. Solid markers show where recuperation is advantageous.
For an example, looking at Figure 30 recuperation is to be employed, for $T_c$ of 36°C, at $T_h$ equal to 141°C. Figure 31 shows an example of $\beta$ versus $P_2$ for a $T_c$ of 36°C and $T_h$ equal to 120°C, 141°C and 170°C.

The zoomed in view of $T_h$ equal to 170°C shows that no recuperation at $P_2$ equal to 928kPa is the optimal configuration. The zoomed in view of $T_h$ equal to 141°C shows that recuperation at 740kPa is the optimum. And the view for $T_h$ equal to 120°C shows that no recuperation at 480kPa is the optimum. There are narrow margins for $\beta$ between the scenarios of recuperated and non-recuperated for pentane in these cases. A decision based on the economics of a particular power station would have to be made whether or not to employ recuperation.

It should be noted that this methodology is being explained assuming that reader of this information would be performing these analysis using a computer. It is difficult to extract exact values from the operating conditions maps when they are in their printed format. When they are used as an electronic image the exact values can be extracted. All the code written as part of this thesis has been...
been placed in the appendices allowing the reader to perform these analyses in Matlab™. The operating conditions maps were contrived as a way to visually identify patterns and see if there are local maximums and minimums, as in there are “sweet spots” for the optimum $\beta$.

There are many variables in the design of a cycle and there is much interaction. When all the variables are considered simultaneously there becomes a very large number of possible operating conditions. Because the number of iterations for each variable increases the possible solutions grows exponentially. And with more working fluids becoming available more cycles are possible. The only way to thoroughly investigate cycles is to use a computer program such as ORCCA discussed in this chapter. It allows for examining the relationship between multiple variables and the ability to select with confidence the optimum operating conditions for a cycle.
Chapter 7  ORC Test Apparatus

7.1 Introduction

At the onset of this thesis there were no facilities for conducting experimentation. All equipment and data acquisition systems were built by the members of QGECE’s power conversion group, including the author. A requirement of this thesis was to build a working ORC test rig capable of testing small turbines and a variety of different fluids in subcritical and trans-critical Rankine cycles. Building and commissioning the ORC test rig became the largest part of this thesis.

7.2 System Design

The laboratory test facility employed was a condensing Rankine cycle consisting of a positive displacement piston pump, evaporator, single stage impulse turbine, recuperator and a condenser. All heat exchangers are brazed plate-heat exchangers. The basic layout of the experimental setup is shown in Figure 32.

![Turbine Test Loop Diagram](image-url)

**Figure 32. Turbine Test Loop**

The actual ORC test rig that was built and installed in the University of Queensland’s QGECE Power Conversion Laboratory is shown in Figure 33.

---

**Table of Equipment Specifications**

- **Hot Oil Assembly**
  - Thermal Fluid: Ethylene Glycol
  - Max Temperature: 180 deg C
  - Max Power: 20 kW

- **Pump** (Model: CAT Pump 2SF29-ELS)
  - Max Inlet Pressure: 0.51 MPa
  - Max Outlet Pressure: 10.3 MPa
  - Max Temperature: 71 deg C
  - Seals: EPDM, Permachem 6235
  - Max/Min Flow Rate: 10.8 lpm
  - Max Temperature: 71 deg C
  - Volume: 8.5 mm stroke by 18 mm bore

- **Evaporator** (Model: AL34-30, AHHT)
  - Max Pressure: 2500 kPa
  - Max Temperature: 185 deg C
  - Volume: 0.76 ltrs per side

- **Refrigerator** (Model: AL34-30, AHHT)
  - Max Pressure: 2500 kPa
  - Max Temperature: 185 deg C
  - Volume: 0.76 ltrs per side

- **Condensor** (Model: AL34-30, AHHT)
  - Max Pressure: 2500 kPa
  - Max Temperature: 185 deg C
  - Volume: 0.76 ltrs per side

- **Pressure Relief Valve**
  - Actuation Pressure: 1800 kPa

- **Torque Meter and Hysteresis Brake**
  - Max Speed: 27000 RPM
  - Max Torque: 22 N·m
  - Measurement outputs: Torque and Speed

- **Coriolis Flow Meter** (Model: Siemens 7ME4100-1ED13-1AB1)
  - Max Pressure: 20 MPa
  - Temperature Range: -52 to 180 deg C
  - Min/Max Flow Rate: 0 to 14 kg/s
  - Connections: 1.1/2" NPT Male
  - Accuracy: 0.1 % of rate

- **Cooling Water Supply**
  - Max Flow Rate: 2.5 kg/s
  - Inlet Temperature: 20 deg C

---

63
The heater on the left of the photo is a 20kW immersion heater. The thermal fluid in the hot oiler is ethylene glycol. The maximum temperature limit for the ethylene glycol was set to 180°C, however the loop was generally only operated at no more than 140°C in order to maintain a margin of safety. All the pipe work in the system is 12.5mm stainless steel swaged tubing with swaged fittings and connections. Swaged connections were selected to minimize the number of o-rings and seals in the system and because of the ability to be fastened and unfastened multiple times while maintaining a pressure seal. The swaged fittings proved to be very effective and reliable.

The pump that was selected for this loop was a CAT triplex piston pump. It has a maximum flow rate of 0.18 kg/s. However, due to the limited heat capacity of the heater (20kW) the maximum flow rate was initially 0.07 kg/s.

This is can be explained by looking at the cycle. Figure 34 shows the T-S diagram for a non-recuperated cycle. An energy balance in the evaporator from points 2 to 5 for a ~20kW heat input translates to a maximum working fluid mass flow rate of 0.074 kg/s.

To try and increase the available working fluid mass flow rate a recuperator was installed in the loop after the loop had been operating for about 1 year. The right hand plot shows the T-S diagram for a recuperated cycle. The recuperated heat allows the cycle to be operated at higher working mass flow rates. The recuperated cycle has a maximum permissible working fluid mass flow rate of 0.09kg/s.
The other limiting factor for the working fluid mass flow rate is the maximum operating pressure of the magnetic coupling from Dauermagnet-SystemTechnik in the turbine. A magnetic coupling with a ceramic canister was incorporated into the turbine design to eliminate issues with shaft seals. The first revision of the impulse turbine used lip seals on the shaft to seal the fluid in the turbine and a direct mechanical jaw coupling for power transmission. But the shaft seals introduced a problem as they would wear. Because they would wear, the friction they imparted on the shaft would change over time. This change in friction made it difficult to consistently measure power and attribute losses to the various loss mechanisms because the loss of the seals was not constant over time. So a magnetic coupling was introduced into the turbine design.

![Figure 34. T-S diagram of average operating conditions for a R245fa cycle without recuperation (left) and with recuperation (right) in ORC Test Loop.](image)

The magnetic coupling proved to be a very beneficial design concept for the turbine by eliminating rotary shaft seal issues (wear, friction, fluid compatibility and lubrication), however, the maximum operating pressure of the coupling was 1,000kPa. As downstream of the turbine are the recuperator and two condensers, the working fluid mass flow rate is limited by the friction pressure developed as the fluid flows through these heat exchangers. For the average operating conditions of the ORC test loop the limiting working fluid mass flow rate is 0.08kg/s. Above 0.08kg/s the turbine outlet pressure approached the pressure limit of the magnetic coupling due to increased pressure drop through the heat rejecting heat exchangers.

Low working fluid mass flow rate has large design implications on turbomachinery that can be operated in the loop. For the axial impulse turbine built and tested in this thesis, it limited the number of stator nozzles therefore limited the rate of admission. The mass flow rate limits the number of stator nozzles because the throat of the stator nozzles can only be made so small. For a given mass flow rate, more stator nozzles means smaller stator nozzle throat size. The stator nozzles in this thesis were machined on a 5 axis CNC machine that had a minimum mill size of
1mm. It was decided that to get the most accurate cut and best finish on the stator nozzle the minimum throat size for the stator nozzles would be 2.5mm.

The final ORC test rig capabilities are listed in Table 8.

<table>
<thead>
<tr>
<th>Table 8. QGECE ORC Test Rig Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluids</td>
</tr>
<tr>
<td>Working Fluid Mass Flow Rate</td>
</tr>
<tr>
<td>Maximum Evaporator Pressure</td>
</tr>
<tr>
<td>Thermal Fluid Maximum Inlet Temperature</td>
</tr>
<tr>
<td>Maximum Heat Addition</td>
</tr>
<tr>
<td>Cooling Water Inlet Temperature</td>
</tr>
<tr>
<td>Maximum Expander RPM</td>
</tr>
<tr>
<td>Maximum Expander Torque</td>
</tr>
<tr>
<td>Maximum Turbine Outlet Pressure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 0.08</td>
<td>kg/s</td>
</tr>
<tr>
<td>2,500</td>
<td>kPa</td>
</tr>
<tr>
<td>180</td>
<td>°C</td>
</tr>
<tr>
<td>20</td>
<td>kW</td>
</tr>
<tr>
<td>20-25</td>
<td>°C</td>
</tr>
<tr>
<td>10,000</td>
<td>RPM</td>
</tr>
<tr>
<td>6</td>
<td>N-m</td>
</tr>
<tr>
<td>1,000</td>
<td>kPa</td>
</tr>
</tbody>
</table>

The components used in the ORC test loop are listed in Table 9. The operating procedures for the ORC test rig are listed in Appendix D.

During the first eighteen months of this the thesis the ORC test rig was being designed and built. All pipe work, data acquisition systems, turbine and test stand, auxiliary systems (water, air, heater) were designed and built in house. The pump, instrumentation, immersion heater and heat exchangers were bought-in components. Upon completion of the build, a further three months was required to commission the rig and ensure that it was safe for operation and that the data acquisition system and controls were operating properly.

7.3 Experimentation

After the rig was commissioned and operating safely and reliably, experimentation began. The impulse turbine was first tested as a single stage, single stator nozzle expander using R134a and R245fa as the working fluids. The turbine was tested for a range of inlet pressures, speeds, torques, inlet temperatures, and outlet pressures. After the single stator nozzle tests were finished the turbine was taken out of the rig and a second stator nozzle assembly was machined so that the turbine could be tested as a two stator nozzle machine. The intent was to be able to see the effects of admission rate. The turbine was then put back into the rig and tested with two stator nozzles. Only R245fa was able to produce usable two nozzle data because for the R134a case the minimum achievable turbine outlet pressures was too high for the stator nozzle to achieve supersonic flow rates as designed. If a chiller were available to reduce the cooling water temperature or if heat exchangers with less friction pressure were available then the R134a dual nozzle case could have been tested. But in the time frame and budget of this thesis those options were not available.
7.3.1 Test Procedure

1. The loop was filled with a known mass of refrigerant to act as the working fluid. The mass would be calculated based on the desired operating conditions using ORCCA.

2. The cooling water would then be turned on to start condensing the refrigerant on the pump inlet side. It’s imperative that the working fluid be a saturated liquid for the pump to operate effectively. Thermocouples measured the cooling water inlet and outlet temperatures from the condenser.

3. Then the heating loop would be started and set to the required temperature. The hot oiler pump would be turned on to start circulating thermal fluid through the evaporator. Thermocouples measured the inlet and outlet temperature of the thermal fluid in the evaporator.

4. Once the thermal fluid reached the temperature set point and stabilised, the working fluid pump was turned on at minimum rate to start circulating working fluid through the loop.

5. The pressure and temperature of the working fluid was monitored at critical points throughout the loop (turbine inlet, turbine outlet, pump inlet, pump outlet). As the working fluid was circulated the temperatures and pressures stabilise across the loop.

6. A thermal couple was used to measure the temperature of the turbine body. The loop was allowed to circulate at a low rate until the turbine body temperature stabilised.

7. When the turbine body and the monitored pressures and temperatures in the loop were stabilised the pump rate was increased to achieve the required mass flow rate. The mass flow rate was measured using a Coriolis mass flow meter. The Coriolis flow meter also provided density reading which was used to ensure that saturate liquid was entering the pumps inlet.

8. In order to achieve the required test conditions the mass of the working fluid had to be correct inside the loop. If the required test conditions could not be achieved the current working fluid mass, working fluid would either be added or withdrawn until the required conditions were achieved.

9. Once all required conditions were achieved and all pressures and temperatures were stable the turbine would then be tested at a range of speeds by controlling the applied torque to the turbine shaft with the hysteresis brake. For each set of conditions, the rotational speed was varied from high to low speed achieving 10 different rotational speeds. The speed variation was done two times in each recorded data set for redundancy. Each speed was held stable
for 2 min. An example plot of rotational speed versus time for a test set is illustrated in Figure 35.

![Example plot of rotational speed versus Time](image)

Figure 35. Example plot of rotational speed versus Time

10. The following parameters were measured and recorded for the duration each test.
   a. Torque
   b. Turbine shaft speed
   c. Turbine inlet and outlet pressure and temperature
   d. Pump inlet and outlet pressure and temperature

The above methodology was used to test the turbine for a range of conditions and all the recorded data was used to calibrate the loss model that is described in Chapter 9.
<table>
<thead>
<tr>
<th>Item</th>
<th>Model</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluid Pump</td>
<td>CAT Pump 2SF29ELS</td>
<td>Max Inlet Pressure: 510kPa&lt;br&gt;-Max Outlet Pressure: 10,300kPa&lt;br&gt;-Max/Min Flow Rate: 0.18liters/s&lt;br&gt;-Max Fluid Temperature: 71ºC&lt;br&gt;-Seals: EPDM, Permachem 6235&lt;br&gt;-Displacement: 8.5mm stroke by 18mm bore</td>
</tr>
<tr>
<td>Heat Exchangers</td>
<td>AHHT AL34-30</td>
<td>-Max Pressure: 2,500kPa&lt;br&gt;-Max Temperature: 185ºC&lt;br&gt;-Volume: 0.76liters per side</td>
</tr>
<tr>
<td>Torque Meter</td>
<td>Honeywell 1804-200</td>
<td>-Max Speed: 27,000RPM&lt;br&gt;-Max Torque: 22.6N-m&lt;br&gt;-Excitation: 3.28kHz optimum @ 10Vac RMS&lt;br&gt;-Output @ rated capacity: 2 mV/V (nominal)</td>
</tr>
<tr>
<td>Pressure Transducers</td>
<td>Druck PMP 1400</td>
<td>-Max Pressure: 10,000kPa&lt;br&gt;-Analogue Output: 0-5Vdc&lt;br&gt;-Supply: 9-30Vdc&lt;br&gt;-Pressure Connection: G1/4” Female (1/4” BSPPF)&lt;br&gt;-Operating Temperature: -20 to 80 ºC</td>
</tr>
<tr>
<td>Thermocouples</td>
<td>TC Direct Type K</td>
<td>-Temperature Range (Continuous): 0 - 1100 ºC&lt;br&gt;-Temperature Range (Short Term): -180 - 1300 ºC&lt;br&gt;-Tolerance: ±1.5 between 40ºC and 375ºC&lt;br&gt;-Colour Code: IEC, Green (+), White (-)</td>
</tr>
<tr>
<td>Cooling Source</td>
<td>Underground Water Tank With</td>
<td>-Max Flow Rate: 2.5liters/s&lt;br&gt;-Inlet Temperature: 20-30 ºC (Ambient Dependent)</td>
</tr>
<tr>
<td></td>
<td>Grundfos Vertical Centrifugal Pump (CR5-29)</td>
<td></td>
</tr>
<tr>
<td>Heat Source</td>
<td>Ethylene Glycol Immersion Heater Assembly with Electrically Driven Centrifugal Pump</td>
<td>-Max Temperature: 180ºC&lt;br&gt;-Max Flow Rate: 1liter/s&lt;br&gt;-Type of Fluid: Ethylene Glycol&lt;br&gt;-Max Thermal Output: 20kw</td>
</tr>
<tr>
<td>Mass Flow Meter</td>
<td>Siemens 7ME4100-1ED13-1AB1</td>
<td>-Max Pressure: 20,000kPa&lt;br&gt;-Temperature Range: -50 to 180ºC&lt;br&gt;-Min/Max Flow Rate: 0 to 14kg/s&lt;br&gt;-Connections: 1/2” NPT Male&lt;br&gt;-Accuracy: 0.1 % of rate</td>
</tr>
<tr>
<td>Hysteresis Brake</td>
<td>Magtrol AHB-6</td>
<td>-Torque: 6N-m @ 1.5Amps&lt;br&gt;-Max Speed: 20,000RPM&lt;br&gt;-Voltage: 24.8Vdc&lt;br&gt;-Nominal Power Supply: 37W&lt;br&gt;-Air Supply: 620kPa&lt;br&gt;-Kinetic Power Rating: 3kW (with air) 0.225 (w/o air)</td>
</tr>
</tbody>
</table>
Chapter 8  Single Stage Supersonic Impulse Turbine

A simple supersonic impulse turbine consists of a single stator and rotor stage. The expansion of the gas takes place in the nozzles (stator). The expanded gas is accelerated through a converging-diverging nozzle to supersonic speeds. The gas then enters the blades (rotor). The shape of the blades is such that it turns the gas thereby transferring momentum from the gas to the blades. The transferred momentum to the blades causes the rotor to rotate and thus creating mechanical energy. Figure 36 illustrates the relationship between pressure and velocity of the gas as it passes through the stator and rotor.

![Impulse Turbine Stator and Rotor Axial Pressure-Velocity Distribution](image)

8.1 Impulse Turbine Design Program

An axial impulse turbine design program was written as a major portion of this thesis which consists of several modules; (1) 2D divergent nozzle design (2) velocity triangle analysis (3) 2D path and surface generation (3) 3D path and surface generation (4) CAD import geometry generation. The axial design program was short named AXIAL and full source code is included in Appendix B. The AXIAL code was written by the author as a major portion of this thesis. As well the author designed, built (with the support of QGECE Mechanical Engineering Workshop) and commissioned the single stage axial impulse turbine as a major portion of this thesis.
8.1.1 Nozzle Design

A minimum length divergent nozzle design code published by Olson (2012) was adapted to the axial turbine design program. The nozzle design program is based on the method of characteristics as described by Anderson (2007). The program gives the wall contours for the expansion section such that the flow is not over expanding. The calculations begin with determining the theoretical exit conditions using the following relationships:

\[
\frac{A_{II}}{A_{Ii}}^2 = \frac{1}{M_{II}^2} \left[ \frac{2}{\gamma + 1} \left( 1 + \frac{\gamma - 1}{2} M_{II}^2 \right) \right]^{(\gamma + 1)/\gamma - 1} \tag{39}
\]

\[
\frac{p_I}{p_{II}} = \left( 1 + \frac{\gamma - 1}{2} M_{II}^2 \right)^{\gamma/(\gamma - 1)} \tag{40}
\]

Based on the theoretical exit conditions the maximum nozzle wall angle is determined as follows:

\[
\theta_{\text{max}} = \sqrt{\frac{\gamma + 1}{\gamma - 1}} \tan^{-1} \left( \frac{\sqrt{(\gamma - 1)(M_{II}^2 - 1)}}{\gamma + 1} \right) - \tan^{-1} \left( \frac{\sqrt{M_{II}^2 - 1}}{2} \right) \tag{41}
\]

Then based on \( \theta_{\text{max}} \), iteratively determine the constants along the characteristic lines and the local Mach angle as defined by the following relationship

\[
\sin(\mu) = \frac{1}{M} \tag{42}
\]

For the range of \( \theta \) and \( \mu \) values that are iterated across, determine the characteristic lines (Refer to the Axial Code Appendix section to see the python code used).

8.2 Velocity Triangle

Based on the calculated stator exit velocity from the nozzle design, the stator and rotor analysis continues with a velocity triangle analysis. Figure 37 illustrates the relationships between the velocity vectors in a velocity triangle as defined by Glassman (1994).
Figure 37. Single Stage Stator-Rotor Velocity Triangle

The relationships of all components in the velocity triangle are described in equations 43 - 60 as listed in Table 10.

For an impulse turbine there are two critical assumptions to make in order to calculate all the components of the velocity triangle. First is that the axial velocity is constant through the rotor and the second is that there is no reaction in the rotor ($R = 0$).

The two common parameters for quantifying efficiency are static efficiency and total efficiency. The static efficiency is based on the ideal work from the inlet total conditions to the exit static conditions.
Table 10. Velocity Diagram Equations

Rotational Velocity
\[ \omega = \frac{2\pi \text{ RPM}}{60} \] (43)

Blade Tip Speed
\[ U = \frac{D_r \omega}{2} = Z(V_{u,II} - V_{u,III}) \] (44)

Absolute Tangential Velocity, Rotor Inlet
\[ V_{u,II} = V_{II} \sin \alpha_{II} \] (45)

Absolute Axial Velocity, Rotor Inlet
\[ V_{x,II} = V_{II} \cos \alpha_{II} \] (46)

Relative Tangential Velocity, Rotor Inlet
\[ Y_{u,II} = V_{u,II} - U \] (47)

Relative Axial Velocity, Rotor Inlet
\[ Y_{x,II} = V_{x,II} \] (48)

Relative Velocity, Rotor Inlet
\[ Y_{II} = \left( Y_{u,II}^2 + Y_{x,II}^2 \right)^{1/2} \] (49)

Relative Flow Angle, Rotor Inlet
\[ \beta_{II} = \sin^{-1} \frac{Y_{u,II}}{Y_{II}} \] (50)

Absolute Axial Velocity, Rotor Outlet
\[ V_{x,III} = V_{x,II} \text{ (Assume Constant Axial Velocity)} \] (51)

Relative Axial Velocity, Rotor Outlet
\[ Y_{x,III} = V_{x,III} \] (52)

Relative Tangential Velocity, Rotor Outlet
\[ Y_{u,III} = -Y_{x,III} \tan(\beta_{III}) \] (53)

Relative Velocity, Rotor Outlet
\[ Y_{III} = Y_{II} = \left( Y_{u,III}^2 + Y_{x,III}^2 \right)^{1/2} \] (54)

Absolute Tangential Velocity, Rotor Outlet
\[ V_{u,III} = V_{u,II} \left( \frac{Z - 1/2}{Z + 1/2} \right) = Y_{u,III} + U \] (55)

Absolute Velocity, Rotor Outlet
\[ V_{III} = \left( V_{u,III}^2 + V_{x,III}^2 \right)^{1/2} \] (56)

Absolute Flow Angle, Rotor Outlet
\[ \alpha_{III} = \tan^{-1} \frac{V_{u,III}}{V_{x,III}} \] (57)

Speed-Work Parameter
\[ Z = \frac{U}{V_{u,II} - V_{u,III}} \] (58)

Reaction
\[ \text{Reaction} = \frac{Y_{III}^2 - Y_{II}^2}{Y_{III}^2 - Y_{II}^2 + V_{II}^2} \] (59)

Work Output
\[ W_o = m U (V_{u,II} - V_{u,III}) \] (60)

The total efficiency is based on the ideal work from the inlet total conditions to the exit total conditions. Static efficiency is useful when the gas exit velocity from the turbine is useful (as in a jet engine) but in this case the exit velocity doesn’t provide any further work as the gas is being directed to a condenser so the total efficiency is used and defined as
\[ ETAT = \frac{W_o}{h_I - h_{II}} \]  \hspace{1cm} (61)

A specified range of minimum to maximum operating speeds, flow angles, and turbine geometric variables were analysed and the conditions that provided the maximum work output were passed to the next module to generate 2D geometric paths for the stator and rotor passage.

8.2.1 2D Paths

The stator and rotor passages were broken up into multiple curvilinear paths. This is done because it allows the geometry to be readily imported into CAD and CFD programs for modelling and analysis. Figure 38 shows the path geometry and notation for a single stator and rotor passage. The z-axis is the axial direction, the y-axis is the tangential direction, and the x-axis is the tip to shroud direction.

![2D Paths For Single Stator and Rotor Passage](image)

Figure 38. 2D Stator and Rotor Passage Geometric Path Designation

The letters in the figure above are used to denote end points of the paths that comprise the 2D geometry. The primary dimensions of the stator and rotor passage of the impulse turbine are illustrated in Figure 39.
Figure 39. Nozzle-blade passage geometry

**Stator Paths**

The divergent portions of the nozzle (paths ON and BC in Figure 38) are derived from the paths created in the nozzle design module. The convergent portions of the nozzle (paths AB and PO in Figure 38) are simply designed as straight line segments with an inlet width that is greater than the distance from O to B. The points are rotated about the origin (0, 0) with respect to nozzle absolute flow angle ($\alpha_{II}$) as follow;

$$z_{rotated} = \cos(\alpha_{II}) - y_{unrotated} \sin(\alpha_{II})$$  \hspace{1cm} (62)$$

$$y_{rotated} = \sin(\alpha_{II}) - y_{unrotated} \cos(\alpha_{II})$$  \hspace{1cm} (63)$$

**Rotor Paths**

Based on the stator paths, the rotor paths are calculated. The chord, pitch, throat, solidity and pressure surface radius were calculated as follows;
Pitch
\[ p = \frac{\pi D_{r,tp}}{N_{bld}} \]  \hspace{1cm} (64)

Rotor Throat Width
\[ w_{r,tt} = (p - te_r) \sin\left(\frac{\pi}{2} - \beta_{II}\right) \]  \hspace{1cm} (65)

Solidity
\[ s = \left(\frac{2}{zw}\right) \left(\frac{\cos(-\beta_{III})}{\cos(\beta_{II})}\right) \sin(\beta_{II} + \beta_{III}), \text{ (Glassman 1994)} \]  \hspace{1cm} (66)

Chord
\[ c = s p \]  \hspace{1cm} (67)

Pressure Surface Radius
\[ R_{ps} = R_{ss} w_{r,tt} \]  \hspace{1cm} (68)

The points that outline the suction (paths DE, EF, and FG) and pressure (paths ML, LK, and KJ) surfaces of the rotor passage and the centre of curvature (point Q) are determined as follows:

\[ z_E = z_D + \psi_{II} c \]  \hspace{1cm} (69)

\[ y_E = y_D + (z_E - z_D) \tan(\beta_{II}) \]  \hspace{1cm} (70)

\[ z_Q = z_E + R_{ss} \cos\left(\frac{\pi}{2} - \beta_{II}\right) \]  \hspace{1cm} (71)

\[ y_Q = y_E - R_{ss} \sin\left(\frac{\pi}{2} - \beta_{II}\right) \]  \hspace{1cm} (72)

\[ z_F = z_Q + R_{ss} \cos\left(\frac{\pi}{2} - \beta_{III}\right) \]  \hspace{1cm} (73)

\[ y_F = y_Q + R_{ss} \sin\left(\frac{\pi}{2} - \beta_{III}\right) \]  \hspace{1cm} (74)

\[ z_G = z_F + \psi_{III} c \]  \hspace{1cm} (75)

\[ y_G = y_F - \psi_{III} c \tan(\beta_{III}) \]  \hspace{1cm} (76)

\[ z_M = z_D \]  \hspace{1cm} (77)

\[ y_M = y_D + \frac{z_{II} w_{r,tt}}{\cos(\beta_{II})} \]  \hspace{1cm} (78)

\[ z_L = z_E - w_{r,tt} z_{II} \sin(\beta_{II}) \]  \hspace{1cm} (79)

\[ y_L = y_E + w_{r,tt} z_{II} \cos(\beta_{II}) \]  \hspace{1cm} (80)
\[ z_K = z_F + w_{r,t} z_{III} \sin(\beta_{III}) \]  
(81)

\[ y_K = y_F + w_{r,t} z_{III} \cos(\beta_{III}) \]  
(82)

\[ z_J = z_G \]  
(83)

\[ y_J = y_G + z_{III} \frac{w_{r,t}}{\cos(\beta_{III})} \]  
(84)

### 8.2.2 3D Paths

The 2D paths are then rotated to create 3D paths of the stator and rotor passage. The points are rotated by first defining the angle of rotation (\( \theta \)). The x and y points for the 3D paths are rotated from the 2D paths as follows;

\[ \theta = (2\pi) \left( \frac{y}{\pi D_{r,m}} \right) \]  
(85)

\[ x = x \cos(\theta) \]  
(86)

\[ y = x \sin(\theta) \]  
(87)

In the 3D path notation, z points remain the same as the rotation only takes place about the x-axis.

### 8.3 Impulse Turbine Design

As part of this thesis a small axial impulse turbine was designed and machined using the computational design tools developed in the ORCCA and AXIAL programs. The intent of the turbine was to provide a platform for validating the Rankine cycle analysis program (ORCCA) and to validate an axial impulse turbine loss model. The thought process behind the first test turbine was as follows:

To design a turbine based on velocity triangles only. This is contrary to good design process but the thought was that a turbine based on velocity triangles only will have significant losses. As the first turbine was intended to be a platform for studying losses and gathering data that could later be used to develop a loss model a turbine based on velocity triangles was built as a test machine. This was also in the interest of time so that a turbine could be machined in parallel to the development of the ORC test rig and computational models (ORCCA, AXIAL and SSAL).
The validated loss model would then be incorporated into AXIAL to provide turbine designs based on evaluation of velocity triangles and the associated losses. The loss model would also be incorporated into the ORC analysis program (ORCCA) to provide the cycle analysis program with calculated turbine efficiencies rather than assumed efficiencies. The turbine had to be designed to operate within the constraints of the ORC test apparatus which were imposed by limited space and available resources. The axial design program was run for the conditions listed in Table 11. The conditions listed in the table are conditions that were within a comfortable operating range for the ORC test apparatus. The outlet pressure of the stator were based on a cycle that operated with a source temperature of 145°C and cooling water at 22°C. The stator outlet temperature was based on the assumption that the gas would be expanded with an isentropic efficiency of 85%.

R134a was selected as the working fluid because of its availability, relatively low cost, and low hazard levels.

<table>
<thead>
<tr>
<th>Turbine Dimension</th>
<th>Symbol</th>
<th>Value</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluid</td>
<td></td>
<td>R134a</td>
<td></td>
</tr>
<tr>
<td>Working Fluid Mass Flow Rate</td>
<td>m</td>
<td>0.05</td>
<td>kg/s</td>
</tr>
<tr>
<td>Stator Inlet Temperature</td>
<td>T_I</td>
<td>140</td>
<td>°C</td>
</tr>
<tr>
<td>Stator Inlet Pressure</td>
<td>P_I</td>
<td>2,500</td>
<td>kPa</td>
</tr>
<tr>
<td>Stator Outlet Temperature</td>
<td>T_II</td>
<td>100</td>
<td>°C</td>
</tr>
<tr>
<td>Stator Outlet Pressure</td>
<td>P_II</td>
<td>820</td>
<td>kPa</td>
</tr>
<tr>
<td>Speed-Work Parameter Range</td>
<td>Z range</td>
<td>0.10 – 0.90</td>
<td></td>
</tr>
<tr>
<td>Stator Outlet Absolute Flow Angle Range</td>
<td>α_II range</td>
<td>up to 80</td>
<td>°</td>
</tr>
</tbody>
</table>

8.3.1 Impulse Turbine Computational Design

The nozzle design module for the given conditions provided the nozzle contours for a supersonic nozzle as shown in Figure 40 (top left). The predicted exit Mach number for the nozzle was 1.45.

Following the results of the nozzle calculations, the velocity triangle module performs an analysis of varying flow angles and speed-work parameters (α_II range and Z range) to determine the maximum work conditions. Velocity triangles for the range of flow angles and speed-work parameters based on the nozzle exit velocity were calculated and analysed. The flow angle that provided the maximum work output was selected and the speed-work parameter for that flow angle that yield the maximum work condition was selected as the design condition. The condition of α_II = 80° and Z = 0.50 produced the most work as shown in Figure 40 (top right).
The velocity triangle for the ideal maximum work condition is produced as shown Figure 40 (bottom left). This case though is for a turbine with no losses. This is a theoretical maximum with no losses. The ideal best case design would produce a turbine with zero exit swirl, because swirl velocity leaving a turbine is a loss (Glassman 1994), which is shown in the velocity triangle by the absolute rotor exit velocity being perpendicular to the blade speed direction. This is to be expected in an ideal case as exit swirl is a source of losses in a turbine.

Based on the nozzle contours from the nozzle module and the angles from the velocity triangle, the 2D paths for a single rotor and stator passage combination are produced as shown in Figure 40 (bottom right).

The 3D paths are then generated by rotating the 2D paths about the x-axis for a given rotor mean diameter as shown in Figure 41.

The rotor mean diameter selected was based on the limitation of the ORC test loop. Initially the ORC test loop had a load that was only capable of 10,000RPM. This was a car alternator that had been fitted to the torque meter. A mean rotor diameter was selected that would be as close to 10,000RPM for the given velocity triangle. By aiming to have the turbine spin as fast as possible it allowed to minimise its size. The smaller the turbine was the easier and less costly it would be to machine.
The 3D paths are translated into x, y, z coordinates and written to a .csv file for importing the exact geometry into a CAD model. The surfaces that define the tip and root contours of the flow passage for the stator and rotor passage are then generated based on the 3D paths.

The surfaces are used in generating a CFD model of the stator and rotor passage in the Eilmer3, the in-house CFD code. The test turbine dimensions resulting from the AXIAL program are listed in Table 12. This table also lists some of the constraints that influenced the geometry. Some of the constraints were working limits of the ORC test loop and some were based on machining capabilities. This turbine was designed to operate in the ORC test apparatus discussed in Chapter 7.

Again it should be noted that this impulse turbine was built with the intent of being a turbine based on velocity triangles that would have significant losses and therefore allow for losses to be investigated for calibrating a loss model (to be discussed in the next chapter).
### Table 12. Test Impulse Turbine Design Conditions and Dimensions

<table>
<thead>
<tr>
<th>Turbine Dimension</th>
<th>Value</th>
<th>UoM</th>
<th>Design Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluid</td>
<td>R134a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Working Fluid Mass Flow Rate</td>
<td>m</td>
<td>0.05</td>
<td>kg/s</td>
</tr>
<tr>
<td>Stator Inlet Temperature</td>
<td>T_I</td>
<td>140</td>
<td>°C</td>
</tr>
<tr>
<td>Stator Inlet Pressure</td>
<td>P_I</td>
<td>1,800</td>
<td>kPa</td>
</tr>
<tr>
<td>Stator Outlet Temperature</td>
<td>T_II</td>
<td>100</td>
<td>°C</td>
</tr>
<tr>
<td>Stator Outlet Pressure</td>
<td>P_II</td>
<td>720</td>
<td>kPa</td>
</tr>
<tr>
<td>Speed-Work Parameter Range</td>
<td>Z</td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>Stator Outlet Absolute Flow Angle Range</td>
<td>α_II</td>
<td>80</td>
<td>°</td>
</tr>
<tr>
<td>Maximum RPM</td>
<td>RPM</td>
<td>10,000</td>
<td>RPM</td>
</tr>
<tr>
<td>Rotor Mean Diameter</td>
<td>D_r,m</td>
<td>0.1977</td>
<td>m</td>
</tr>
<tr>
<td>Number of Nozzles</td>
<td>N_nozzles</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Stator Nozzle Throat Height (Root to Tip)</td>
<td>h_s,tt</td>
<td>0.0025</td>
<td>m</td>
</tr>
<tr>
<td>Stator Nozzle Throat Width</td>
<td>w_s,tt</td>
<td>0.0023</td>
<td>m</td>
</tr>
<tr>
<td>Rotor Tip Diameter</td>
<td>D_r,tp</td>
<td>0.2000</td>
<td>m</td>
</tr>
<tr>
<td>Rotor Inlet Metal Angle</td>
<td>β_II</td>
<td>71</td>
<td>deg</td>
</tr>
<tr>
<td>Rotor Outlet Metal Angle</td>
<td>β_III</td>
<td>71</td>
<td>deg</td>
</tr>
<tr>
<td>Number of Blades</td>
<td>N_blades</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>Rotor Blade Throat Height (Root to Tip)</td>
<td>h_r,t</td>
<td>0.0026</td>
<td>m</td>
</tr>
<tr>
<td>Stator Outlet Absolute Flow Angle</td>
<td>α_II</td>
<td>80</td>
<td>deg</td>
</tr>
<tr>
<td>Rotor Inlet Metal Angle</td>
<td>β_II</td>
<td>71</td>
<td>deg</td>
</tr>
<tr>
<td>Rotor Outlet Metal Angle</td>
<td>β_III</td>
<td>71</td>
<td>deg</td>
</tr>
<tr>
<td>Number of Blades</td>
<td>N_blades</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>Rotor Blade Throat Width</td>
<td>w_r,t</td>
<td>0.0041</td>
<td>m</td>
</tr>
<tr>
<td>Rotor Blade Suction</td>
<td>R_s,s</td>
<td>0.02</td>
<td>m</td>
</tr>
<tr>
<td>Stator Rotor Gap</td>
<td>g</td>
<td>0.0015</td>
<td>m</td>
</tr>
<tr>
<td>Blade Tip Clearance</td>
<td>j</td>
<td>2.50E-05</td>
<td>m</td>
</tr>
<tr>
<td>Blade Meridional Passage Length</td>
<td>l</td>
<td>0.0791</td>
<td>m</td>
</tr>
<tr>
<td>Blade Chord</td>
<td>c</td>
<td>0.0499</td>
<td>m</td>
</tr>
<tr>
<td>Blade Tip Thickness</td>
<td>t_e</td>
<td>0.001</td>
<td>m</td>
</tr>
<tr>
<td>Stator Exit Mach Number</td>
<td>M_II</td>
<td>1.45</td>
<td></td>
</tr>
<tr>
<td>Rotor Inlet Relative Mach Number</td>
<td>M_I</td>
<td>0.75</td>
<td></td>
</tr>
</tbody>
</table>
8.3.2 Impulse Turbine Mechanical Design

A complete turbine was designed based on importing the 3D paths of the stator and rotor passages into a CAD program. Siemens Solid Edge™ CAD package was used in this investigation to design a single stage axial impulse turbine. The rotor and stator were designed and drawn so that they could be machined in a 5-axis CNC machine. The material selected to machine all components was 6130 aluminium except stainless steel was used for the rotor shaft. Originally sealed stainless steel deep groove roller ball bearings were used to support the shaft on both sides of the rotor for maximum stability. But after six months of testing the bearings were changed to ceramic bearings. Ceramic bearings were selected because of their ability to operate in high temperature, corrosive environments without need for hydrocarbon lubrication. Because there was concern about degradation of the refrigerants or some secondary reactions with impurities when operating at elevated temperatures it was important to maintain an oil free system so that the working fluid could be kept as pure as possible and therefore as stable as possible.

In order to avoid issues with seals at high temperatures and compatibility with refrigerants a magnetic coupling was incorporated into the design to eliminate sealing issues at the shaft. The magnetic coupling provided two critical advantages by eliminating losses due to seal friction and eliminating the need to lubricate seals.

All o-rings used in the machine were silicone 70-FDA. Silicone was selected because of its availability and its resistance to swelling and degradation in a refrigerant environment. Silicone proved to be a good o-ring material and proved reliable. Further advantages are that it is readily available and relatively inexpensive.

The stator was designed in two pieces and machined using a 5-axis CNC machine. The assembly was sealed using Loctite™ 510 and silicone o-rings. Bolts were used to clamp the top and bottom stator pieces together. Figure 42 shows CAD renderings and machined parts of the stator nozzle block.

By designing stator nozzle as a two piece assembly it allowed the parts to be machined on the 5-axis mill rather than cast. The advantage of machining was a high quality surface finish, accuracy, and the ability to make the parts in-house at the University of Queensland’s Mechanical Engineering Workshop. The rotor was designed as a single piece. Figure 43 shows images of the rotor.
The blades of the rotor were machined into the body of the rotor. This allowed for a simple fabrication process and allowed for maintaining tight tolerances by using CNC. The disadvantage to this design was that the blade height had to be kept to a minimum to allow the blade passages to be machined. But because the accuracy of the CNC fabrication process is high, low clearance between rotor tip and the housing was achieved negating some of the negative effects of the low blade height.

The turbine assembly is shown in Figure 44. The image is a CAD rendering of the assembly with a cut-away plane to allow the insides of the assembly to be viewed with respect to the assembly.
All the parts were machined from raw blocks of material. Because the parts were machined from raw stock some of the parts are oversized with respect to the minimum thickness required to operate. This is because machining time has an associated monetary cost and in order to minimize the cost of the fabrication minimal machining was designed into the parts. The casing could have been machined down and made smaller but the only benefit to that would be purely aesthetic as space was not a limitation in the ORC test rig. The placement of the turbine in the ORC test apparatus can be seen in Figure 33.

8.3.3 Stresses

The turbine is a pressure vessel and the pressure inside the turbine generates forces on the components that attempt to push the parts away from one another, explode the turbine. To oppose these internal forces adequate sizes and numbers of bolts need to hold the turbine together. Bolt calculations were done to determine the required number of bolts. The operating conditions of the turbine were set to a maximum inlet pressure of 2,500kPa and outlet pressure of 1,000kPa. Pressure relief valves were set to these pressures upstream and downstream of the turbine to ensure that these pressures were not exceeded. The bolting forces were calculated based on these pressures. Everything down stream of the stator (housing, magnetic coupling) were calculated based on 1,000kPa and the nozzle cover housing was based on 2,500kPa. Table 13 lists the
dimensions, stresses and forces for the bolts. The minimum allowable factor of safety was to be 2 but if a higher safety margin could be achieved it was.

Table 13. Impulse Turbine Bolt Calculations

<table>
<thead>
<tr>
<th>Bolt Calculation</th>
<th>Plate H (m)</th>
<th>Plate W (m)</th>
<th>Plate Area (m²)</th>
<th>Pressure (kPa)</th>
<th>Force (kN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nozzle Cover Housing Bolt Calculations</td>
<td>0.1355</td>
<td>0.1235</td>
<td>0.02</td>
<td>2,500.00</td>
<td>41.84</td>
</tr>
<tr>
<td>Housing Through Bolt Calculations</td>
<td>0.3</td>
<td>0.3</td>
<td>0.09</td>
<td>2,500.00</td>
<td>Force</td>
</tr>
<tr>
<td>Magnetic Coupling Bolt Calculations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Force</td>
</tr>
</tbody>
</table>

Another item of concern was the torque transmission between the rotor and shaft. A press fit was used to fix the rotor to the shaft because it was thought this would eliminate balancing issues if a keyed or splined connection was used. Press fit calculations (Table 14) were done to determine what degree of interference was required between the rotor and the shaft to be able to transmit the required torque.

Table 14. Press fit calculations for the rotor to shaft fit

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shaft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Young's Modulus, Ei (Gpa)</td>
<td>200.0</td>
<td>GPa</td>
</tr>
<tr>
<td>Poisson's Ratio, νi</td>
<td>0.290</td>
<td></td>
</tr>
<tr>
<td>Nominal Shaft Radius, d (m)</td>
<td>0.012</td>
<td>m</td>
</tr>
<tr>
<td>Hub</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Young's Modulus, Eo (Gpa)</td>
<td>68.9</td>
<td>GPa</td>
</tr>
<tr>
<td>Poisson's Ratio, νo</td>
<td>0.330</td>
<td></td>
</tr>
<tr>
<td>Nominal Hub Radius, do (m)</td>
<td>0.020</td>
<td>m</td>
</tr>
<tr>
<td>Nominal Hole Radius, d (m)</td>
<td>0.011988</td>
<td>m</td>
</tr>
<tr>
<td>Results</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running Radial Displacement</td>
<td>0.000001</td>
<td>m</td>
</tr>
<tr>
<td>Radial Interference, δ (m)</td>
<td>0.0000011</td>
<td>m</td>
</tr>
<tr>
<td>Pressure Generated, p (MPa)</td>
<td>2.13</td>
<td>MPa</td>
</tr>
<tr>
<td>Friction Factor Between Shaft and Hub</td>
<td>0.12</td>
<td></td>
</tr>
<tr>
<td>Length of Press Fit Engagement</td>
<td>0.05</td>
<td>m</td>
</tr>
<tr>
<td>Area of Press Fit Engagement</td>
<td>3.77E-03</td>
<td>m²</td>
</tr>
<tr>
<td>Friction Force</td>
<td>963.42</td>
<td>N</td>
</tr>
<tr>
<td>Transmission Torque</td>
<td>11.55</td>
<td>Nm</td>
</tr>
</tbody>
</table>
The radial displacement of the rotor was determined from FEA so that the allowable running torque could be determined by taking into account the enlarged diameter of the rotor when at full temperature and operating speed.

The displacement of the rotor also affects the rotor clearance. The displacement of the rotor is a result of the radial forces created from rotation and from thermal expansion. The running displacement of the rotor was found by conducting FEA on the rotor at the full temperature and operating speeds. The displacement of the housing was also found due to pressure and temperature. A very small clearance was the goal to avoid clearance losses. The housing nominal diameter was taken to be

\[ D_{Housing} = D_{Rotor} + (2\delta_{Rotor}) + (2\delta_{Housing}) - (2j) \]

Where Table 15 lists the values for the displacement of the rotor and the housing.

<table>
<thead>
<tr>
<th>Rotor Nominal Diameter ((D_{Rotor}))</th>
<th>200 (\text{mm})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotor Running Displacement ((\delta_{Rotor}))</td>
<td>0.136 (\text{mm})</td>
</tr>
<tr>
<td>Blade Height ((h_{r,t}))</td>
<td>2.6 (\text{mm})</td>
</tr>
<tr>
<td>Clearance Factor</td>
<td>0.01 (%)</td>
</tr>
<tr>
<td>Radial Clearance Height ((j))</td>
<td>0.026 (\text{mm})</td>
</tr>
<tr>
<td>Housing Running Displacement ((\delta_{Housing}))</td>
<td>0.07 (\text{mm})</td>
</tr>
<tr>
<td>Housing Nominal Diameter ((D_{Housing}))</td>
<td>200.184 (\text{mm})</td>
</tr>
</tbody>
</table>

The FEA reports from SolidEdge are listed in the Appendix E for reference.
Chapter 9 Loss Model

Every turbine will incur losses as the fluid passes through the stators and rotors. The losses are attributed to clearance, windage, partial admission, trailing edge, incidence, rotor passage friction, and bearings. The sum of these losses can be so significant that the optimum operating point of a turbine is changed (Roelke 1994). The meanline analysis employed in this thesis draws upon the work of several authors to frame an appropriate loss model for a single stage supersonic impulse axial turbine. This class of turbomachinery is interesting because it presents an opportunity for small, economic, and robust turbines for use in small scale ORC’s. All are expressed in terms of energy so that they can be easily related to a reduction in turbine shaft work.

The loss model that was employed in this thesis was based primarily on the loss models presented by Roelke. Roelke’s model was selected because it was focused on small axial machines and many of the loss mechanisms were based on test data from single stage impulse machines. The test turbine for this thesis was a single stage impulse turbine so this model seemed like a good foundation. The models for clearance, windage, pumping losses and incidence were taken from Roelke. However the model for sector losses from Roelke seemed to under predict the losses so a modified sector loss model from Varma was employed. The trailing edge loss model was taken from Baines because it had a case specific to impulse turbines. And the secondary losses were accounted for using Aungier’s model for axial flow turbines. This secondary loss model was a modification of the AMDC-KO model with the intention to make it more applicable at extreme-off design conditions and to work at small aspect ratios. One of the aims of this thesis was to recognize the gap between the published loss models and the case tested and to search for coefficients that would make the loss models fit to the experimental data. Knowing that the turbine to be tested was going to be far off design conditions this model seemed like a good choice. Aungier’s model for supersonic shock and expansions was employed as well because Roelke’s model didn’t have a model for this mechanism.

9.1 Clearance Losses

In any axial turbine there is going to be a gap between the rotor and the casing so that the rotor can rotate freely. This gap allows some of the working fluid to pass through the turbine without passing through the blade passage. Because the leaked fluid did not pass through the blade and perform work on the blade it is considered a loss. The complexity of the leakage flow makes accurate correlation of the tip leakage difficult, however it is known that important factors are the gap size
and the blade loading (Baines 1997). As the focus of this loss model is for single stage impulse turbines, the loss model employed for clearance loss is taken from results published by Roelke (1994) that are specific to single stage impulse turbines.

To make the data published by Roelke usable in a computer program, the data is expressed as a polynomial with reaction \((R)\) as the variant and coefficients that are looked up based upon values of the clearance to blade height ratio \((j/h_{r,t})\). The clearance loss coefficient is found by

\[
K_{\text{clearance}} = AR^2 + BR + C
\]  

(89)

where the coefficients \(A, B,\) and \(C\) are found from the following matrix interpolating based upon \(j/h_{r,t}\) ratio. The matrix of values is as follows

<table>
<thead>
<tr>
<th>(A)</th>
<th>(B)</th>
<th>(C)</th>
<th>(j/h_{r,t})</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.023</td>
<td>-0.002</td>
<td>0.983</td>
<td>0.010</td>
</tr>
<tr>
<td>-0.046</td>
<td>-0.005</td>
<td>0.966</td>
<td>0.020</td>
</tr>
<tr>
<td>-0.069</td>
<td>-0.008</td>
<td>0.949</td>
<td>0.030</td>
</tr>
<tr>
<td>-0.092</td>
<td>-0.011</td>
<td>0.932</td>
<td>0.040</td>
</tr>
<tr>
<td>-0.115</td>
<td>-0.014</td>
<td>0.916</td>
<td>0.050</td>
</tr>
<tr>
<td>-0.138</td>
<td>-0.017</td>
<td>0.889</td>
<td>0.060</td>
</tr>
<tr>
<td>-0.161</td>
<td>-0.020</td>
<td>0.882</td>
<td>0.070</td>
</tr>
<tr>
<td>-0.184</td>
<td>-0.023</td>
<td>0.865</td>
<td>0.080</td>
</tr>
<tr>
<td>-0.207</td>
<td>-0.026</td>
<td>0.849</td>
<td>0.090</td>
</tr>
<tr>
<td>-0.230</td>
<td>-0.029</td>
<td>0.832</td>
<td>0.100</td>
</tr>
</tbody>
</table>

The clearance loss is then expressed as

\[
L_{\text{clearance}} = \phi_{\text{clearance}} (1 - K_{\text{clearance}})W_{\text{ideal}}
\]  

(91)

where \(\phi_{\text{clearance}}\) is the machine specific linear clearance loss coefficient and \(W_{\text{ideal}}\) is theoretical available work calculated as

\[
W_{\text{ideal}} = mU(Vu_1 - Vu_2)
\]  

(92)

9.2 Windage Loss

A major source of loss in the turbine is the friction drag on the rotor created by its rotation in a fluid. The windage loss is separated into two components in this loss model. One component accounting for the sides of the rotor (Disc Windage) and the other component is from the drag between the rotor and casing (Clearance Gap Windage).
9.2.1 Rotor Disc Windage Loss

The windage loss results from shear forces acting on the rotor due to the skin friction of the fluid between the rotor and the casing. Roelke (1994) and Aungier (2006) both cite the works of Daily (1958) to describe the mechanism of windage losses. The windage losses are described in terms of a torque coefficient. Daily and Nece conducted thorough investigation of the windage loss as a function of chamber proportions and Reynolds number. There are four flow regimes that can exist in the space between the rotor and the casing. The flow regimes are delineated based upon turbulent or laminar flow and upon merged or separated boundary layers. For small clearances between the rotor and casing the merged boundary layers exist where there is a continuous variation in velocity across the axial gap. In large clearances separate boundary layers exists where between the two boundary layers is a core of rotating fluid in which no change in velocity occurs (Roelke 1994). The four flow regimes are as follows;

i. Laminar, merged boundary layers (small clearance)
ii. Laminar, separate boundary layers (large clearance)
iii. Turbulent, merged boundary layers (small clearance)
iv. Turbulent, separate boundary layers (large clearance)

Figure 45 illustrates the difference between the merged and separate boundary layer conditions.

![Figure 45. Velocity patterns around rotating disks. Flow regimes I and III (left) and Flow regimes II and IV (right) (Glassman 1994)](image)

The windage loss is expressed in terms of a torque coefficient ($C_M$) as
\[ L_{\text{windage}} = \Phi_{\text{windage}} \left[ \frac{1}{2} C_M \rho \omega^{\gamma_{\text{windage}}} \left( \frac{D_{r,tt}}{2} \right)^5 + L_{\text{gap}} \right] \]  

(93)

Where \( \Phi_{\text{windage}} \) and \( \gamma_{\text{windage}} \) are the linear and exponential machine specific windage loss coefficients respectively and \( C_M \) for each flow regime is defined as:

**Flow Regime i**

\[ C_{M,ii} = \frac{2\pi}{\left( \frac{s}{D_{r,tt}} \right) Re} \]  

(94)

**Flow Regime ii**

\[ C_{M,ii} = \frac{3.7 \left( \frac{s}{D_{r,tt}} \right)^{0.10}}{\sqrt{Re}} \]  

(95)

**Flow Regime iii**

\[ C_{M,iii} = \frac{0.08}{\left( \frac{s}{D_{r,tt}} \right)^{0.17} Re^{0.25}} \]  

(96)

**Flow Regime iv**

\[ C_{M,iv} = \frac{0.102 \left( \frac{s}{D_{r,tt}} \right)^{0.10}}{Re^{0.20}} \]  

(97)

The selection of the correct torque coefficient is based on boundary layers and Reynolds number, however the boundary layers may not be known. Aungier (2006) says that the relevant flow regime is identified by the torque coefficient equation that yields the largest value.

The shear force that exists in the clearance gap between the tip of the rotor and the casing are a source of loss. Aungier (2006) showed that the fluid in the clearance gap between the rotating and stationary walls produces viscous shear forces on both walls and in the case of the rotating wall work is done. This work is a loss and is expressed as

\[ L_{\text{gap}} = \pi \rho c_f \left( \frac{D_{\text{tip}}}{2} \right)^4 \omega^3 \frac{l}{4} \]  

(98)

The skin friction coefficient \( c_f \) is a function of the Reynolds number where the assumed mean rotational speed of the fluid in the gap is half of the rotor speed. Therefore the Reynolds number is expressed as

\[ Re = \frac{\rho \left( \frac{D_{\text{tip}}}{2} \right) \omega j}{2 \mu} \]  

(99)
The friction coefficient is then determined as

\[ \begin{align*}
\text{For } Re \leq 2,000 & \quad c_f = \frac{16}{Re} \\
\text{For } Re > 2,000 & \quad c_f = \frac{0.0791}{Re^{1/4}}
\end{align*} \] (100) (101)

9.3 Partial Admission Loss

Full admission axial turbines will in general yield higher efficiencies, however circumstances arise that don’t permit full admission such as low mass flow rates. In cases were full admission is not possible, partial admission turbines are used. Partial admission turbines incur performance penalties that can be significant. The partial admission losses are divided into two categories, pumping losses and sector losses. The pumping losses refer to inactive blades rotating in a fluid filled casing (Roelke 1994) as well as flow from active blades re-entering inactive blade passages (Aungier 2006). The sector losses refer to the stagnant fluid in inactive blades having to be accelerated by the fluid from stator nozzles as the inactive blade enters the active arc. The sector loss encompasses the loss from high velocity fluid in active blades decelerating as the blade leaves the active arc. In partial admission machines if all stator nozzles are grouped together there will only be one sector but in cases where the stators nozzles are spaced apart there are multiple sectors and the sector losses proportional to the number of active stator sectors. More sectors induce greater losses.

9.3.1 Sector Losses

Roelke (1994) cites findings reported by Stenning (1953) which investigated the effects of partial admission on axial turbine performance across a range of admission rates for predicting sector losses. The model is based upon fitted data between 12% to 100% admission rates. The sector loss coefficient is described as

\[ K_{sector} = U Y_{II} \sin \beta_{II} (1 + K_w K_s) \] (102)

where \( K_s \) is the rotor velocity coefficient, and \( K_w \) is the rotor relative velocity ratio.

\[ K_s = 1 - \frac{p}{3\lambda} \] (103)

\[ K_w = \frac{Y_{III}}{Y_{II}} \] (104)
However this model seemed to under predict the losses. This was reported by Cho (2006) et al in a paper on the performance prediction on partially admitted turbines. Similar findings were encountered in this study. Aungier (2006) reported that the model published by Suter and Traupel is about the best model available to describe the sector losses. Varma and Soundranayagam (2012) reported that the theory of Stenning’s sector loss combined with Suter and Traupel’s empirical correlation for the effect of multiple sectors provided reasonably accurate results of sector losses for their experimental study on low aspect ratio axial turbines. They also showed experimentally that increasing the number of sectors did increase the losses as predicted by the Suter and Traupel model. Taking into the account the findings of Varma et al. the $Ks$ factor becomes

$$K_s = 1 - \frac{N_{sector} \rho}{3\lambda}$$

(106)

where $N_{sector}$ is the number of active sectors. This sector loss model provided a good fit for the experimental data gathered in this investigation. The addition of the multiplicative $N_{sector}$ term to $Ks$ allowed the model to work for both the single and dual nozzle case experimentally tested. The implication of $N_{sector}$ on $K_s$ is that for partial admission machines all stator nozzles should be grouped together so that only one active sector exists. The greater the number of active sectors the greater the sector losses.

9.3.2 Pumping Losses
Aungier (2006) and Roelke (1994) both published models for partial admission pumping losses that are very similar. The difference in the two models is that Augnier’s model does not place as much emphasis on blade height by raising $h_{r,tt}$ to the power of 1.5 as in Roelke’s model. Roelke’s model was found to yield a better match to the experimental data for the axial machine tested in this investigation and is implemented into the loss model. Also, as Varma noted that multiple sectors influenced the sector losses, it was found that pumping losses were also influenced by multiple sectors. A multiplicative term (number of active sectors) was added to the model as $N_{sector}$. The partial admission pumping losses (with the addition of $N_{sector}$) are expressed as

(Roelke) \[ L_{pumping} = \Phi_{pumping} \left( N_{sector} 3.63 \rho \beta_l \beta_t U_m^{\gamma_{partial}} h_{r,tt}^{1.5} (1 - \epsilon) \right) \]

(107)

and $\epsilon$ is the active fraction expressed as follows;
\[ \epsilon = \frac{\lambda}{\pi D_{\text{mean}}} \]  

(108)

9.4 Trailing Edge Losses

The trailing edge thickness of the blade causes the flow to separate at two points. Between these two points there is a region where the pressure is significantly lower than the freestream pressure. Downstream of the trailing edge, this low pressure region merges with the boundary layers from the pressure and suction surfaces of the blade to form a single wake, which then dissipates into a single stream through shear (Baines 1997). This is illustrated in Figure 46.

![Figure 46. Illustration of trailing edge wake](image)

Okapuu and Kacker (Kacker 1982) generated a relationship for the trailing edge loss in terms of an energy coefficient versus trailing edge thickness to throat opening ratio. They published relationships for two distinct cases, one for an axial entry (\( \beta_{II} = 0 \)) and one for an impulse blade (\( \beta_{II} = \beta_{III} \)). The curves published for the two cases are presented here as polynomials fitted to their results. The energy coefficient for the case of axial entry is

\[ \xi_{\text{trail}, \beta_{II}=0} = 0.275 \left( \frac{te}{o} \right)^2 + 0.080 \frac{te}{o} \]  

(109)

and the energy coefficient for the impulse case is

\[ \xi_{\text{trail}, \beta_{II}=\beta_{III}} = 0.478 \left( \frac{te}{o} \right)^2 + 0.158 \frac{te}{o} \]  

(110)

For all cases between these two cases the energy coefficient can be found by interpolation between the two cases as follows;

\[ \xi_{\text{trail}} = \xi_{\text{trail}, \beta_{II}=0} + \left( \frac{\beta_{II}}{\beta_{III}} \right)^2 \left( \xi_{\text{trail}, \beta_{II}=\beta_{III}} - \xi_{\text{trail}, \beta_{II}=0} \right) \]  

(111)
The energy coefficient is the difference between the actual enthalpy and the equivalent ideal process expanding through the same pressure ratio defined as (Baines 1997)

$$\xi_{\text{trail}} = \frac{H_2 - H_{2s}}{\frac{1}{2} Y_{III}^2}$$ (112)

and the power loss due to trailing edge effects is expressed as

$$L_{\text{trail}} = \Phi_{\text{trail}} \frac{1}{2} m \xi_{\text{trail}} Y_{III}^2$$ (113)

9.5 Incidence Losses

Incidence losses occur when the gas flow angle differs from the designed blade angle (absolute angle for stators and relative angles for rotors). Figure 47 shows the nomenclature used to describe incidence.

![Figure 47. Rotor blade incidence nomenclature according to Roelke (Glassman 1994)](image)

The incidence angle is defined as

$$i = \beta_f - \beta$$ (114)

The incidence angle may be positive or negative and the sign of the incidence is important as Roelke (1994) showed that variation of incidence loss differs for positive and negative incidence angles. Roelke also showed that minimum loss does not occur at zero incidence but an optimum incidence, $i_{opt}$, usually between -4° to -8°.

The kinetic energy loss due to incidence is described as
\[ K_{\text{incidence}} = \frac{Y_I^2}{2} (1 - \cos(i - i_{\text{opt}})^n) \]  
\[ n = \begin{cases} 
2 & \text{for } i < 0 \\
3 & \text{for } i \geq 0 
\end{cases} \]

and the power loss due to incidence is then described as

\[ L_{\text{incidence}} = \Phi_{\text{incidence}} m K_{\text{incidence}} \]  

### 9.6 Secondary Loss

The secondary loss model employed here is similar to the AMDC model reported by Dunham and Came as revised by Kacker and Okapuu as described by Aungier (2006). This model developed to provide a more accurate loss at low aspect ratios by the introduction of an aspect ratio correction factor. The Aungier method also employs a Reynolds number correction factor \((K_{Re})\) and a compressibility correction factor \((K_s)\) to account for extreme off-design conditions without predicting excessive loss values. The Aungier modified secondary loss coefficient is as follows

\[ \xi_{\text{secondary}} = K_{Re} K_s \left( \frac{\xi'_{\text{secondary}}}{\sqrt{1 + 7.5 \xi'_{\text{secondary}}}} \right)^2 \]  

where the unmodified secondary loss coefficient is

\[ \xi'_{\text{secondary}} = 0.0334 F_A Z \frac{\sin(\beta_{u,II})}{\sin'(\beta'_{u,II})} \]  

\[ F_A = \frac{c}{h}; \frac{h}{c} \geq 2 \]  

\[ F_A = 0.5 \left( \frac{2c}{h} \right)^{0.7}; \frac{h}{c} < 2 \]

Like the AMDC model, the secondary loss coefficient is a function of the Ainley loading parameter \((Z)\), the lift coefficient \((C_L)\) and mean flow angle \((\bar{\beta})\) which are as follows;

\[ Z = C_L^2 \left( \sin(\beta_{u,III})^2 / \sin(\beta_u)^3 \right) \]  
\[ C_L = 2[(1/\tan(\beta_{u,II})) - (1/\tan(\beta_{u,III}))] \]  
\[ \bar{\beta} = \frac{\pi}{2} - \tan^{-1}(1/\tan(\beta_{u,II} + \beta_{u,III}))/2 \]
The modified compressibility factor as described by Aungier includes a term in the denominator to prevent excessive $K_s$ values for extreme cases of where the axial chord to blade height ($c_x/h_{r,tt}$) is very large.

$$K_s = 1 - \frac{(1 - K_p) \left( \frac{c_x}{h_{r,tt}} \right)^2}{1 + \left( \frac{l}{h_{r,tt}} \right)^2}$$

(124)

$K_s$ is a function of the unmodified compressibility factor ($K_p$) which is as follows;

$$K_p = 1 - (1 - K_1)X^2$$

(125)

where $K_p$ is based on

(125a)

$$K_1 = 1 - 0.625(\bar{M}_{III} - 0.2 + |\bar{M}_{III} - 0.2|)$$

(125b)

$$X = \frac{2\bar{M}_{II}}{\bar{M}_{II} + \bar{M}_{III} + |\bar{M}_{III} - \bar{M}_{II}|}$$

(125c)

$$\bar{M}_{III} = \frac{1}{2} (M_{III} + 1 - |M_{II} - 1|)$$

(125d)

$$\bar{M}_{II} = \frac{1}{2} (M_{II} - 0.566 + |0.566 - M_{II}|)$$

(125e)

$$M_{III} = \frac{Y_{III}}{a_{III}}$$

$$M_{II} = \frac{Y_{II}}{a_{II}}$$

The Reynolds number correction factor is as follows;

(126)

For $Re_c < 1 \times 10^5$

$$K_{Re} = \sqrt{(1 \times 10^5)/Re_c}$$

For $1 \times 10^5 < Re_c < 5 \times 10^5$

$$K_{Re} = 1$$

For $Re_c > 5 \times 10^5$

$$K_{Re} = [\log_{10}(5 \times 10^5)/\log_{10} Re_c]^{2.58}$$

$$Re_c = \rho_{III} Y_{III} c / \mu_{III}$$

Based upon the secondary loss coefficients the secondary losses are expressed as
\[ L_{\text{secondary}} = V_{u,III} \, m \left( \xi_{\text{secondary}} - 1 \right) \]  

(127)

### 9.7 Profile Loss

As the fluid passes through the blade passage there is an inherent loss due to friction. In an axial rotor, the blade passage can be modelled as a pipe. The profile loss coefficient is calculated as follows:

\[ K_{\text{profile}} = f_c \left( l/D_h \right) \left( \tilde{v}^2 / 2 \right) \]  

(128)

Where the hydraulic diameter for a rectangular channel is

\[ D_h = \frac{2 \, w_{r,tt} \, h_{r,tt}}{w_{r,tt} + h_{r,tt}} \]  

(129)

If the flow is laminar (\( Re < 2100 \)), then the Fanning friction factor is found by

\[ f = \frac{16}{Re} \]  

(130)

However, all the flows in these analyses are well within the turbulent regime so the Fanning friction factor is found by using the Colebrook-White equation and solved using the bisection method (Ventura et al. 2012).

\[ \frac{1}{\sqrt{f}} = -4 \log_{10} \left( \frac{k_r}{D_h} \frac{3.7}{3.7} + \frac{1.256}{Re \sqrt{f}} \right) \]  

(131)

The profile loss is then expressed as

\[ L_{\text{profile}} = \Phi_{\text{profile}} \, K_{\text{profile}} \, m \]  

(132)

In this loss model the profile loss and the secondary loss are coupled together in a single loss term called the passage loss where \( \Phi_{\text{passage}} \) is the machine specific linear clearance loss coefficient. The passage loss is then expressed as

\[ L_{\text{passage}} = \Phi_{\text{passage}} \left( L_{\text{profile}} + L_{\text{secondary}} \right) \]  

(133)

### 9.8 Supersonic Shock and Expansion Loss

When the fluid passing through the blade reaches supersonic speeds there are losses associated with shock and expansion. Aungier (2006) presented a model that was a modification of the AMDC-KO
model that seeks to include the influence of diffusion at far off-design conditions. The shock loss coefficient is calculating by imposing as asymptotic upper limit of 1 as follows

\[ K_{\text{shock}} = \frac{K'_{\text{shock}}}{\sqrt{1 + K'_{\text{shock}}^2}} \]  

(134)

where the preliminary shock loss coefficient is

\[ K'_{\text{shock}} = 0.8X_{\text{shock}}^2 + X_{\text{diffusion}}^2 \]  

(135a)

And the shock and diffusion coefficients are

\[ X_{\text{shock}} = 0; \quad M_{II} \leq 0.4, \quad X_{\text{shock}} = M_{II} - 0.4; \quad M_{II} > 0.4 \]  

(136b)

\[ X_{\text{diffusion}} = 0; \quad M_{II} \leq M_{III}, \quad X_{\text{diffusion}} = \frac{M_{II}}{M_{III}} - 1.0; \quad M_{II} > M_{III} \]

Aungier also suggested a loss coefficient to account for over-expanded flow at the blade outlet due to supersonic discharge mach numbers. The supersonic expansion loss coefficient is calculated as

\[ K_{\text{expansion}} = \left( \frac{M_{III} - 1}{M_{III}} \right)^2 \]  

(137)

The shock and expansion loss coefficients are coupled together and provide a supersonic loss as

\[ L_{\text{super}} = \phi_{\text{super}} V_{u,III} m (K_{\text{shock}} + K_{\text{expansion}} - 1) \]  

(138)

where \( \phi_{\text{super}} \) is the machine specific linear clearance loss coefficient

9.9 Bearing Losses

All turbines will need to incorporate some sort of bearing to support the rotating shaft and permit rotation. Depending on the style of bearing there will be an associated loss. In this investigation deep groove roller ball bearings with seals from SKF were used to support the rotating shaft for the first set of tests and then the bearings were changed to deep groove roller ball ceramic bearings with no seals. The losses due to friction in the bearings according to SKF (SKF) are determined as follows

\[ L_{\text{bearing}} = M_{\text{bearing}} \omega \]  

(139)

\[ M_{\text{bearing}} = \left( \mu \frac{F_{\text{bearing}} D_{\text{bearing}} M_{\text{seal}}}{2} \right) \]  

(140)

where \( F_{\text{bearing}} \) is the radial load on bearing, \( D_{\text{bearing}} \) is the core diameter of bearing and \( M_{\text{seal}} \) is the friction moment from the bearing seals. For the ceramic bearings \( M_{\text{seal}} \) is zero.
9.10 Fitting and Use of the Loss Model

The loss model described in this chapter was used to write a program for calculating losses in the stator and rotor passages of an axial turbine. The loss model program was short named SSAL (Single Stage Axial Losses). The loss models calibration based on experimentation is described in the next chapter (Chapter 10) and the loss models incorporation into the ORCCA is described in Chapter 11. This loss model is to be used in conjunction with AXIAL which produces geometry for a single stage supersonic axial impulse turbine. Therefore the calibration of SSAL will be based on experimental results for a turbine of this type and will not necessarily extend to other types of turbomachinery.
Chapter 10  Turbine Experimentation

Experimentation was conducted using the impulse turbine previously discussed in the ORC test apparatus. The turbine was tested across a range of conditions for a total of 24 different test sets as listed in Table 16.

<table>
<thead>
<tr>
<th>Working Fluid</th>
<th>Number of Stator Nozzles</th>
<th>Mass Flow Rate (kg/s)</th>
<th>RPM</th>
<th>Turbine Inlet Temperature (°C)</th>
<th>Turbine Inlet Pressure (kPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R245FA</td>
<td>1</td>
<td>0.035 - 0.044</td>
<td>0 - 5,000</td>
<td>140</td>
<td>1,400 – 1,800</td>
</tr>
<tr>
<td>R245FA</td>
<td>2</td>
<td>0.070 – 0.090</td>
<td>0 – 5,000</td>
<td>140</td>
<td>1,400 – 1,800</td>
</tr>
<tr>
<td>R134A</td>
<td>1</td>
<td>0.035 – 0.043</td>
<td>0 – 4,000</td>
<td>140 - 145</td>
<td>1,500 – 1,800</td>
</tr>
<tr>
<td>R134A*</td>
<td>2</td>
<td>0.070 – 0.090</td>
<td>0 – 4,000</td>
<td>140</td>
<td>1,500 – 1,800</td>
</tr>
</tbody>
</table>

*The results from this data set were not used in the loss model analysis because the turbine outlet pressure was too high to achieve the designed expansion through the stator. The high outlet pressure was a result of friction through the condensers and the limited cooling water minimum temperature.

Note that for a dual nozzle R134a test scenario quality measurements were not able to be attained. Due to the higher mass flow rate of the dual nozzle case, increased friction induced pressure drop through the condensers and pipe work did not permit the required stator exit pressure to achieve supersonic conditions. If lower cooling water temperatures were available this case may have been run but, at present, the ORC test loop is not fitted with a chiller. The cooling water is delivered at ambient conditions and in Queensland the ambient temperatures are relatively high. The average cooling water temperature was 24°C.

Each test performed was carried out across a range of speeds. The speed was regulated by controlling the applied torque with a hysteresis brake. The turbine was tested from its maximum speed (no applied torque) to the minimum speed before stall and varying speed steps in between. Each speed step was held constant until stable pressure, temperature, and speed readings were obtained.

**Single Nozzle R134a Case**

To illustrate the experimentation and data analysis processes undertaken a single nozzle R134a case will be used as an example. The general conditions of the example case are as follows:

<table>
<thead>
<tr>
<th>Working Fluid</th>
<th>Number of Stator Nozzles</th>
<th>Mass Flow Rate (kg/s)</th>
<th>RPM</th>
<th>Turbine Inlet Temperature (°C)</th>
<th>Turbine Inlet Pressure (kPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R134A</td>
<td>1</td>
<td>0.036</td>
<td>0 – 4,000</td>
<td>143</td>
<td>1,500</td>
</tr>
</tbody>
</table>
The acquired speed and work versus time data is shown in Figure 48. As can be seen the speed was varied in steps and held constant for a period of time until stable readings were achieved. Thus, for each test set there are a number of operating points.

A post processing program (called 3D_Eilmer_Post.py with complete source code in Appendix B) was written to filter and gather data so that it could be analysed in a loss model program. The collected data was filtered to only include stable operating conditions as shown by the hollow markers in Figure 48.

Since the stator exit velocity is used in several components of the loss model and the velocity is not directly measure, the pressure and temperature measurements associated with each stable point were used to obtain the nozzle exit velocity by using a 3D CFD model of the stator nozzle. The CFD code used is an in-house code called Eilmer3.

“Eilmer3 is an integrated collection of programs for the simulation of transient, compressible flow in two and three spatial dimensions. It provides a preparation program that can be used to set up a database of simulation parameters, a block-structured grid defining the flow domain and an initial flow field. These items are then used as a starting point for the main simulation program which computes a series of snapshots of the evolving flow” (Jacobs 2010)

The stator geometry in the CFD simulation was built using the 3D paths generated in the axial design program. The fluid properties used in the simulation are taken from Refprop (Lemmon E.W. 2007). For each test case the measured pressure and temperature are used to define boundary conditions. The nozzle is divided into two discreet blocks. The first block is the subsonic...
convergent portion of the nozzle and the second block is the divergent supersonic portion of the nozzle.

![Figure 49. Eilmer3 CFD results of a stator nozzle passage. (Top Left) Pressure, (Top Right) Temperature, (Bottom Left) Velocity Vector, (Bottom Right) Mach Number](image)

The surfaces of the blocks are transfinite-interpolation surfaces between the four paths (as shown in Figure 41) that comprise the surface. The block volumes are bound by six surfaces. The walls of the nozzle passage are defined as adiabatic wall boundaries, the inlet of the nozzle is defined as a subsonic inlet boundary and the outlet of the nozzle is defined as fixed outlet pressure condition. The adiabatic wall condition enables viscous effects to impose no-slip at the wall but does not allow for heat transfer. The subsonic inlet condition assumes subsonic flow with the stagnation temperature and pressure specified from the experimentally measured inlet conditions. The fixed outlet pressure condition sets a back pressure that is set equal to the average experimentally measured pressure.

Figure 49 shows images of the CFD results from an Eilmer3 simulation for a single nozzle passage for this R134a case. The CFD results are then compared to the measured results to verify consistency. Figure 50 shows the measured stator inlet and outlet pressure (top) and the Eilmer3 calculated pressures.
The measured data (Top) is the pressure for the entirety of the test. The calculated pressure (Bottom) is a plot of the CFD simulation finding a convergent solution for a steady state solution. For agreement between the measured and the calculated to be made the average measured values should be equal to the convergent calculated values.

The same is true for the temperature and mass flow rate as shown in Figure 51 and Figure 52.
The fluctuation in the measured temperature is a result of the control system on the immersion heater. It is only able to maintain +/- 3°C from the set point. But the average measured temperature is closely matched to the convergent CFD solution.
In the CFD simulation the only inputs are the inlet pressure and temperature and outlet pressure. Because the CFD results for mass flow rate and exit temperature agree with the measured results the CFD results can be viewed with a high degree of confidence. From the CFD results, the parameter of most interest is the stator exit velocity. For this R134a case the calculated convergent exit velocity solution is shown in Figure 53.
For all experimental data sets the data is examined as described above to determine the stator exit velocity for each test case. The post processed experimental data is then used in a loss model program.

10.1 Loss Model Validation

The gathered, post-processed experimental data is presented as measured work versus measured speed and is compared against the loss model programs calculated work versus measured speed using the calculated nozzle exit velocity and recorded pressures and temperatures as inputs for the
loss model program. The loss model is then calibrated against experimental data by determining machine specific coefficients (ϕ and γ). An optimization method inspired by the genetic algorithm technique was used to find the coefficients to yield the best fit between the experimental work data and the work calculated by the loss model. Figure 54 shows the simplified algorithm of the coefficient search method.

![Diagram of coefficient search algorithm](image)

Figure 54. Machine specific loss coefficient search algorithm

The search is designed to minimise the error between the measured and the calculated work at the minimum speed, maximum work and maximum speed condition. The fitness function for the search is the average of the error at these three points. These three points define the shape of the work versus speed curve. By using these points, each iteration is improved incrementally by adjusting the machine specific loss coefficients according to the error. For loss mechanisms that
have a predominantly low speed effect the loss coefficients are adjusted according to the minimum speed error and similarly the maximum work and maximum speed mechanisms are adjusted according to error at maximum work and maximum speed conditions.

For every iteration, a population of loss coefficients is generated. The initial population is based on default values as shown in Figure 54. Each coefficient population is defined by an array that is bound by a constrained but random maximum and minimum (defined in Figure 54 in the “Define Coefficient Ranges” box). The ranges are constrained to avoid implausible solutions and the ranges are given a degree of randomness to mitigate the possibility of getting stuck in local minimums. For each new population the coefficients that yield the lowest fitness function value are kept and added to the next population. As the search progresses and the error decreases, the coefficient ranges decrease so as to increase the resolution of the search as it progresses.

The progress of the coefficient search is plotted and continues until no further improvements can be found. All experimental data was consolidated and a coefficient search was performed on the entire data set. Figure 55 shows the progress of the search in terms of the error fitness function value versus iteration.

![Figure 55. R134a Case (1 Nozzle): Loss model coefficient search progress](image)

From the plot it can be seen that the progress asymptotically approaches a limit. Once that limit is approached the search is terminated and the set of coefficients that yield the best fit are saved.

The calculated work is a function of the ideal work (based on the velocity triangle at a respective speed) minus the losses. Each of the individual loss mechanisms were calculated in terms of power and plotted against speed to see how each individual loss mechanism is related to speed. Figure 56 shows the calculated losses for each mechanism versus RPM for the best fit case.
The resulting work versus speed plot is then created for the best fit to compare the measured work versus speed against the calculated work versus speed. Figure 57 shows the best fit case of measured work to calculated work for this R134a example.

The coefficient search method just described was performed on all experimental data sets. All data for the R245fa tests (both single and dual nozzle cases) and the single nozzle R134a test cases were gathered and analysed for loss coefficients. The average coefficients for all cases were used to fit the measured work versus calculated work. Figure 58 shows the best coefficient fit for all the single nozzle R134a case and Figure 59 shows the best coefficient fit results of the measured work versus the calculated work for all the R245fa cases. The two distinct clusters in the R245fa data represent the admission cases, a single nozzle and a dual nozzle case.

The tested turbine had significant losses and can be fully appreciated when looking at the design conditions compared to the tested conditions (Table 17). The designed operating speed was 10,000RPM but the actual operating speed was only 5,000RPM. The designed Z for maximum
work was to 0.5 but the actual was 0.15. The turbine was designed based solely on velocity triangles which is why the operating conditions vary so much between the design and the tested. The intent of this first turbine was to be a machine designed from ideal conditions that would have significant losses. The machine was to be a test platform for investigating losses.

![Figure 58. Gathered R134a case loss coefficient search: Measured work versus calculated work](image)

![Figure 59. Gathered R245fa single nozzle and dual nozzle loss coefficient search. Measured work versus calculated work](image)

Table 17. Comparison of turbine design conditions and test conditions

<table>
<thead>
<tr>
<th>Turbine Parameter</th>
<th>Design Conditions</th>
<th>Tested Conditions</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluid</td>
<td>R134a</td>
<td>R134a and R245fa</td>
<td></td>
</tr>
<tr>
<td>Working Fluid Mass Flow Rate</td>
<td>( m ) 0.05</td>
<td>0.035-0.090</td>
<td>kg/s</td>
</tr>
<tr>
<td>Stator Inlet Temperature</td>
<td>( T_I ) 140</td>
<td>140 – 145</td>
<td>°C</td>
</tr>
<tr>
<td>Stator Inlet Pressure</td>
<td>( P_I ) 2.500</td>
<td>1,400-1,800</td>
<td>kPa</td>
</tr>
<tr>
<td>Stator Outlet Temperature</td>
<td>( T_{II} ) 100</td>
<td>107 - 112</td>
<td>°C</td>
</tr>
<tr>
<td>Stator Outlet Pressure</td>
<td>( P_{II} ) 812</td>
<td>380 - 600</td>
<td>kPa</td>
</tr>
<tr>
<td>Speed-Work Parameter Range</td>
<td>( Z ) range 0.50</td>
<td>0.025 – 0.220</td>
<td></td>
</tr>
<tr>
<td>Maximum RPM</td>
<td>( RPM ) 10,000</td>
<td>0 – 5,000</td>
<td>RPM</td>
</tr>
<tr>
<td>Number of Nozzles</td>
<td>( N_{nozzles} ) 2</td>
<td>1-2</td>
<td></td>
</tr>
<tr>
<td>Stator Exit Mach Number</td>
<td>( M_{II} ) 1.45</td>
<td>1.30 – 1.60</td>
<td></td>
</tr>
</tbody>
</table>
The averaged best fit loss coefficients for all the test cases are listed in Table 18. These best fit coefficients are for the impulse turbine tested in this thesis. Deviation from the default values suggests that the loss models are sensitive to flow phenomenon and geometric conditions not accounted for in the equations describing the loss mechanism. This coincides with what Moustapha (2003) et al. noted, that a mean-line loss cannot reproduce the full complexity of the flow in a real turbine and therefore machine specific loss coefficients obtained experimentally will be needed to accurately model the performance of a turbine.

Table 18. Average Loss Coefficients For All Test Cases

<table>
<thead>
<tr>
<th>Loss Coefficient</th>
<th>R134a Single Nozzle</th>
<th>R245fa Single and Dual Nozzle</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Phi_{\text{clearance}} )</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>( \Phi_{\text{windage}} )</td>
<td>1.02</td>
<td>1.11</td>
<td>1.07</td>
</tr>
<tr>
<td>( \Phi_{\text{sector}} )</td>
<td>1.02</td>
<td>1.11</td>
<td>1.07</td>
</tr>
<tr>
<td>( \Phi_{\text{partial}} )</td>
<td>1.00</td>
<td>1.11</td>
<td>1.07</td>
</tr>
<tr>
<td>( \Phi_{\text{trail}} )</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>( \Phi_{\text{incidence}} )</td>
<td>1.00</td>
<td>0.89</td>
<td>0.95</td>
</tr>
<tr>
<td>( \Phi_{\text{passage}} )</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>( \Phi_{\text{super}} )</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>( \gamma_{\text{windage}} )</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
</tr>
<tr>
<td>( \gamma_{\text{partial}} )</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
</tr>
</tbody>
</table>

For an initial mean line analysis, non-calibrated loss model will provide rough estimates of turbine performance but for more accurate results the loss model should be calibrated for the specific machine of interest.

**Note on Partial Admission Losses and Number of Sectors**

The original loss model did not take into account the findings of Varma et. al. regarding the influence of the multiple sectors on partial admission losses. Without the addition of the multiplicative term \( N_{\text{sector}} \) to sector loss as Varma suggested the loss model needed machine specific loss coefficients that deviated greatly from the default values.

While trying to fit the loss model to the experimental data Varma’s suggestion of the addition of \( N_{\text{sector}} \) was implemented to the sector loss model. This improved the results of the fitting the loss model to the experimental data. But even with this addition the loss model required significant machine specific loss coefficients to fit the experimental data.

It was found that the addition of the \( N_{\text{sector}} \) term to the pumping losses, in addition to the sector losses, improved the models fit. The addition of \( N_{\text{sector}} \) to pumping losses improved the fit as well as reduced the loss models reliance on machine specific coefficients. The machine specific los
coefficients found from the search algorithm are much closer to the default values and produce a better fit to the experimental data.

As a comparison between the different implementations of \( N_{\text{sector}} \), Figure 60 shows three different measured work versus calculated work for experimental data of 1 and 2 nozzle, R245fa data at 0.035 and 0.070kg/s mass flow rate.

![Figure 60](image)

Figure 60. Comparison of measured work versus calculated work for no \( N_{\text{sector}} \) modification (top left), \( N_{\text{sector}} \) modification to sector loss only (top right) and \( N_{\text{sector}} \) modification to sector and pumping losses (bottom)

The non-modified case has a fairly poor fit and relies heavily on the loss coefficients. The case with the modified sector only has a slightly better fit and relies less on the coefficients. And the case with both sector and pumping losses modified by \( N_{\text{sector}} \) is the best fit and the loss coefficients have a lower deviation from the default values as listed in Table 19.

<table>
<thead>
<tr>
<th>Loss Coefficient</th>
<th>No Modification</th>
<th>Sector Only Modified</th>
<th>Sector and Pumping Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Phi_{\text{clearance}} )</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>( \Phi_{\text{windage}} )</td>
<td>1.95</td>
<td>1.52</td>
<td>1.17</td>
</tr>
<tr>
<td>( \Phi_{\text{sector}} )</td>
<td>1.95</td>
<td>1.52</td>
<td>1.17</td>
</tr>
<tr>
<td>( \Phi_{\text{partial}} )</td>
<td>1.95</td>
<td>1.52</td>
<td>1.17</td>
</tr>
<tr>
<td>( \Phi_{\text{trail}} )</td>
<td>0.88</td>
<td>0.92</td>
<td>1.00</td>
</tr>
<tr>
<td>( \Phi_{\text{incidence}} )</td>
<td>0.63</td>
<td>0.72</td>
<td>0.82</td>
</tr>
<tr>
<td>( \Phi_{\text{passage}} )</td>
<td>0.88</td>
<td>0.87</td>
<td>0.91</td>
</tr>
<tr>
<td>( \Phi_{\text{super}} )</td>
<td>0.88</td>
<td>0.92</td>
<td>1.00</td>
</tr>
<tr>
<td>( \gamma_{\text{windage}} )</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
</tr>
<tr>
<td>( \gamma_{\text{partial}} )</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
</tr>
<tr>
<td>Error</td>
<td>32%</td>
<td>30%</td>
<td>22%</td>
</tr>
</tbody>
</table>
10.2 ORCCA versus Experimental Data

Several data points were pulled from the turbine test data to compare cycle points measured during turbine testing and the cycle points calculated by ORCCA. The measured parameters that were inputs to ORCCA were the hot oilers inlet temperature to the evaporator, the turbine shaft power, the turbine inlet and outlet pressures and the measured heat exchanger measured temperature differentials. Iterations were run in ORCCA adjusting the turbine thermodynamic and mechanical efficiencies until the working fluid mass flow rate and the turbine outlet temperature matched the experimental data. A tabulated comparison of the three cases is listed in Table 20.

The error in the experimental data compared to the ORCCA results is in general relatively low, less than 5%. Given the tolerance of the thermocouples used was +/-1.5°C. However there was a fairly large difference in the measured regeneration outlet temperatures compared to ORCCA.

Reasons for this could have been related to measurement error. But it is believed that the high cold side regeneration outlet temperature readings are a result of the sensors proximity to the evaporator.

Table 20. Experimental ORC Data versus ORCCA calculated values

<table>
<thead>
<tr>
<th>Test Set Name</th>
<th>Variable</th>
<th>ORCCA</th>
<th>Error</th>
<th>Data</th>
<th>ORCCA</th>
<th>Error</th>
<th>Data</th>
<th>ORCCA</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Time Point</td>
<td>Working Fluid</td>
<td>20120605_R245_044_140</td>
<td>1338875776.16</td>
<td>R245fa</td>
<td>R245fa</td>
<td>20120612_R245_038_140</td>
<td>1339478691.13</td>
<td>R245fa</td>
<td>R245fa</td>
</tr>
<tr>
<td>CH1: Pump Out</td>
<td>T2</td>
<td>28.79</td>
<td>26.90</td>
<td>6.6%</td>
<td>28.30</td>
<td>26.29</td>
<td>7.1%</td>
<td>26.35</td>
<td>25.15</td>
</tr>
<tr>
<td>CH2: Regen Cold Out</td>
<td>T11</td>
<td>81.76</td>
<td>62.70</td>
<td>23.3%</td>
<td>78.79</td>
<td>66.50</td>
<td>15.6%</td>
<td>76.66</td>
<td>64.86</td>
</tr>
<tr>
<td>CH3: Evaporator Out</td>
<td>T5</td>
<td>142.59</td>
<td>142.60</td>
<td>0.0%</td>
<td>140.76</td>
<td>140.76</td>
<td>0.0%</td>
<td>140.51</td>
<td>140.66</td>
</tr>
<tr>
<td>CH4: Turbine In</td>
<td>T5</td>
<td>141.98</td>
<td>142.60</td>
<td>0.4%</td>
<td>141.00</td>
<td>140.76</td>
<td>0.2%</td>
<td>142.96</td>
<td>140.66</td>
</tr>
<tr>
<td>CH7: Turbine Out</td>
<td>T6</td>
<td>109.53</td>
<td>108.90</td>
<td>0.6%</td>
<td>109.65</td>
<td>110.17</td>
<td>0.5%</td>
<td>107.13</td>
<td>107.64</td>
</tr>
<tr>
<td>CH8: Regen Hot Out</td>
<td>T9</td>
<td>67.53</td>
<td>63.90</td>
<td>5.4%</td>
<td>63.61</td>
<td>58.40</td>
<td>8.2%</td>
<td>61.59</td>
<td>56.00</td>
</tr>
<tr>
<td>CH9: Water In</td>
<td>Tci</td>
<td>17.16</td>
<td>17.89</td>
<td>4.3%</td>
<td>16.54</td>
<td>16.69</td>
<td>0.9%</td>
<td>17.40</td>
<td>17.55</td>
</tr>
<tr>
<td>CH10: Water Out</td>
<td>Tco</td>
<td>19.74</td>
<td>20.03</td>
<td>1.5%</td>
<td>19.61</td>
<td>19.45</td>
<td>0.8%</td>
<td>19.74</td>
<td>19.95</td>
</tr>
<tr>
<td>CH11: Oil In</td>
<td>Thi</td>
<td>146.14</td>
<td>146.14</td>
<td>0.0%</td>
<td>148.34</td>
<td>148.49</td>
<td>0.1%</td>
<td>147.24</td>
<td>147.39</td>
</tr>
<tr>
<td>CH12: Oil Out</td>
<td>Tcho</td>
<td>135.39</td>
<td>134.65</td>
<td>0.5%</td>
<td>138.32</td>
<td>138.11</td>
<td>0.2%</td>
<td>137.59</td>
<td>137.68</td>
</tr>
</tbody>
</table>

| CH2: Turbine In | P5         | 1.79         | 1.79 | 0.0%          | 1.44         | 1.44 | 0.0%          | 1.48         | 1.48 | 0.0% |
| CH3: Turbine Out | P6        | 0.51         | 0.51 | 0.6%          | 0.43         | 0.43 | 0.4%          | 0.40         | 0.40 | 0.4% |
| CH4: Pump In | P1         | 0.50         | 0.51 | 1.5%          | 0.43         | 0.43 | 0.0%          | 0.40         | 0.40 | 0.4% |
| CH5: Pump Out | P2         | 1.81         | 1.79 | 0.7%          | 1.45         | 1.44 | 0.8%          | 1.49         | 1.48 | 0.7% |
| CH5: Power (W) | Power     | 138.19       | 140.00 | 1.3%          | 101.13       | 101.00 | 0.1%          | 95.31        | 95.00 | 0.3% |
| CH1: Mass Flow (kg/s) | mf | 0.043       | 0.044 | 2.4%          | 0.038       | 0.038 | 0.1%          | 0.036        | 0.035 | 3.2% |
inlet. The regenerations outlet was nearly adjacent to the inlet of the evaporator. Heat could be seen to be conducted through the pipe work while the loop was warming up. Conducted heat from the evaporator could be skewing the regeneration outlet temperatures. However, the main the focus of the experimentation was the turbine. The instrumentation on the turbine was a greater distance from other components and the pipe work around the turbine and the turbine itself were well insulated. The experimental data and ORCCA data agree closely for the data upstream and downstream of the turbine.

10.3 Remarks on the Loss Model Coefficients

The calibrated loss model presented here provided a good fit to the measured data. It predicted the correct trend of work versus speed and the magnitude of the calculated work compared well with the measured work.

The deviation of the loss coefficients from the default values of one for linear coefficients and three for exponential can be attributed to the fact that each individual loss mechanism is a complicated flow phenomenon and machine specific geometries will influence the losses beyond the ability and scope of the loss models. The loss model calculations, however well devised they are, are still a simplified interpretation of a complex issue.

With the addition of the $N_{sector}$ term to the loss model the reliance of the loss model on the coefficients is significantly reduced. The deviation from the default values was 0% for 6 out of 10 coefficients ($\Phi_{\text{clearance}}$, $\Phi_{\text{trail}}$, $\Phi_{\text{passage}}$, $\Phi_{\text{super}}$, $\gamma_{\text{windage}}$, $\gamma_{\text{partial}}$) 5% for $\Phi_{\text{incidence}}$, and 7% for 3 out of 10 ($\Phi_{\text{clearance}}$, $\Phi_{\text{sector}}$, $\Phi_{\text{partial}}$).

Not all geometric parameters are considered and interactions between loss mechanisms are not accounted for. To be more certain of each loss mechanism they would need to be isolated and investigated individually. Turbine cascades are often used to investigate blade rows and seek to examine more closely individual loss mechanisms. However they are limited in that they are often two dimensional and mounted in linear rows and do not include the effects of rotation (Baines 1997). As the scope of this thesis is the investigation of the effect of calculated turbine efficiency on cycle analysis results, a more detailed investigation of each individual loss mechanism using cascades is beyond the scope of this thesis. However all the test experimental data is presented in Appendix C so that it may be revisited in more detail in future work.
Because the model is based on a real rotating turbine, the losses attributed to interactions between loss mechanisms are accounted for in the overall summation of losses. One could argue that a simple polynomial curve could have been fitted to the data as a simplified loss model. But this work is intended to be built upon and expanded by future research students at UQ. By using a loss model that is comprised of individual losses it would be easier to scale and expand the model as the centre’s facilities expand and allow for testing of larger turbines. With the individual losses and loss coefficients they can be expanded to larger machines by making the loss coefficients a function of power or mass flow rate. This would give the model a wider range of applicability.
Chapter 11    Incorporated Cycle Analysis

A Rankine cycle analysis presents a chicken and egg type conundrum with regards to turbine efficiency. A turbine can’t be designed without knowing the conditions it will operate within and a cycle can’t be analysed without knowing the efficiency of the turbine. So typically turbine efficiency is assumed based upon average reported turbine efficiencies. But this can be misleading as turbine efficiency is dependent upon specific operating conditions of a cycle.

For a more meaningful and informative cycle analysis, real turbine efficiency should be calculated integral to the cycle analysis. To calculate the turbine efficiency a loss model must be employed to determine losses and thereby determine the actual gross work produced by a turbine and therefore the actual work produced by the cycle.

This chapter will discuss the incorporation of the axial impulse turbine design program (AXIAL), a loss model program and the cycle analysis program (ORCCA) that were detailed in Chapter 5 and Chapter 8 and Chapter 9. This means that all further discussions are based on turbines designed by the AXIAL program and losses calculated by SSAL within the cycle analysis calculations of ORCCA. AXIAL turbine designs are strictly limited to single stage axial impulse turbines. So all further discussion on turbines and turbines performance is referring to that of single stage axial impulse turbines designed by AXIAL. Different types of turbomachines (i.e. radial, multi-stage axial, high reaction turbines etc.) will have a different influence as part of an incorporated cycle analysis. This analysis is strictly limited to single stage axial impulse turbines.

11.1 Method of Implementation

To incorporate the loss model into the cycle analysis program (ORCCA) and the axial impulse turbine design program (AXIAL) it was written as a function that accepts inputs from ORCCA and AXIAL. The loss model program was short named SSAL (Single Stage Axial Losses). It is run inside a loop that iterates across a range of geometries, operating speeds and flow angles until it finds the turbine design with the maximum efficiency and satisfies the power requirements of the cycle being analysed. The process flow of SSAL is graphically shown in Figure 61 and the source code is in Appendix B.

For every cycle analysed, every feasible turbine (within a range of specified stator nozzle angles, rotor diameters, number of blades and number of nozzles) is analysed for losses. Based upon the calculated losses the turbine efficiency can be determined. The loss model function determines the
turbine geometry and operating speed that will yield the maximum gross work for the given flow conditions of the cycle.

11.1.1 Turbine Feasibility Checks

In the loss model function there are feasibility checks to ensure that the proposed turbine geometry is realistic. The program first performs the geometric feasibility checks that are listed below. If any one of the conditions is true then the turbine design is marked as unfeasible and discarded from the

Figure 61. Incorporated loss model program (SSAL) flow chart
list of potential solutions. For this thesis the limitations are based on the machining limitations of the Mechanical Engineering Workshop at QGECE. The feasibility checks are as follows;

Rotor pitch check  \[ p_r < t e_r \] (141)  
Rotor passage width check  \[ w_{r,tt} < w_{r,tt,\text{min}} \] (142)  
Admission check  \[ \varepsilon > 1 \] (143)  
Stator passage width check  \[ w_{s,tt} < w_{s,tt,\text{min}} \] (144)  
Stator passage height check  \[ h_{s,tt} < h_{s,tt,\text{min}} \] (145)  
Circumference check  \[ \lambda > C_r \] (146)  
Stator pitch check  \[ p_s < t e_s \] (147)  
Rotor height check  \[ h_{r,tt} < h_{r,tt,\text{min}} \] (148)  

The **Rotor pitch check** prevents erroneous solutions being created where the rotor pitch is less than the trailing edge thickness of the rotor blades. Such a scenario would result in a rotor with blades that were so close together that they would be touching.

The **Rotor passage width check** ensures that the width of the throat through the rotor is not less than the minimum specified width. The minimum specified width is an input to the program. For this thesis the minimum allowable width was based on the machining limitations of the Mechanical Engineering Workshop at QGECE. The smallest milling bit that could be run was 1mm. To allow for multiple passes to provide the best surface finish, a minimum allowable throat width of 2mm was specified.

The **Admission check** was performed to ensure that no erroneous geometries producing turbines with admission rates greater than 1 were accepted.

The **Stator passage width check** is similar to the Rotor passage width check. It was specified as 2mm for the same reasons.

The **Stator passage height check** is intended to give the ability to specify a minimum passage height. 2mm was also used for the limitation in this thesis.

The **Circumference check** compares the stator nozzle exit arc length multiplied by the number of nozzles (total nozzle arc length or active arc) versus the circumference of the stator. This is to ensure that the active arc is not greater than the stator circumference.

The **Stator pitch check** is the same as the Rotor pitch check but for the stator.
The **Rotor height check** is the same as the Stator passage height check but for the rotor.

If the turbine has acceptable geometry then several mechanical strength checks are performed. The first strength check is the shaft strength. The shaft strength is based on the shaft strength factor of safety as compared to the endurance limit of the shaft material. The calculations for shaft factor of safety \( F_{s\text{ shaft}} \), shaft stress \( \sigma_{\text{ shaft}} \), and shaft polar moment of inertia are as follows;

\[
F_{s\text{ shaft}} = \frac{\sigma_{\text{ Endurance}}}{\sigma_{\text{ shaft}}} > 2 \quad (149)
\]

\[
\sigma_{\text{ shaft}} = \frac{W_o D_{r,rt}}{4\omega I_p} \quad (150)
\]

\[
I_p = \frac{\pi D_{r,m}^4}{32} \quad (151)
\]

The maximum shaft diameter is assumed to be limited to half the rotor root diameter. The endurance limit of the shaft is specified based upon the specified shaft material. The material in this analysis for the shaft was specified as 316 stainless steel because stainless steel has good chemical resistance and strength properties.

Following the shaft strength check, the blade strength is checked. The average force on the blade is determined and then the portion of the blade with the greatest radius of curvature is modelled as a 2D simply supported beam. The blade strength check is calculated as follows;

\[
F_{s\text{ blade}} = \frac{\sigma_{\text{ yield}}}{\sigma_{\text{ blade}}} > 2 \quad (152)
\]

\[
\sigma_{\text{ blade}} = \frac{m\Delta V_u h_{r,tt} p}{6c\lambda (p - w_{r,tt})^2} \quad (153)
\]

Where the \( m\Delta V_u \) term is taken to be the force acting on the blade as the force is equal to the change in momentum (Shlyakhin 2005). The material in this analysis for the rotor was assumed to be an aluminium alloy, Al 2618 T-61, because of its machinability, light weight, corrosion resistance and because it has been proven as a good material for high pressure, high temperature applications. It has been used for tip speeds nearing 520 m/s and pressure ratios of about 4.5 (Baines 1997). Based on the reported maximum operating tip speed of 520 m/s for rotors made from Al 2618 T-61 a feasibility check was imposed for tip speed as follows;

\[
U > U_{\text{max allow}} \quad (154)
\]

Note that these strength checks are simplistic, but to allow ORCCA and SSAL to process a very large number of conditions, more complex structural analysis would be computationally expensive and greatly limit the number of cycles and geometries that could be analysed. These structural
checks of course would not be taken as absolute and any turbine predicted by the model should be assessed in more detail. But they are useful to screen and eliminate unrealistic geometries from the list of potential turbines.

For all turbines that pass the feasibility criteria, their designs and performance generated from SSAL are stored for comparison. The turbines are compared based upon their efficiency. The turbine with the highest efficiency is passed back to ORCCA and the cycle analysis continues using the highest calculated turbine efficiency.

11.1.2 Influence of Aspect Ratio

One of the main geometric variables in the rotor design is the aspect ratio. The aspect ratio (ratio of passage height to passage width) can have a range of values. The losses and efficiency will vary as the aspect ratio changes.

In order to reduce the number iterates in the loss model, the influence of aspect ratio on efficiency for a 2, 5 and 10kW pentane case were examined at 1,500kPa with $T_h$ and $T_c$ as 170 and 20°C respectively. Figure 62 shows the results of aspect ratio versus turbine efficiency, admission rate, number of blades and number of nozzles.

There was an optimum aspect ratio to yield the highest efficiency. For the 5 and 10kW case the optimum aspect ratio was 0.75 and for the smaller 2kW machine the optimum aspect ratio was 0.45. It can also be noticed in Figure 62 that the admission rate decreases with an increase in the aspect ratio. As the aspect ratio increases the passage height increases compared to the passage width. As the height increases and the width decreases the active arc decreases as well which reduces the rate of admission. For small machines like the ones modelled here it is suggesting that machine designs with short blading are desirable as it maximises the rate of admission with greater active arc and thereby increases the efficiency by minimising partial admission losses.
Figure 62. Influence of Aspect Ratio

Figure 63 shows the losses for two 10kW cases. One with an aspect ratio of 0.75 (left) and one with an aspect ratio of 4.00 (right).

Figure 63. Losses as a percent of the turbine work for two scenarios. Aspect ratio 0.75 (left) and aspect ratio 4.00 (right).

The machine with the higher efficiency is the one with the aspect ratio of 0.75. It can be seen that the losses for the 0.75 aspect ratio machine are less and that the primary difference between the two
cases is the partial admission losses (sector and pumping) for the 4.00 aspect ratio machine are much larger. As mentioned before, this is because the high aspect ratio leads to low active arc and therefore low admission rates which in turn lead to high partial admission losses.

11.1.3 Influence of Pitch/Chord Ratio (Zweifel coefficient)

The Zweifel coefficient (pitch to chord ratio) also has an influence on the design and efficiency. Again the coefficient was calculated across a range to examine its influence on efficiency so as to determine the best value to use in SSAL.

![Graphs showing influence of Zweifel coefficient on turbine efficiency](image)

Figure 64. Zweifel coefficient influence on turbine efficiency

Figure 64 shows the results of the Zweifel coefficient versus turbine efficiency. A typical Zweifel coefficient used is about 0.90 (Aungier 2006) but in this analysis the loss model showed that slight gains can be made by using 2.00. The analysis indicates that the efficiency gains can be realized by increasing the Zweifel coefficient.
There does appear to be marginal gains for increases in the Zweifel coefficient (Zw) beyond 2. As the Zw increases the pitch is increasing with respect to chord, this leads to a narrow rotor blade which will decrease the surface area of the rotor thereby reducing the windage losses. Figure 65 shows the losses for two 10kW cases.

![Figure 65. Losses as a percent of the turbine work for two scenarios. Zw of 0.75 (left) and Zw of 4.00 (right).](image)

One with a Zw of 0.75 (left) and one with a Zw of 4.00 (right). The difference is in the losses for the two cases is primarily in the increased windage and passage losses for the lower Zw case. The low pitch-chord ratio will produce a wide rotor which will have a higher surface area therefore greater drag losses or windage losses. It also will have longer rotor blade passage length which will increase the passage losses.

11.2 Incorporated Loss Model Cycle Analysis Example

To illustrate the incorporation of SSAL into ORCCA an example case for a single cycle will be analysed. The conditions for the example cycle are listed in Table 21.

<table>
<thead>
<tr>
<th>Cycle Scenario Input Variables</th>
<th>Value</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluid</td>
<td>R245fa</td>
<td>---</td>
</tr>
<tr>
<td>$T_h$</td>
<td>170</td>
<td>ºC</td>
</tr>
<tr>
<td>$T_c$</td>
<td>20</td>
<td>ºC</td>
</tr>
<tr>
<td>$P_2$</td>
<td>2.500</td>
<td>kPa</td>
</tr>
<tr>
<td>$EHX_{d_T}$</td>
<td>12</td>
<td>ºC</td>
</tr>
<tr>
<td>$CHX_{d_T}$</td>
<td>12</td>
<td>ºC</td>
</tr>
<tr>
<td>$ETAC$</td>
<td>75%</td>
<td>---</td>
</tr>
<tr>
<td>$W_o$</td>
<td>2, 5, 10 and 100</td>
<td>kW</td>
</tr>
</tbody>
</table>

The cycle analysis was run with an assumed turbine efficiency and then for a calculated turbine efficiency using AXIAL and SSAL incorporated into ORCCA. Figure 66 shows the T-S diagram of the cycle for the conditions listed in Table 21.
Figure 66. Optimum T-S diagram of the R245fa cycle based on Table 21

Running the loss model inside the cycle analysis calculates a range of feasible turbines where a feasible turbine passes the acceptance criteria as defined in section 11.1.1. For each feasible turbine the losses are calculated and the work output from the turbine is determined. The set of feasible turbines can be viewed according to work versus RPM as shown in Figure 67.

The colour gradient corresponds to the ETAT as shown in the colour bar. The circle size is qualitatively indicative of rotor mean diameter, $D_{r,m}$. In this plot it shows that there are a group of poor performing small turbines (represented by the small blue dots). These machines are dominated by partial admission losses and have low efficiencies. The higher performing machines are high speed with a moderate size.

Figure 67. Feasible turbines for 100kW R245fa cycle (work versus $\omega$). All results on the left and results for $\alpha_{II} = 80^\circ$ only on the right.
The plot on the left shows results for a range of $\alpha_{II}$ and $Z$, but the plot on the right isolates $\alpha_{II}$ to 80° to more easily view the relationship of speed and power. The right hand plot shows an increase in efficiency with increase in speed up until $\sim$50,000RPM then a decline. This coincided with the optimum $Z$ value, typically around 0.5 for full admission. It also shows that there are a few feasible larger slower turbines that have comparable performance to the smaller faster turbines at 80°.

The feasible turbines can also be viewed according to speed-work parameter, stator nozzle angle and work as shown in Figure 68. This plot shows that there is a range of geometry options for a feasible turbine to operate in the cycle. There are a range of diameters, flow angles and speed-work parameter values that satisfy the feasibility requirements.

![Figure 68. Feasible turbines for 100kW R245fa cycle, work versus $\alpha_{II}$ (left) and Z (right)](image)

For an impulse turbine, increasing the stator nozzle flow angle ($\alpha_{II}$) increases the work output. In general the greater the stator nozzle flow angle the greater the work output. There are both theoretical and practical limitations to how high $\alpha_{II}$ may be. Theoretically $\alpha_{II}$ could asymptotically approach 90° and achieve the highest possible efficiency. However, in practical terms this would require almost infinitesimally small gaps between stator and rotor for the fluid to transfer from nozzle to blade at large angles of attack that it would be impossible. Also, creating stator nozzles over 80° is difficult because it requires very thin walled sections at the nozzle outlet that lead to weaknesses in the design. From the experience of machining the test turbine in this thesis 80° was the limit that could comfortably be machined. Another limitation is operating speed. High angle machines will have a high operating speed. Operating speeds can be limited by material strength, bearings, coupling limitations, and generator limits.
The Z versus work plot shows a clear relationship between power and size and Z. There is an optimum Z to achieve the highest efficiency and work output. Also as Z increases, so does the size of the turbine. In this case the turbine that produces the most work is a turbine with a medium Z and a high $\alpha_{II}$. There is a balance between speed and size that will yield the optimum machine. The turbine design that produces the most work is highlighted by the dark black circle around the marker. This turbine represents the design that is the optimum balance between size and speed to achieve the highest efficiency.

For each feasible turbine the losses are calculated and stored. The losses for the feasible turbines can be examined to determine from where the inefficiencies originate. Figure 69 shows the individual losses for each of the feasible turbines versus shaft power (left) and the losses for the maximum efficiency case (right).

![Figure 69. Individual turbine losses for 100kW R245fa cycle of feasible turbines (left) and losses for maximum efficiency case (right)](image)

The plot on the left shows how the individual losses stack up to form the overall losses. At the lower power (low efficiency) the losses are dominated by incidence and trailing edge. Referring back to Figure 68 it can be seen that the lowest power turbines are the ones with low Z and $\alpha_{II}$. As the power increases with increasing $\alpha_{II}$ the incidence and trailing edge losses decrease. As the power output increases (i.e. efficiency increases) the losses are dominated by the partial admission and clearance losses. The loss distribution for the highest efficiency turbine is shown as the bar graph on the right in Figure 69.

And as the real efficiency increases, the losses as a percentage of the shaft power decrease as would be expected. The best performing turbine is the one that combines the optimum speed, size and geometry to minimise the overall losses. This means that all the individual losses have to be investigated in tandem for a range of conditions to determine the best design.
If the power requirement for the turbine is reduced then losses become even more pronounced and a more erratic behaviour appears. Feasible turbines for a cycle using the same conditions as listed in Table 21 (except for the power requirement is reduced to 10kW) are shown below. Figure 70 shows the feasible turbines according to speed-work parameter, stator nozzle angle and work for the 10kW case.

Notice that with the reduced power requirement, the working fluid mass flow rate is reduced. By reducing the working fluid mass flow rate the rate of admission is reduced (i.e. a partial admission turbine). With a reduced admission rate the efficiency decreases due to increases in pumping and sector losses.

It’s interesting to note that the optimum speed-work parameter calculated is 0.33. Generally turbines operate most efficiently at speed-work parameters around 0.50. But because in this example the mass flow rate is relatively low and therefore does not permit a full admission machine, partial admission losses tend to shift the optimum speed-work parameter to a lower value.
The lower optimum speed-work parameter coincides with what Roelke (1994) showed, that for lower admission rates there are lower optimum speed-work parameters. Figure 71 shows the individual losses for each of the feasible turbines for the 10kW cycle.

The partial admission losses (pumping and sector) are a major portion of the losses at the low admission rate. These losses are much less ordered than the losses for the 100kW cycle. At the low admission rates the losses are greater and the geometric constraints tend to filter out more geometries which leads to the erratic appearance of the losses.

The plot shows that the partial admission losses make up a much greater component of the overall losses for the lower power cycle.

For the maximum efficiency case for the 10kW case, compared to the losses in the 100kW case (Figure 72), the losses are larger in terms of percentage of work. The partial admission losses (sector and pumping) are a larger portion on a percentage basis. The low admission rates of the small turbine have large implications on performance because of the associated high partial admission losses.

As the power is reduced even further to a 2kW case the partial admission becomes even more prevalent and the losses appear even more erratic compared to the 100kW. The reason for the erratic losses in this analysis is because at the smaller powers the machines are smaller and the feasibility checks filter out more geometries.

Notice Figure 73 how the optimum Z is pushed even further down as the power decreases and the admission decreases.
In all of the cases (2, 10, 100 kW) described above, the optimum stator exit angle is always the maximum allowed. The most aggressive stator nozzle exit angle (for the case of single stage axial impulse turbines) will produce the most power. This is important as it allows the loss model to be constrained to a single stator nozzle exit angle value and thereby reducing the number of iterations required. However this has design implications, the high angle will produce machines with very high operating speeds. If there are limitations to operating speed then there will also be limits to the stator nozzle outlet angle and the speed-work parameter.

11.3 Cycle Sensitivities with Calculated Turbine Efficiency

In Chapter 5 the influence of various operating conditions on the performance of a cycle were discussed. When a loss model is incorporated into a cycle analysis the relationship between cycle performance and operating conditions varies from that of an assumed efficiency scenario. In this section the difference between an assumed turbine efficiency cycle and a calculated turbine efficiency analysis will be examined. Table 22 lists the range of conditions for which the assumed efficiency and the calculated efficiency cycle analysis were conducted.
11.3.1 Influence of Pressure ($P_2$)

The influence of evaporator pressure is very important to the performance of a cycle. To illustrate the impact of a loss model on the relationship between evaporator pressure and cycle performance (ranked according to $\beta$), cycles were analysed for the following conditions listed in Table 22.

Table 22. Conditions for the influence of calculated turbine efficiency on evaporator pressure example

<table>
<thead>
<tr>
<th>Cycle Scenario Input Variables</th>
<th>Value</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluid</td>
<td>R245fa</td>
<td>---</td>
</tr>
<tr>
<td>$T_h$</td>
<td>110-170</td>
<td>ºC</td>
</tr>
<tr>
<td>$T_c$</td>
<td>20</td>
<td>ºC</td>
</tr>
<tr>
<td>$P_2$</td>
<td>200 – 5,000</td>
<td>kPa</td>
</tr>
<tr>
<td>$EHX_{eff}$</td>
<td>12</td>
<td>ºC</td>
</tr>
<tr>
<td>$CHX_{eff}$</td>
<td>12</td>
<td>ºC</td>
</tr>
<tr>
<td>ETAC</td>
<td>75%</td>
<td>---</td>
</tr>
<tr>
<td>ETAT</td>
<td>55% – 80%</td>
<td>---</td>
</tr>
<tr>
<td>$W_o$</td>
<td>10</td>
<td>kW</td>
</tr>
</tbody>
</table>

Figure 75 shows $\beta$ versus evaporator pressure for the conditions listed in Table 22. The calculated efficiency case is top left and the calculated efficiency versus pressure is shown in the top right and examples of assumed efficiency are also shown for comparison ($ETAT = 55\%$ bottom left and $ETAT = 80\%$ bottom right). The curves in all plots show similar trends of $\beta$ versus evaporator pressures.

If the approximate ETAT is not known and an assumed efficiency is used in the calculated cycle conditions the optimum evaporator pressure could be selected incorrectly. Table 23 lists the optimum evaporator pressure for the cases of calculated efficiency and assumed efficiency of 55% and 80%.

Table 23. Comparison of optimum evaporator pressure for assumed efficiency and calculated efficiency

<table>
<thead>
<tr>
<th>ETAT</th>
<th>$T_h = 130^\circ C$</th>
<th>$T_h = 150^\circ C$</th>
<th>$T_h = 170^\circ C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculated</td>
<td>1,200</td>
<td>1,750</td>
<td>1,480</td>
</tr>
<tr>
<td>55%</td>
<td>1,230</td>
<td>1,625</td>
<td>1,950</td>
</tr>
<tr>
<td>80%</td>
<td>1,230</td>
<td>1,380</td>
<td>2,100</td>
</tr>
</tbody>
</table>

Notice the difference in optimum evaporator pressure for the case of $T_h$ equal to $170^\circ C$ for the calculated efficiency and assumed efficiency of 80%. The optimum pressure for the calculated efficiency case occurs when ETAT is calculated at 80%. But the optimum pressure is 1,480kPa as opposed to 2,100kPa if an assumed efficiency of 80% is used. That’s a difference of ~40%. Notice that at the assumed efficiency, the optimum pressure is 2,100kPa and the calculated efficiency at this pressure is ~76%.

This is important because evaporator pressure has a major influence on $\beta$. Based on this, it suggest that using ORCCA with incorporated AXIAL and SSAL could lead to a better estimation of the
optimum cycle conditions by calculating the efficiency and selecting the optimum evaporator pressure.

![Graphs showing the comparison of evaporator pressure influence of assumed efficiencies versus calculated efficiency for R245fa.](image)

Even when assuming an efficiency that is very close to the calculated efficiency there are differences in the optimum evaporator pressure.

### 11.3.2 Influence of Heat Exchanger Temperature Differential (EHX\(_{DT}\) and CHX\(_{DT}\))

The temperature differential between the working fluid and the thermal fluid at the evaporator inlet has an effect on cycle performance. To illustrate the impact of a loss model on the relationship between heat exchanger inlet temperature differential and \(\beta\), cycles were analysed for the conditions listed in Table 24.

<table>
<thead>
<tr>
<th>Cycle Scenario Input Variables</th>
<th>Value</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluid</td>
<td>R245fa</td>
<td>---</td>
</tr>
<tr>
<td>(T_h)</td>
<td>110-170</td>
<td>°C</td>
</tr>
<tr>
<td>(T_c)</td>
<td>20</td>
<td>°C</td>
</tr>
<tr>
<td>(P_2)</td>
<td>1,500</td>
<td>kPa</td>
</tr>
<tr>
<td>(EHX_{DT})</td>
<td>2-40</td>
<td>°C</td>
</tr>
<tr>
<td>(CHX_{DT})</td>
<td>12</td>
<td>°C</td>
</tr>
<tr>
<td>ETAC</td>
<td>75%</td>
<td>---</td>
</tr>
<tr>
<td>ETAT</td>
<td>55% – 80%</td>
<td>---</td>
</tr>
<tr>
<td>(W_o)</td>
<td>10</td>
<td>kW</td>
</tr>
</tbody>
</table>
Figure 76 shows $\beta$ versus $EHX_{dT}$. The calculated efficiency case is top left and the calculated efficiency versus pressures is shown in the top right and examples of assumed efficiency are also shown for comparison ($ETAT = 55\%$ bottom left and $ETAT = 80\%$ bottom right). As before, we can compare the optimum $EHX_{dT}$ selected from an assumed efficiency case to that of the calculated efficiency.

Table 25 lists the optimum $EHX_{dT}$ for the calculated cases and the assumed efficiency cases.

**Table 25. Comparison of optimum $EHX_{dT}$ for assumed efficiency and calculated efficiency**

<table>
<thead>
<tr>
<th>$ETAT$</th>
<th>$T_h = 130^\circ C$</th>
<th>$T_h = 150^\circ C$</th>
<th>$T_h = 170^\circ C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculated</td>
<td>19$^\circ$C</td>
<td>30$^\circ$C</td>
<td>23$^\circ$C</td>
</tr>
<tr>
<td>55$%$</td>
<td>6$^\circ$C</td>
<td>23$^\circ$C</td>
<td>18$^\circ$C</td>
</tr>
<tr>
<td>80$%$</td>
<td>20$^\circ$C</td>
<td>31$^\circ$C</td>
<td>20$^\circ$C</td>
</tr>
</tbody>
</table>

The difference is not as significant as that of the evaporator pressure, but there are still large enough differences to warrant using a calculated efficiency in the cycle analysis to select the optimum $EHX_{dT}$.

Similarly, $CHX_{dT}$ also has an influence on cycle performance. Table 26 lists the conditions for examining the influence of a loss model on the condenser inlet temperature differential.
Table 26. Conditions for the influence of calculated turbine efficiency on CHX<sub>dT</sub>

<table>
<thead>
<tr>
<th>Cycle Scenario Input Variables</th>
<th>Value</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluid</td>
<td>R245fa</td>
<td>---</td>
</tr>
<tr>
<td>( T_0 )</td>
<td>110-170</td>
<td>ºC</td>
</tr>
<tr>
<td>( T_c )</td>
<td>20</td>
<td>ºC</td>
</tr>
<tr>
<td>( P_2 )</td>
<td>1,500</td>
<td>kPa</td>
</tr>
<tr>
<td>( EHX_{dT} )</td>
<td>12</td>
<td>ºC</td>
</tr>
<tr>
<td>( CHX_{dT} )</td>
<td>2-40</td>
<td>ºC</td>
</tr>
<tr>
<td>( ETAC )</td>
<td>75%</td>
<td>---</td>
</tr>
<tr>
<td>( ETAT )</td>
<td>55% – 80%</td>
<td>---</td>
</tr>
<tr>
<td>( W_o )</td>
<td>10</td>
<td>kW</td>
</tr>
</tbody>
</table>

Figure 77 shows \( \beta \) versus \( CHX_{dT} \). Again, the trends are similar but there is enough difference to warrant using a calculated efficiency.

Figure 77. Comparison of \( CHX_{dT} \) influence of assumed efficiency versus calculated efficiency for R245fa

Table 27 summarises the optimum \( CHX_{dT} \) for the calculated efficiency and assumed efficiencies.

Table 27. Comparison of optimum \( CHX_{dT} \) for assumed efficiency and calculated efficiency

<table>
<thead>
<tr>
<th>( ETAT )</th>
<th>( T_h = 130^\circ C )</th>
<th>( T_h = 150^\circ C )</th>
<th>( T_h = 170^\circ C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculated</td>
<td>11ºC</td>
<td>20ºC</td>
<td>15ºC</td>
</tr>
<tr>
<td>55%</td>
<td>22ºC</td>
<td>21ºC</td>
<td>21ºC</td>
</tr>
<tr>
<td>80%</td>
<td>19ºC</td>
<td>21ºC</td>
<td>18ºC</td>
</tr>
</tbody>
</table>

Notice the difference in the range of efficiencies for the calculated \( CHX_{dT} \) case versus the range of efficiencies for \( EHX_{dT} \) case. Turbine efficiency is very sensitive to outlet temperature because it affects the turbine outlet pressure which in turn affects gas expansion and the fluid properties in the
rotor. All of the shear and drag forces such as windage and profile losses are dependent on fluid properties and therefore are dependent on $CHX_{dT}$. This is because $CHX_{dT}$ will determine the turbine outlet pressure which will also influence the turbine outlet temperature.

11.3.3 Influence of Power ($W_o$)

The influence of an incorporated loss model is prominent when sensitising across a range of powers. Table 28 lists the conditions for examining the influence of power.

Table 28. Conditions for the influence of calculated turbine efficiency on power example

<table>
<thead>
<tr>
<th>Cycle Scenario Input Variables</th>
<th>Value</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluid</td>
<td>R245fa</td>
<td>---</td>
</tr>
<tr>
<td>$T_h$</td>
<td>110-170</td>
<td>°C</td>
</tr>
<tr>
<td>$T_c$</td>
<td>20</td>
<td>°C</td>
</tr>
<tr>
<td>$P_2$</td>
<td>1,500</td>
<td>kPa</td>
</tr>
<tr>
<td>$EHX_{dT}$</td>
<td>12</td>
<td>°C</td>
</tr>
<tr>
<td>$CHX_{dT}$</td>
<td>12</td>
<td>°C</td>
</tr>
<tr>
<td>$ETAC$</td>
<td>75%</td>
<td>---</td>
</tr>
<tr>
<td>$W_o$</td>
<td>2-100</td>
<td>kW</td>
</tr>
</tbody>
</table>

Figure 78 shows $\beta$ versus power for R245fa. For assumed efficiency case there is no optimum power because the efficiency is the same for all cases. But in the calculated efficiency case there is an optimum power as efficiency is closely related to mass flow rate and mass flow rate is proportional to power.
Notice at powers below 20kW there is a large change in $\beta$ for a given power. ETAT can vary significantly at the lower powers. As discussed in section 11.2, this is because at the lower power the mass flow rate is low and that means the turbine will be small and have partial admission. The partial admission losses at the lower powers reduce the efficiency of the turbine and thereby reduce the efficiency of the cycle. After a certain point (and it will be slightly different for each fluid and pressure) full admission is realized and turbine efficiency becomes constant with power as the upper limit of efficiency is reached.

**Comparison of Geometry and Losses for 2-100kW Turbines**

The turbines modelled by AXIAL and SSAL are single stage axial impulse turbines and they show a relationship between power and mass flow rate and efficiency. As discussed earlier, partial admission losses become prominent in low power machines. Figure 79 shows the losses for turbines calculated by AXIAL and SSAL for two cases. A 2kW and a 100kW turbine for R245fa with inlet pressure of 2,500kPa. There is a significant reduction in efficiency between the 100kW and the 2kW case and the majority of the losses are coming from the pumping and the sector losses.

![Figure 79. Comparison of a 2kW (left) and a 100kW (right) turbine for R245fa at 2,500kPa](image)

This is representative of what’s occurring in ETAT shown in Figure 78 for powers below 10kW. The trends between power, admission, arc length and flow area are shown in Figure 80. The rate of admission is decreasing with power as the flow area decreases. With decreasing flow area, the stator nozzle arc decreases which in turn decreases the active arc. Notice that the rate of admission follows the same trend as the active arc because admission is the ratio of active arc to mean diameter.

If small turbines could be made with extremely small passages and wall thicknesses and still withstand the stresses and the forces required, the effects of partial admission could be eliminated.
But as mentioned previously, practical limitations were placed on feasible geometry and minimum passage sizes calculated in AXIAL. It’s not to say it can’t be done, but in the application of this thesis and the manufacturing capabilities of the facilities available, the limitations on feasible geometries were practical.

Recall from the section on the loss model (Chapter 9) that pumping losses have a cubic relationship to blade tip speed ($U$) and that sector losses are linearly proportional to $U$.

Figure 80. Relationship between power and admission for a 2500kPa R245fa turbine

Figure 81. Relationship between power and $U$ and $D_{r,m}$ for a 2500kPa R245fa turbine
Figure 81 shows the relationship between power and blade tip speed and rotor mean diameter ($D_{r,m}$). As the admission rate falls the pumping and sector losses become more prevalent and to minimise their impact, the blade speed is reduced. And for a given stator flow angle the nozzle exit velocity remains the same so to reduce $U$ the diameter of the rotor is reduced.

11.3.4 Influence of Working Fluid

The losses in a turbine are dependent upon fluid properties, particularly viscosity and density. In a cycle analysis with assumed turbine efficiency, a cycle may have a very high $\beta$ because the temperature profiles in the heat exchangers are closely matched with the thermal fluid and the coolant fluid but the pressure ratio, density, and viscosity in the turbine could be poorly suited for turbine performance. Table 29 lists the conditions for examining the influence of fluids on cycle performance when incorporating a turbine loss model.

<table>
<thead>
<tr>
<th>Cycle Scenario Input Variables</th>
<th>Value</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluids</td>
<td>Pentane, R245fa, RC318, R134a, R143a</td>
<td></td>
</tr>
<tr>
<td>$T_h$</td>
<td>110-170</td>
<td>ºC</td>
</tr>
<tr>
<td>$T_c$</td>
<td>20-40</td>
<td>ºC</td>
</tr>
<tr>
<td>$P_2$</td>
<td>200-8,000</td>
<td>kPa</td>
</tr>
<tr>
<td>$EHX_{in}$</td>
<td>12</td>
<td>ºC</td>
</tr>
<tr>
<td>$CHX_{in}$</td>
<td>12</td>
<td>ºC</td>
</tr>
<tr>
<td>ETAC</td>
<td>75%</td>
<td>---</td>
</tr>
<tr>
<td>ETAT</td>
<td>60%, 70% and 80%</td>
<td>---</td>
</tr>
<tr>
<td>$W_o$</td>
<td>10</td>
<td>kW</td>
</tr>
</tbody>
</table>

An analysis was conducted across the range of conditions listed above for assumed efficiencies of 60%, 70%, 80% and calculated efficiency. The results from the analysis are plotted as the optimum condition (condition that produces the highest $\beta$ for a given set of $T_c$ and $T_h$) from all possible conditions as listed in the above table. Figure 82 shows the operating conditions maps of optimum $\beta$ for the assumed turbine efficiency cases and for the calculated turbine efficiency case.

The trends of $\beta$ versus $T_c$ and $T_h$ are similar in that $\beta$ is at a maximum with the highest $T_h$ and lowest $T_c$ and at a minimum when $T_h$ is a minimum and $T_c$ is a maximum. However, ETAT varies across the range of temperatures enough to produce $\beta$ values that do not align to any single assumed efficiency.
Figure 82. Comparison of fluid influence on $\beta$ of assumed efficiencies and calculated efficiency. Calculated efficiency (top left), assumed 60% efficiency (top right), assumed 70% efficiency (bottom left), assumed efficiency 80% (bottom right).

Figure 83 shows the calculated ETAT associated with the optimum $\beta$ for each combination of $T_c$ and $T_h$ for the conditions listed in Table 29.

The efficiencies for the optimum $\beta$ range from 76% to 87% which will have an impact on the calculated $\beta$. Figure 84 shows ETAT for several $T_h$ versus $P_2$ for several fluids calculated for the conditions in Table 14. As seen before in 11.3.1 calculated efficiency is influenced by pressure and
it’s also affected by temperature. Some of the fluids are more sensitive than others as well. Take the case of R143a compared to R245fa. R143a ETAT decreases quite dramatically as pressure increases. And notice on RC318 there is a significant separation in ETAT between $T_h$ equal to 140°C and $T_h$ equal to 170°C as $P_2$ increases past 2,000kPa.

The influence of calculating ETAT on $P_2$ can be seen as well by looking at the operating conditions maps for the calculated ETAT case and the assumed efficiency cases for $P_2$ as shown in Figure 85.

The trend in evaporator pressure is different when considering the calculated turbine efficiency. And the principle reason for the difference in $\beta$ and $P_2$ is that the turbine efficiency is not constant across all the conditions as shown in Figure 83 and Figure 84.

The turbine efficiency falls as the thermal fluid temperature decreases and the coolant temperature increases. There is a large enough difference in the assumed and calculated efficiency for certain combinations of $T_c$ and $T_h$ to warrant investigating a cycle with an incorporated loss model.

This difference in ETAT influences the fluid selection as well. Figure 86 shows the optimum fluids plotted against $T_c$ and $T_h$ for both the assumed and the calculated efficiency cases. The colours correspond to the fluids labelled in the colour bar.
One of the most important aspects of designing an ORC is the selection of the working fluid. To thoroughly investigate the optimum working fluid for a cycle an incorporated turbine loss model should be implemented to accurately calculate turbine efficiency. Figure 86 shows that the incorporation of a loss model into the cycle analysis does influence fluid selection. For example if a cycle is to be designed for $T_h$ equal to 120°C and $T_c$ equal to 20°C the assumed 80% efficiency case would predict R143a but the calculated efficiency case would predict RC318.
11.4 Discussion

The incorporated loss model (SSAL) into the cycle analysis program (ORCCA) does prove to have an influence on the results. But note that this model and analysis is very specific to the type of turbines investigated. Single stage axial impulse turbines with supersonic stator nozzles were examined. AXIAL and SSAL incorporated into ORCCA are solely based on this type of machine. The characteristics of this type of machine are that it requires large turning angles of the fluid in the rotor blade row which leads to aggressive inlet and outlet angles. Because the flow angles are aggressive, the tip speed is large to minimise exit swirl losses. Being a single stage machine, the turbine needs to extract all the energy in a single blade row and to do so requires the blade tip speed to be about half the nozzle exit velocity. The nozzle exit velocity from the supersonic stator is large (Mach numbers around 1.4 for the R245fa case) so the blade speed is large which translates to high RPMs. So the types of turbine geometries produced by AXIAL are small, high speed, aggressive flow angle turbines.

The tested turbine was for a very small turbine (less than 1kW) of this kind and the loss model was calibrated to that turbine. Some caution should be used when extrapolating the loss model to larger power ratings. However, the coefficients used to fit the loss model to the experimental only deviated
slightly from the default values so there is a degree of confidence that the loss model would work for larger powers.

The aspect ratio and Zweifel coefficients were investigated to determine which one to use in the AXIAL program incorporated into ORCCA. Aspect ratio and Zw could be iterates as well but the number of cycle conditions was already large enough that increasing the number of iterations was time prohibitive. So an initial look into the influence of aspect and Zw was conducted to determine what values to use in the incorporated loss model. From the analysis the aspect ratio showed to have an optimum with respect to turbine efficiency at around 0.75. There was shown to be a correlation between aspect ratio and admission, as the aspect ratio increased the admission decreased. As the aspect ratio was increased the passage height was increased thus reducing the passage width and thereby the active arc. This leads to a reduction in the rate of admission. This suggests that for small machines short blades are desirable as it increases admission with greater active arc and thereby reducing partial admission losses.

Zw was shown to have an influence on the efficiency as well. With increasing values of Zw an asymptotic upper limit of efficiency was approached. But there were diminishing gains beyond a Zw value of 2. Lower values of Zw will produce a wide rotor which will have higher surface area therefore greater windage losses. It also will have longer rotor blade passage length which will increase the passage losses. A Zw of 2 was used in the analysis even though the model showed that Zw values greater than 2 could produce more efficient designs, it was thought that these designs may not be realistic. With high Zw values the blading would be narrow and have a very small turning radius in the blades which could in turn lead to flow separation in the blade and a set of losses that wouldn’t be captured by the loss model. This raises a point that should be made about the model. It is a mean line initial analysis tool. The results from these tools should be further scrutinized using CFD for specific stator and rotor geometries to ensure that a turbine will perform as intended.

The turbine designs from AXIAL have large stator outlet angles with large blade inlet and outlet angles as well. And because the turbines are supersonic, the exit velocity of the stator nozzle is very high which requires a high blade tip speed. To achieve a large blade tip speed either the rotational speed must be high or the diameter must be large. High speeds and large diameters have large associated drag losses. So a balance of size and speed is required. It was seen in the results of Z versus power versus ETAT that the optimum solutions were ones with moderate to low values of Z, less than 0.5, and a moderate size (relative to the range diameters for the set of feasible turbines calculated). The stator outlet angle was found to be the maximum allowed for all cases. In this analysis the stator outlet was
limited to 80° because that was found to be a practical limit for machining when the first test turbine was machined. For this single stage case, the maximum stator outlet angle provides the greatest potential for turning the fluid. The energy extracted is proportional to the difference in the tangential velocities so for the impulse case a maximum inlet tangential velocity is required and a zero exit tangential velocity is the goal.

The incorporated loss model has shown to have a significant effect on the cycle analysis. The selection of optimum evaporator pressure and heat exchanger inlet temperature differentials varies significantly for assumed turbine efficiency when compared to calculated turbine efficiency. The efficiency of the turbine is related to pressure so that the efficiency varies as the pressure varies. The case examined in this section shows that even when the assumed efficiency (80%) is close to the average calculated (76%), for a given pressure range, the selection of the optimum pressure was a difference of 40%. The same was shown for $EHX_{IT}$ and $CHX_{IT}$, the selection of the optimum was significantly different for the incorporated cycle versus the cycle with an assumed efficiency.

The influence of power on turbine and system performance was examined. The effects of partial admission losses have been a focal point of this examination. Particularly when dealing with small powers. In this section it was shown that admission rates decline with power and it declines quickly below a power of 20kW. As the power declines, the mass flow rate declines and the total stator flow area declines. As the stator area declines, the total active arc decreases and thus the admission rate declines. As the admission rate decreases, the influence of partial admission loss becomes more prominent and the response to increased partial admission losses is the optimum speed work parameter decreasing. This is because the pumping and sector losses are strongly dependent on $U$. Decreasing $U$ decreases partial admission losses but it also increases the swirl losses so there is a balance point between the reduction of partial admission losses by reducing $U$ and the increased swirl losses. This is why the optimum speed work parameter begins to decline from the full admission optimum of 0.5 when admission rates fall.
Chapter 12  Loss Model Based Turbine Design

Earlier in this thesis (Chapter 7) a single stage impulse turbine was built in order to test and validate a computational loss model program. The turbine was initially built based on a velocity triangle analysis but did not account for losses. This chapter will show the redesign of the turbine using the loss model incorporated into the cycle analysis program. The turbine will be designed to achieve the maximum $\beta$ using R245fa within the limits of the small ORC test rig.

12.1 Cycle Conditions

The design of the new turbine began with a cycle analysis constrained by the operating limits of the ORC test rig as listed in Table 30 below. For comparison, the previous design parameters and ORC test rig limits are listed.

<table>
<thead>
<tr>
<th>Table 30. ORC Test Rig Operating Limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redesign Parameter</td>
</tr>
<tr>
<td>Working Fluids</td>
</tr>
<tr>
<td>Maximum Working Fluid Mass Flow Rate</td>
</tr>
<tr>
<td>Maximum Evaporator Pressure</td>
</tr>
<tr>
<td>Thermal Fluid Maximum Inlet Temperature</td>
</tr>
<tr>
<td>Maximum Heat Addition</td>
</tr>
<tr>
<td>Cooling Water Inlet Temperature</td>
</tr>
<tr>
<td>Maximum Expander RPM</td>
</tr>
<tr>
<td>Maximum Expander Torque</td>
</tr>
<tr>
<td>Maximum Turbine Outlet Pressure</td>
</tr>
<tr>
<td>Recuperation Employed</td>
</tr>
</tbody>
</table>

The operating limits were imposed by the equipment limitations in the ORC test rig. At the onset of this thesis the dynamometer operating limit was 10,000RPM, however the dynamometer has been improved and can now operate at speeds up to 27,000RPM. R245fa was selected as the working fluid based on its cost, availability and relatively low hazard level. Also because it has a lower condensation pressure for the cooling source of 25°C as compared to that of R134a. This helps in two ways, it reduces the density in the rotor housing which reduces drag losses and it also increases the factor of safety for the magnetic coupling which is rated to 1,000kPa. Also, after the original turbine design, a recuperator was added which allowed for consideration of cycles employing recuperation. And a further improvement was that the heater was improved so that the thermal source could operate up to 180°C.

12.2 Cycle Analysis Results

The cycle analysis program was run using the incorporated loss model to search for a cycle that maximizes $\beta$ for the ORC test rig. Figure 87 shows $\beta$ versus evaporator pressure for the conditions
in Table 30 and the cycle that yields the best performance is the highest temperature cycle with an evaporator pressure of 760kPa.

![Figure 87. β versus P (left) and T-S diagram of maximum β cycle (right) for ORC test rig based on cycle analysis with incorporated loss model](image)

It shows that a low pressure recuperated cycle yields a high β cycle. A high value of β is desirable when the main objective is to make the most efficient use of the thermal fluid. Because this analysis is focused on making the most efficient use of the thermal fluid rather than the most efficient turbine, the turbine with the highest efficiency was not selected. Bear in mind, this analysis is based solely on the single stage axial impulse turbines designed by AXIAL. Figure 88 shows the efficiencies versus β and the efficiency versus pressure. Notice that the highest efficiency turbines are the low pressure conditions for all source temperatures. Less than 500kPa turbine inlet pressure will produce the highest turbine efficiency for the scenarios investigated from Table 30. But as seen in the β versus pressure plot overall cycle performance is not the optimum.

![Figure 88. β versus ETAT (left) and ETAT versus Pressure (right) for ORC test rig based on cycle analysis with incorporated loss model](image)
The relationship between efficiency and pressure is principally based on the influence that pressure has on the stator nozzle exit velocity. As the pressure differential across the stator increases so too does the exit velocity. Figure 89 shows the relationships between turbine characteristics and pressure for the ORC test rig turbine investigated. The larger exit velocity is achieved by a greater expansion of the gas. In a supersonic turbine the expansion occurs in the divergent portion of the nozzle. Greater expansion requires a longer divergent portion and that translates to a greater nozzle arc length. As the arc length increases for the stator, the number of permissible nozzles begins to decrease for a given mean diameter. And with a decrease in the number of nozzles in stator the rate of admission decreases. As the rate of admission decreases $Z$ begins to decrease. This is because the partial admission losses are strongly dependent on $U$. As admission decreases partial admission losses increase and to compensate $Z$ decreases to reduce $U$ and thereby reduce the impact of partial admission losses. This is why the low pressure turbine for this case has the higher efficiency because it is able to achieve a higher admission rate and a higher $Z$.

But as noted, the aim here is to achieve the highest $\beta$ with a single stage axial impulse turbine using R245fa for the ORC test rig.
ORCCA, with AXIAL and SSAL, incorporated show that the operating condition of $P_2$ equal to 760kPa and source temperature equal to 180°C is the optimum cycle. Figure 90 shows a plot of the feasible turbines for the ORC test rig for the optimum cycle.

![Feasible Turbines Power vs $a_{in}$ marker size indicative of $D_{in}$](image1)

![Feasible Turbines Power vs $Z$ marker size indicative of $D_{out}$](image2)

**Figure 90. Feasible turbines for optimum R245fa cycle in ORC test rig. $a_{in}$ versus power (left) and $Z$ versus power (right)**

The small available mass flow rate of the ORC test rig limits the rate of admission of the turbine. The low admission rate pushes the speed-work parameter down as well as the efficiency. In this case, the turbine that produces the most work for this cycle has an admission rate of 48% with an efficiency of 66%. Note that the incorporated analysis (ORCCA, AXIAL and SSAL) predicts a higher efficiency turbine could be produced for the cycle conditions but it would require a higher rotational speed. The analysis here was limited to feasible machines that had a rotational speed of 27,000RPM or less (working limit of the dynamometer).

In this low flow rate scenario, the lower speed-work parameter cases produce the majority of the top performing turbines. The maximum work turbines are clustered around the lower $Z$ and higher operating speeds. In this low mass flow rate condition, the losses are concentrated in the partial admission, trailing edge, clearance and windage losses. Figure 91 shows the losses for the maximum work case shown in Figure 90.

The low admission rate causes the partial admission losses (pumping and sector) to comprise a majority of the total turbine losses. There are significant losses due to large exit swirl velocities in this case. The exit swirl originates from the low $Z$. For an impulse turbine the optimum speed-work parameter is 0.5 to produce no exit swirl but in this small, speed-limited scenario exit swirl is generated. The exit swirl is fluid leaving the rotor with usable energy that is not being converted into mechanical energy and is therefore a loss.
In this figure (Figure 91) the total losses are the sum of the losses calculated by the loss model and the swirl is the amount of kinetic energy leaving the rotor. It is unused energy in the fluid that is lost.

**Turbine Selection Process**

The process for selecting the turbine geometry for the redesigned turbine for the ORC test rig is summarised by the following steps;

1. Conduct a cycle analysis in ORCAA with AXIAL and SSAL modules incorporated calculating turbine geometry, losses and efficiency.
2. Review the $\beta$ versus evaporator pressure plots and select the cycle that produces the highest $\beta$ value. Figure 87 shows that a cycle with a source temperature of 180°C and pressure of 760kPa and employing recuperation produces the highest $\beta$ for the operating conditions available in the ORC test rig as listed in Table 30. Keeping in my mind the aim of this thesis is maximising the system performance not just turbine performance. A lower efficiency turbine may be selected to achieve a higher cycle performance.
3. Review $\alpha_{II}$ and $Z$ plots for maximum $\beta$ condition. The maximum work condition for the feasible turbines calculated for the optimum cycle is a turbine with $\alpha_{II}$ of 80° and $Z$ equal to 0.3.
4. Retrieve geometry for the turbine that produces the highest $\beta$ condition from the output of AXIAL.

**12.3 Paths and Surface Geometry**

The cycle analysis produces the basic required geometry to design a single stage impulse turbine. The basic geometry for the best performing turbine was used as inputs to the design module in the
AXIAL program. The AXIAL program creates the 2D and 3D paths that frame the stator and rotor passage volumes shown in Figure 92.

The turbine rotor and stator passage geometry designed by SSAL (Single Stage Axial Losses) incorporated into ORRCA was imported into a Solid Edge™ CAD model. Based upon the imported stator and rotor passages the turbine was designed as a single stage axial turbine. The same shaft, ceramic bearing and magnetic coupling concepts were incorporated into the new design because of their good performance and reliability in the original design.

![Figure 92. Path and surface geometry for optimum R245fa turbine for ORC test rig. 2D paths of stator and rotor passage (upper left), 3D paths of stator and rotor passage (upper right), 3D surfaces of stator and rotor passage (lower left), divergent nozzle geometry (lower right)](image)

12.4 Comments on Redesigned Turbine

This initial turbine built for the test rig was built as a turbomachine test platform for an ORC test rig. The goal of the initial turbine design was to build a machine that could be tested for performance in order to validate a loss model to be incorporated into the cycle analysis. The original turbine was designed based on a R134a cycle that operated within the constraints of the ORC test rig. The turbine was also designed based on the velocity triangle results with no loss model. The implication of which is the turbine would operate at off-peak conditions because of losses not taken into account in the design.

Figure 93 shows a comparison of the original cycle and the redesigned cycle. Both cycles are superheated cycles but the new design employs recuperation which allowed for higher mass flow.
rates but remain within the limitations of 20kW available heat. The design is intended to operate at an efficiency of 66% as opposed to the tested efficiency of the original of 35% for the original turbine. The new turbine is significantly smaller, 70% smaller mean diameter.

This is possible because of the upgraded dynamometer which allows for rotation speeds up to 27,000RPM (original was constrained to a maximum of 10,000RPM). The new design has more nozzles because of the higher mass flow rate and as such it has a higher rate of admission. The rate of admission is much higher because of the greater number of nozzles and overall active arc length but also because the new machine is much smaller due to its allowed higher rotational speed. The previous design had a very small active arc length and a large mean diameter which meant for a very low rate of admission, ~5%. This very low rate of admission was the primary source of losses. With the new design the rate of admission is 48% which will substantially decrease partial admission losses. The new design also has a supersonic velocity from to stator at 1.40 which is similar to the original design which was 1.45. But both have subsonic Mach numbers relative to the blade inlets when operating at the design point. The original machine however wasn’t able to operate at its design point because of excessive losses so when it was in operation the Mach numbers entering the blade were supersonic, about 1.2. A tabulated comparison of the new design and the original design is presented in Table 31 for reference.
Table 31. Dimensions of original turbine and the redesigned turbine for the small ORC test rig

<table>
<thead>
<tr>
<th>Turbine Dimension</th>
<th>Original Turbine</th>
<th>Redesigned Turbine</th>
<th>UoM</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluid</td>
<td>R134a</td>
<td>R245fa</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rotor Mean Diameter</td>
<td>$D_{r,m}$</td>
<td>0.1977</td>
<td>0.0600</td>
<td>m</td>
</tr>
<tr>
<td>Number of Nozzles</td>
<td>$N_{nozzles}$</td>
<td>2</td>
<td>5</td>
<td>150%</td>
</tr>
<tr>
<td>Stator Nozzle Throat Height (Root to Tip)</td>
<td>$h_{s,rt}$</td>
<td>0.0025</td>
<td>0.0023</td>
<td>m</td>
</tr>
<tr>
<td>Stator Nozzle Throat Width</td>
<td>$w_{s,rt}$</td>
<td>0.0023</td>
<td>0.0031</td>
<td>m</td>
</tr>
<tr>
<td>Stator Tip Diameter</td>
<td>$D_{s,tip}$</td>
<td>0.2000</td>
<td>0.0623</td>
<td>m</td>
</tr>
<tr>
<td>Stator Root Diameter</td>
<td>$D_{s,rt}$</td>
<td>0.1950</td>
<td>0.0577</td>
<td>m</td>
</tr>
<tr>
<td>Rotor Pitch</td>
<td>$p$</td>
<td>0.0134</td>
<td>0.0118</td>
<td>m</td>
</tr>
<tr>
<td>Rotor Tip Diameter</td>
<td>$D_{r,tip}$</td>
<td>0.2000</td>
<td>0.0631</td>
<td>m</td>
</tr>
<tr>
<td>Rotor Root Diameter</td>
<td>$D_{r,rt}$</td>
<td>0.1948</td>
<td>0.0569</td>
<td>m</td>
</tr>
<tr>
<td>Rotor Blade Height (Root to Tip)</td>
<td>$h_{r,rt}$</td>
<td>0.0026</td>
<td>0.0031</td>
<td>m</td>
</tr>
<tr>
<td>Rotor Blade Throat Width</td>
<td>$w_{r,tt}$</td>
<td>0.0041</td>
<td>0.0029</td>
<td>m</td>
</tr>
<tr>
<td>Stator Outlet Absolute Flow Angle</td>
<td>$\alpha'_{II}$</td>
<td>80</td>
<td>80</td>
<td>deg</td>
</tr>
<tr>
<td>Rotor Inlet Metal Angle</td>
<td>$\beta'_{II}$</td>
<td>71</td>
<td>74.2</td>
<td>deg</td>
</tr>
<tr>
<td>Rotor Outlet Metal Angle</td>
<td>$\beta''_{II}$</td>
<td>71</td>
<td>74.2</td>
<td>deg</td>
</tr>
<tr>
<td>Number of Blades</td>
<td>$N_{blades}$</td>
<td>47</td>
<td>16</td>
<td>66%</td>
</tr>
<tr>
<td>Rotor Blade Suction Surface Arc Radius</td>
<td>$R_{s,ss}$</td>
<td>0.0200</td>
<td>0.004</td>
<td>m</td>
</tr>
<tr>
<td>Stator Rotor Gap</td>
<td>$g$</td>
<td>0.0015</td>
<td>0.0015</td>
<td>m</td>
</tr>
<tr>
<td>Blade Tip Clearance</td>
<td>$j$</td>
<td>5.00E-05</td>
<td>3.00E-5</td>
<td>m</td>
</tr>
<tr>
<td>Blade Meridional Passage Length</td>
<td>$l$</td>
<td>0.0791</td>
<td>0.0153</td>
<td>m</td>
</tr>
<tr>
<td>Blade Chord</td>
<td>$c$</td>
<td>0.0499</td>
<td>0.0062</td>
<td>m</td>
</tr>
<tr>
<td>Blade Tip Thickness</td>
<td>$t_{et}$</td>
<td>0.0010</td>
<td>0.001</td>
<td>m</td>
</tr>
<tr>
<td>Admission</td>
<td></td>
<td>0.05</td>
<td>0.48</td>
<td>860%</td>
</tr>
<tr>
<td>Blade Tip Speed</td>
<td>$U$</td>
<td>130</td>
<td>84</td>
<td>m/s</td>
</tr>
<tr>
<td>Stator Nozzle Exit Absolute Velocity</td>
<td>$V_{II}$</td>
<td>264</td>
<td>228</td>
<td>m/s</td>
</tr>
<tr>
<td>Stator Nozzle Exit Absolute Mach Number</td>
<td></td>
<td>1.45</td>
<td>1.40</td>
<td>3%</td>
</tr>
<tr>
<td>Blade Inlet Relative Velocity</td>
<td>$Y_{II}$</td>
<td>137</td>
<td>146</td>
<td>m/s</td>
</tr>
<tr>
<td>Blade Inlet Relative Mach Number</td>
<td></td>
<td>0.80</td>
<td>0.91</td>
<td>14%</td>
</tr>
</tbody>
</table>
Chapter 13  Conclusion

The intent of this thesis was to investigate the implications of incorporating a single stage axial impulse turbine loss model into a Rankine cycle analysis and to follow the process through to implementation of a turbine in a test rig. Single stage axial impulse turbines were selected as the turbine to investigate because of their promise to accommodate a wide range of pressure ratios, working fluids, and to be relatively small, simple and inexpensive.

**Thermal Conditions and Parameter of Merit for an Effective Cycle**

In all cases investigated the greatest differential between geothermal source temperature \( T_{h,i} \) and coolant source temperature \( T_{c,i} \) provided the most efficient cycle. The most efficient cycle will have the greatest differential between \( T_{h,i} \) and \( T_{c,i} \). But these are limited by the natural conditions of the resource and the environment. For a geothermal power station the only way to improve upon the natural temperature limitations of the environment is to boost the heat source or cool the cooling source (i.e. solar boosting and/or cooling). It was found that when designing a geothermal binary power station that \( \beta \) is a better metric than \( \eta_{1st} \). The first law efficiency, \( \eta_{1st} \), does not correspond to the most efficient use of a geothermal brine flow. It was shown that when \( \beta \) is plotted versus \( \eta_{1st} \) there is an optimum \( \beta \) that is not the maximum \( \eta_{1st} \). It was shown that if a cycle were selected based on maximum \( \eta_{1st} \) as opposed to \( \beta \) a 70% reduction in \( \beta \) was realized. This shows the significance in using \( \beta \) as a metric for geothermal binary power station design when the goal is to maximize the energy extracted per unit mass flow of geothermal brine.

**Cycle Conditions for an Effective Cycle**

But within a given resource and environments temperature range the cycle conditions can be manipulated to produce the most amount of power for a given mass flow rate of the geothermal source. The evaporator pressure is a critical operating condition for the cycle. It influences the effectiveness of heat transfer between the working fluid and the geothermal source and it influences the performance of the turbine. It’s been shown by other authors that there is an optimum evaporator pressure for cycle performance and the findings of this analysis agree. But it was also shown that the optimum pressure for cycle performance does not necessarily coincide with the optimum pressure for turbine performance. In the redesigned turbine it was shown that the best turbine efficiency was achieved at a pressure ratio less than 2.2. But the cycle performance was best at a higher pressure ratio of 3.3. The higher pressure in the evaporator increased the heat recovery from the geothermal source enough to offset the decrease in turbine efficiency.
The cycle and turbine designer’s goal would be to align the pressure at which the turbine operates at a maximum efficiency and the cycle as well operates at a maximum performance. However this can’t always be achieved given constraints on a cycle. The results presented here do show that it’s valuable to consider both turbine performance and cycle performance in parallel to achieve the objective, which is to make the most effective use of the geothermal resource.

This analysis also showed that cycle conditions are interrelated. By changing one parameter, others are influenced. The relationship between pressure and heat exchanger inlet temperature differential ($EHX_{dT}$ and $CHX_{dT}$) was shown to be highly dependent on one another. When a cycle was analysed for a range of $EHX_{dT}$ and $CHX_{dT}$, but for a constant pressures, there was a diminishing effect on $\beta$ with an increase in $EHX_{dT}$ and a distinct maximum $\beta$ for $CHX_{dT}$. But when the analysis is run over range of pressure and $EHX_{dT}$ and $CHX_{dT}$ there is an almost straight line relationship (zero slope) between $\beta$ and $EHX_{dT}$ and $CHX_{dT}$ approaches an asymptotic upper limit. It was seen that as the pressure moved into the supercritical regime the $EHX_{dT}$ dropped quickly as the pinch point was diminishing due to the reduction in latent heat of the fluid. And for the cases investigated here, fan forced air cooling, $CHX_{dT}$ would increase quickly as the evaporator pressure moved into the supercritical regime and $EHX_{dT}$ decreased. As $EHX_{dT}$ decreased there was more available enthalpy across the turbine because the turbine inlet temperature was higher. However there was an overall net benefit to sacrifice some of that available enthalpy across the turbine by increasing $CHX_{dT}$ which reduced the coolant mass flow rate and thus the cooling work. The increase in $CHX_{dT}$ does increase the turbine outlet pressure and thus reduces the available enthalpy that could be converted into work in the turbine, but from an overall system perspective, the increased $CHX_{dT}$ maximizes the net work produced by unit mass of geothermal fluid produced by reducing the cooling work.

The other condition that was investigated was the impact of employing recuperation. Recuperation can be beneficial in that it reduces the cooling work of a cycle. But it can also be beneficial in how it affects the evaporator. It was shown that for low pressure cycles (low pressure being relative and meaning less than the critical pressure) that $\beta$ could be improved with recuperation by having the fluid that exits the recuperator and enters the evaporator to have already initiated evaporation. By initiating evaporation in the recuperator the pinch point in the evaporator is removed. The absence of the pinch point in the evaporator leads to increased utilisation of the geothermal fluids energy and a higher overall $\beta$. 
Importance of an Incorporated Loss Model into a Cycle Analysis

It was shown that the incorporation of the loss model into the cycle analysis will significantly influence the selection of operating conditions for the cycle. Depending on the resource conditions the selection of evaporator pressure was shown to vary by as much as 40% between an incorporated cycle analysis and one that assumes turbine efficiency. The same was shown for $E_{HT}$ and $CH_{VT}$, the selection of the optimum was significantly different for the incorporated cycle versus the cycle with an assumed efficiency. The incorporation was particularly important to low power cycles (less than 20kW). It was shown that at powers less than 20kW there begins to be a decline in turbine performance. At the small power ratings the mass flow rate decreases which reduces the stator passage flow area. This in turn reduces the stator exit arc which then reduces the admission rate.

Also it was seen that pumping losses have a cubic relationship to blade tip speed ($U$) and that sector losses are linearly proportional to $U$. As the admission rate falls the pumping and sector losses become more prevalent and to minimize their impact, the blade speed is reduced. And for a given stator flow angle the nozzle exit velocity remains the same so to reduce $U$ the diameter of the rotor is reduced. Also to minimize the influence of partial admission losses the speed-work parameter falls with a decline in power, this is also part of reducing the impact of a strong relationship to $U$ for partial admission losses.

Partial Admission Cycle Characteristics

Partial admission designs for single stage axial impulse turbines are predominantly influenced by power. A very strong correlation between power, mass flow rate and admission exists. What was shown in this analysis was that admission declines with power and that it rapidly decreased at powers less than 20kW. But another condition that can influence admission is the pressure ratio. As the pressure differential across the stator increases so too does the exit velocity. The larger exit velocity is achieved by a greater expansion of the gas. In a supersonic turbine the expansion occurs in the divergent portion of the nozzle. Greater expansion requires a longer divergent portion and that translates to a greater nozzle arc length. As the arc length increases for the stator, the number of permissible nozzles begins to decrease for a given mean diameter. And with a decrease in the number of nozzles in stator the rate of admission decreases. As the rate of admission decreases $Z$ begins to decrease. This is because the partial admission losses are strongly dependent on $U$. As admission decreases partial admission losses increase and to compensate, $Z$ decreases to reduce $U$ and thereby reduce the impact of partial admission losses.
Influence of Zweifel Coefficient and Aspect Ratio on Single Stage Axial Impulse Turbine

A typical Zweifel coefficient ($Zw$) used is about 0.90 (Aungier 2006) but in this analysis the loss model showed that slight gains can be made by using 2.00. The analysis indicates that efficiency gains can be realized by increasing the Zweifel coefficient.

There does appear to be marginal gains for increases in the $Zw$ greater than 2. As the $Zw$ increases the pitch increases with respect to chord, this leads to a narrow rotor blade which will decrease the surface area of the rotor thereby reducing the windage losses. The low pitch-chord ratio will produce a wide rotor which will have a higher surface area therefore greater drag losses or windage losses. It also will have longer rotor blade passage length which will increase the passage losses. There are limits to $Zw$ however. If $Zw$ becomes too large then the blade is going to be very narrow, and the turning radius too small, which may cause flow separation on the suction side of the blade.

There was an optimum aspect ratio to yield the highest efficiency. For the 5 and 10kW case the optimum aspect ratio was 0.75 and for the smaller 2kW machine the optimum aspect ratio was 0.45. It was also noticed that the admission rate decreases with an increase in the aspect ratio. As the aspect ratio increases the passage height increases compared to the passage width. As the height increases and the width decreases the active arc decreases which reduces the rate of admission. For small machines like the ones modelled here it is suggesting that machine designs with short blading are desirable as it maximises the rate of admission with greater active arc and thereby increases the efficiency by minimising partial admission losses.

Impact of Multiple Sectors on Sector and Pumping Losses

The original loss model employed in this thesis did not take into account the findings of Varma et. al. regarding the influence of multiple sectors on partial admission losses. Without the addition of the multiplicative term $N_{sector}$ to sector loss as Varma suggested the loss model needed machine specific loss coefficients that deviated significantly from the default values.

While trying to fit the loss model to the experimental data, Varma’s suggestion of the addition of $N_{sector}$ was implemented to the sector loss model. This improved the results of fitting the loss model to the experimental data. But even with this addition the loss model required significant machine specific loss coefficients to fit the experimental data.

It was found that the addition of the $N_{sector}$ term to the pumping losses, in addition to the sector losses, improved the models fit. The addition of $N_{sector}$ to pumping losses improved the fit as well as reduced the loss models reliance on machine specific coefficients. The machine specific loss
coefficients found from the search algorithm were much closer to the default values and produced a better fit to the experimental data.

With the addition of the $N_{\text{sector}}$ term to the loss model, the reliance of the loss model on the coefficients is significantly reduced. The deviation from the default values was 0% for 6 out of 10 coefficients ($\Phi_{\text{clearance}}$, $\Phi_{\text{trail}}$, $\Phi_{\text{passage}}$, $\Phi_{\text{super}}$, $\gamma_{\text{windage}}$, $\gamma_{\text{partial}}$) 5% for $\Phi_{\text{incidence}}$, and 7% for 3 out of 10 ($\Phi_{\text{clearance}}$, $\Phi_{\text{sector}}$, $\Phi_{\text{partial}}$). From the results of this thesis, it suggests that the multiplicative term $N_{\text{sector}}$ should be employed to both sector and pumping losses for partial admission machines.

**Selection of a Working Fluid and Cycle Conditions**

The selection of a working fluid is an important parameter for a cycle. There are many working fluids to select from and more are being developed. It’s difficult to give absolute criteria for selecting a working fluid because the fluids are very diverse in their properties and behaviours across a range of conditions. But there are a few general rules that can be applied when selecting a fluid for an ORC employing a single stage axial impulse turbine.

1) The critical temperature of the fluid should be greater than the coolant source temperature.
2) A fluid that provides a large enthalpy differential between the geothermal source temperature and the coolant source temperature will be a good candidate because there will be a large available enthalpy to exploit in the turbine.
3) Fluids with narrow vapour domes will have a lesser impact from pinch points in both the evaporator and the condenser. This will increase the $\beta$ and decrease the cooling work.
4) Fluids that have low density and viscosity at the saturated liquid pressure (for the coolant source temperature) and turbine outlet temperature will have lower shear and drag forces in the turbine.
5) With using a supersonic stator nozzle the pressure ratio is critical to the exit velocity. Too large of a pressure ratio is going to require a long expansion section and this could lead to excessive nozzle arc lengths and this can lead to partial admission. Also, the greater the exit velocity the greater the tip speed and the faster the turbine will need to operate. In general, without considering practical limitations, the faster the turbine and the smaller the turbine the more efficient it will be. But in reality there are limitations to speed and size so a good design will consider these limitations. A simple nozzle calculation for a set of resource temperatures and a range of inlet pressures will give an indication of the nozzle exit speed and thereby the required speed and size of a turbine. Fluids can be initially screened based on this simple analysis.
There are as well environmental, health, safety, cost, availability, material compatibility and serviceability issues to consider. But these are too application specific to provide selection criteria. It’s up to the designer to consider these issues and balance these issues with the performance of the turbine and the cycle.

**Design Characteristics of Single Stage Axial Impulse Turbine**

Supersonic single stage axial impulse turbines were investigated for use with geothermal binary power stations because of their potential for being small, simple and inexpensive. The loss model calibrated was for a very small turbine and as such the model’s results should be used with caution when extrapolating out to larger powers. However, the model did show, with the inclusion of the $N_{sector}$ term, little deviation from the default models so it suggest that the model is applicable to a range of admissions and powers tested in this thesis. Based on the programs developed in this thesis (AXIAL and SSAL) the turbines that are designed have the potential to attain efficiencies upwards of 87%. Being single stage machines, the turbines must turn the fluid in the rotor a large degree in order to extract the maximum amount of energy. This leads to designs that have large stator outlet angle with large blade inlet and outlet angles as well. And because the turbines are supersonic, the exit velocity of the stator nozzle is very high which requires a high blade tip speed. To achieve a large blade tip speed either the rotational speed must be high or the diameter must be large. High speeds and large diameters have large associated drag losses. So a balance of size and speed is required. But these turbines are still much smaller and faster than traditional multistage axial machines. They require couplings, bearings and generators capable of high speed operation. A small, fast, full admission turbine is desirable from the point of view of losses and efficiency, but consideration needs to be given to the practical limitations for each application.

### 13.1 Future Work

A great deal of time and effort was put into the development of the ORC test rig. The capabilities of which extend beyond the experimentation that took place in this thesis. The ultimate goal of the greater research objective that this thesis is a part of is to be able to comprehensively investigate power conversion cycles for low temperature heat sources, particularly geothermal. It would be of great interest to develop models of all the components in the ORC test rig (i.e. Heat Exchangers, Generator, Piping, and Pump) and see how the cycle analysis is influenced by incorporating real models of all the components not just the expander.
The ORC test rig can also be used to investigate variations in the impulse turbine, such as rotor and stator geometry variations (i.e. aspect ratio, pitch-chord ratio, stator nozzle spacing, shrouding, admission rates, and nozzle design). The turbine can also be altered to investigate rotors based on zero-exit swirl and symmetric velocity triangles.

Also future work could include the investigation of binary fluids such as R245fa and R134a mixtures. These binary fluid mixtures show promise in achieving higher $\beta$ values because they do not evaporate isothermally. Their varying evaporative temperature profile allows for better thermal fluid utilisation and higher $\beta$ values.

**ORC Test Rig**

There are many facets of an ORC that can be investigated using the ORC test rig. One such application that began at the end of this thesis was the investigation of using binary refrigerant mixtures as working fluids in ORC’s. A suite of tests were conducted in the loop using the dual nozzle turbine configuration with a range of R245fa-R134a fluid mixtures. The tests were not analysed as part of this thesis because accurate and validated fluids properties for binary refrigerant mixtures were not available. However, a binary refrigerant working fluid is an area of study that has great potential for cycle efficiency improvement. Future work in the ORC test rig that would be informative is continued binary fluid testing with a range of fluids and expanders.

The test rig should also be used to continue testing variations of the impulse turbine such as variations in blade height to tip clearance ratio, degree of reaction in the blade, stator nozzle spacing, rotor pitch-chord ratios etc. Testing variations of the impulse turbine will help to improve the robustness of the loss model developed in this thesis and make its range of applicability greater when incorporated into the cycle analysis software ORCCA.

**ORCCA Cycle Analysis Code**

The cycle analysis code written in this thesis could be expanded and developed further making it a more comprehensive tool. Pipe friction and heat exchanger modules were written but have not been validated against experimental data. These modules would be very useful in modelling overall cycle efficiencies. To make these modules useful though a metric based on cost or feasibility would have to be implemented to the pipe and heat exchanger models because friction losses in pipe and heat exchangers can be made negligible if the size of the pipe is large enough and the number and/or size of the heat exchangers is large enough. To decide what is feasibly maximum size of piping and heat exchangers a good cost model would have to be developed.
Loss Model

Future recommendations for work on turbine performance prediction using the test facilities that were developed during the course of this thesis would be as follows:

- Investigation of stator nozzle spacing influence on partial admission losses
- The influence of leading edge and trailing edge shape on the trailing edge and incidence losses
- CFD investigation of the stator and rotor interactions and their impact on turbine performance
- An experimental analysis of clearance losses that focuses on the influence of surface roughness, blade height, blade width, rotational speed, and blade passage pressure
- Investigation of blade passage shape, curvature, and surface finish on profile losses
- Investigation of the optimum relationship between meridional length, blade arc radius to minimise passage losses and windage losses
- Investigation of the influence of aspect ratio (passage height to width ratio) and Zweifel coefficient
- Investigate the influence of mass flow rate on the loss model. Test larger turbines at higher mass flow rates and see if the loss coefficients are dependent on the mass flow rate.

The ORC test rig developed as part of this thesis has many potential applications to experimentally investigate power conversion technology for low temperature sources such as geothermal, solar-thermal, and waste heat recovery.
References


Cuenot N., FJ, Fritsch D, Genter A, Szablinski D, Pfalzwerke AG. 2008, 'The European EGS project at Soultz-sous-Forêts: from extensive exploration to power production', *IEEE*.


Legmann, H 2003, 'The Bad Blumau geothermal project: a low temperature, sustainable and environmentally benign power plant', *Geothermics,* vol. 32, no. 4-6, pp. 497 - 503.


Stenning, AH 1953, Design of Turbines for High-energy-fuel Low-power-output Applications, M.I.T. Dynamic Analysis and Control Laboratory, Massachusetts Institute of Technology. Dynamic Analysis Control Laboratory.


Appendix A. Cycle Analysis Code (ORCCA)

The cycle analysis code developed in this thesis was written in Matlab and it evaluates condensing rankine cycles for a range of fluids and operating conditions. It allows for investigating the influence of many different parameters on the performance of a cycle. The cycle also includes the loss model for a single stage impulse turbine to calculate real turbine efficiency a given cycle.

The process flow of the program is as follows;

1. F1a_Main.m
   a. F1b_List.m
   b. F1b_Inputs.m
   c. F1b_States.m
      i. -F1b_Nozzle.m
      ii. -F1b_Velocity_Triangle.m
         1. -F1b_Losses.m
         iii. -F1b_Pinch_Rhx.m
         iv. -F1b_SC_State_3.m
         v. -F1b_SC_State_4.m
   d. F1b_Pinch_Ehx
   e. F1b_Pinch_Chx
   f. F1b_Cycle_Calc
   g. F1b_Plot_TS
   h. F1b_Plot_Cycle
   i. F1b_Store_Results
   j. F1b_Write_Results
   k. F1b_Gather_Files

Several post processing codes were written to help analyse the results from the cycle analysis. A script (F3a_BvP.m) was written examine the relationship between $\beta$ and several key operating parameters (i.e. $P_2$, $CHX_{dT}$, $EHX_{dT}$, $E$, and Recuperation). Another performance investigating script (F3a_Performance.m) was written to generate performance contour maps of optimum operating conditions for various ranges of $T_c$ and $T_h$. And another script (F3a_FLD_Map.m) to compare fluids and create contour maps of optimum fluid and operating conditions.

The codes are listed here for reference.
fclose('all'); close all; clear all; clc;

%%%System Setup

set(0,'defaultAxesFontName', 'Times New Roman') % Set Axis Font
set(0,'defaultTextFontName', 'Times New Roman') % Set Text Font
font_size = 10; % Set Font size

%%%Cycle Analysis

sheet = 'Top_R218_Cal'; % i.e. sheet = 'SF6'
workers = 4; % Number of CPU's (fastest at half of CPUs)
mk_list = 1; % Make List Flag
mk_annual = 0; % Make Annual Calculations Flag
mk_plots = 0; % Make Plots Flag
mk_calcs = 1; % Make Cycle Calculations Flag
mk_cool = 1; % Make Coolant Losses Flag
mk_loss = 1; % Make Loss Model Calculations Flag
mk_pipe = 0; % Make Pipe Friction Calculations Flag
mk_phe = 0; % Make PHE Flag
mk_tri = 0; % Make Trialateral Cycle Flag

Rhx = 5; % Regen hx temperature differential, must be greater than degC
Ph = 200; % Pressure of Thermal Fluid Stream (i.e. brine pressure in evap)
Pc = 102.0; % Pressure of Cooling Fluid Stream (i.e. water pressure in cond)
% If using air, Pc >101, use 102.00 for Air
FLD_h = 'Water'; % Thermal Fluid: Water, Thermia, Ethylene-Glycol, Exhaust
FLD_c = 'Air.ppf'; % Cooling Fluid, Water or Air (Air.ppf)

%%%Loss Model

power_error = 0.01; % Set the acceptable power error for Loss Loop
its_a = 1; % Number of iterations per a_I variable (5)
its_Z = 1; % Number of iterations per u_U variable (5)
a_I_min = 80*pi/180; % Minimum nozzle absolute exit angle, radians
a_I_max = 80*pi/180; % Maximum nozzle absolute exit angle, radians
Z_min = 0.50; % Minimum speed work parameter
Z_max = 0.50; % Maximum speed work parameter
N_noz_its = 20; % Number of nozzles iterates (20)
N_bld_its = 20; % Number of nozzles iterates (20)
RPM_max = 25000; % Maximum allowable RPM
% Use this option to print props to data_mass.xls for determining how much
% refrigerant to put into the small loop, 'Yes', 'No'.
print_mass = 'No';

if mk_list == 1
    t_list_start = tic;
    F1b_List(sheet, mk_annual);
    t_list = toc(t_list_start);
    fprintf('Finished Making Lists\n');
else
    t_list = 0;
end

%%%Import Colors

[plot_colors] = my_gray();

%%%Setup File Directories and FIDs

directory = sprintf('%s_Results',sheet);

166
if exist(directory,'dir') == 0
    mkdir(directory);
end

%%%Setup Temporary Parallel Log Files Director
directory_log = sprintf('%s\PP_Logs',directory);
if exist(directory_log,'dir') == 7
    rmdir(directory_log, 's')
end
mkdir(directory_log);

%Make file names for data in and out
data_in = sprintf('%s\%s_inputs.txt',directory, sheet);
data_out = sprintf('%s\%s',directory, sheet);

%%%Determine number of lines from input file%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fid = fopen(data_in, 'r');
ng = 0;
while fgets(fid) ~= -1,
    ng = ng+1;
end
fclose(fid);

matlabpool(workers) %Open Pool of Workers for Parallel Computing
t_loop_start = tic;
t_cycle_sum = 0;
if mk_calcs == 1 %Make Calculations Flag
    spmd
        fid_in = fopen(data_in, 'r');  %open text file for writing results
data_all = textscan(fid_in, '%s','delimiter', '
);
        % for ig = 1:ng
        %    t_cycle_start = tic; %Initial cpu time
        %    %%%Read inputs%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
data_line = data_all;
        % [ETAC,ETAT,FLD,FRCT1,FRCT2,Tgeo,Tamb,P2,Ehx,...
        Chx,power,regen,T5,T1,Tcrit,TS_Smin,TS_Smax,...
        TS_Min,TS_Tmax,mmol,Tmax] = F1b_Inputs(data_line);
        if regen == 0
            reg_tag = 'R0';
        else
            reg_tag = 'R1';
        end
        FLD_save = textscan(FLD, '%{^.');
        FLD_save = {FLD_save{1}];
directory_iter = ...
sprintf('%s\%s - P2_%.0f T5_%.0f T1_%.0f Ehx_%.0f Chx_%.0f E_%.0f %s', ...
directory, FLD_save, P2, T5, T1, Ehx, Chx, power, reg_tag);
        %%%Define State Points in Matrix props%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        try
            [props,mf,qhx,Scrit,wf_reg_h,wf_reg_c,ETAT,check,error] = ...
            F1b_States(T1,P2,T5,ETAC,ETAT,power,Rhx,Pcrit,Tcrit,regen,...
directory_iter,FLD,FRCT1,FRCT2,mk_plots,mk_loss,...
mk_tri,mk_phe,plot_colors,RPM_max,...
its_a,its_ZZ_min, Z_max,a_H_min,a_H_max,N_noz_its,N_bld_its,power_error);
        catch ME
            check = 1;
        end
    end
end

167
% fprintf('
');
% fprintf('Error 13: REFPROP Error in states\n');
% error = 13;
% [props,mf,qhx,Scrit,wf_reg_h,wf_reg_c,ETAT] = deal(0);
% directory_iter = sprintf('%d\n', check);
% end
fprintf('Run States(), check= %.0f\n', check)

%%%Pinch Analysis for Evp and Cnd%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if check ~= 1
    try
    % Run Evp Pinch and PHE
    [Tha,Thb,mfh,ch,wf_evp_h,wf_evp_c,check,error] = ...
    F1b_Pinch_Ehx(Ehx,mf,regen,FLD_h,Ph,directory_iter,...
    plot_colors,mk_plots,props,FLD,FRCT1,FRCT2,mk_phe,power);
    % Run Cnd Pinch and PHE
    [Tca,Tcb,mfc,cc,Dc,wf_cnd_h,wf_cnd_c,check,error] =...
    F1b_Pinch_Chx(Chx,mf,regen,FLD_c,Pc,directory_iter,...
    plot_colors,mk_plots,props,FLD,FRCT1,FRCT2,mk_phe,power,Tamb);
    catch ME
        check = 1;
        fprintf('Error 14: REFPROP Error in pinch\n');
        error = 14;
        [Tha,Thb,mfh,ch,wf_evp_h,wf_evp_c] = deal(0);
        [Tca,Tcb,mfc,cc,Dc,wf_cnd_h,wf_cnd_c] = deal(0);
    end
    [Tha,Thb,mfh,ch,wf_evp_h,wf_evp_c] = deal(0);
    [Tca,Tcb,mfc,cc,Dc,wf_cnd_h,wf_cnd_c] = deal(0);
end
else
    [Tha,Thb,mfh,ch,wf_evp_h,wf_evp_c] = deal(0);
    [Tca,Tcb,mfc,cc,Dc,wf_cnd_h,wf_cnd_c] = deal(0);
end
fprintf('Run Pinch(), check = %.0f\n', check)

%%%Cycle Calculations%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if check ~= 1
    try
    % Cycle Calculations
    [qi,qo,wi,wo,wnet,wf_pipe,wf_hx,wf_total,dht,B,n1st,...
    n2nd,ETAT,ncarnot,check,error] = ...
    F1b_Cycle_Calc(props,mf,mfh,mfc,ch,Tca,Tha,FLD,Dc,...
    Pc,wf_evp_h,wf_evp_c,wf_cnd_h,wf_cnd_c,wf_reg_h,...
    wf_reg_c,check,regen,mk_pipe,mk_cool,mk_phe,ETAT,...
    directory_iter);
    catch ME
        [qi,qo,wi,wo,wnet,wf_pipe,wf_hx,wf_total,dht,B,n1st,...
        n2nd,ETAT,ncarnot,error] = deal(0);
    end
    else
    [qi,qo,wi,wo,wnet,wf_pipe,wf_hx,wf_total,dht,B,n1st,...
    n2nd,ETAT,ncarnot,error] = deal(0);
    end
end
fprintf('Run Cycle(), check = %.0f\n', check)

%%%Plot T-S%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if check ~= 1
    try
    if mk_plots == 1
        [Slo,Shi,Tlo, Thi] = F1b_Plot_TS(props,Pcrit,Tcrit,...
plot_colors, TS_Smin, TS_Smax, TS_Tmin, TS_Tmax, FLD...
FRCT1, FRCT2, Tgeo);
end
catch ME
{Slo, Shi, Tlo, Thi} = deal(0);
end
else
{Slo, Shi, Tlo, Thi} = deal(0);
end
fprintf('Run PlotTS(), check = %.0f\n', check)

%%% Plot Cycle
if check ~= 1 && mk_plots == 1
try
{fig1} = F1b_Plot_Cycle(Tha, Thb, Tca, Tcb, mf, mfh, mfc, qi, qo, qhx, ...
wi, wo, wf_pipe, wf_hx, wf_total, wnet, Slo, Shi, Tlo, Thi, B, ETAT,...
P2, props, Perit, FLD, FRCT1, FRCT2, regen, plot_colors, font_size);
catch ME
check = 1;
fprintf('Error 17: REFPROP Error in plot_cycle\n');
error = 17;
end
fprintf('Run Plot Cycle(), check = %.0f\n', check)
else
fig1 = 0;
end

%%% Store Results To Results{ } and Save Figure
{results} = F1b_Store_Results(B, mf, mfh, mfc, qi, qo, wi, wf_total,...
wo, wnet, n1st, n2nd, Tha, Thb, qhx, Tca, Tcb, dht, ETAC, ETAT,...
directory_iter, T1, P2, T5, props, FLD, FRCT1,...
FRCT2, Ehx, Chx, check, error, regen, fig1, mk_plots);
fprintf('Run Store Results()\n')

%%% Write Results to Log Text File
F1b_Write_Results(results, directory_log, check)
if check ~= 1
fprintf('Run Write Results()\n')
end
t_cycle = toc(t_cycle_start);
t_cycle_sum = t_cycle+t_cycle_sum;
t_cycle_ave = t_cycle_sum/ig;
fprintf('Cycle Time: %.3f, Cycle Ave: %.3f\n', t_cycle, t_cycle_ave)
end
end
end

%%% Gather Files All Log Files Into Single Output File
F1b_Gather_Files(directory, directory_log, sheet)
fprintf('Simulation Complete\n');
t_cycle_ave = t_loop/ng;
fprintf('Loop Time: %.2f, Average Cycle Time: %.4f\n', t_loop, t_cycle_ave)
fprintf('List Time: %.2f, Entries: %.2f, Time Per Entry, %.2f\n', t_list, ng, t_list/ng)

%%% Print P,T,D results to data_mass.xls
if strcmp(print_mass,'Yes') == 1

169
print_P = xlswrite('data_mass.xlsx', props(:,1), '1', 'C3');
print_T = xlswrite('data_mass.xlsx', props(:,2), '1', 'D3');
print_D = xlswrite('data_mass.xlsx', props(:,6), '1', 'E3');
end

close all
fclose('all');

%%% Close Workers
if matlabpool('size')>0
    matlabpool close
end
function F1b_List(sheet, mk_annual)

wks = sprintf('%s_list.xlsm',sheet);
[data, names] = xlsread(wks, 'list', 'E17:M72');
iter = xlsread(wks, 'list', 'B13:J13');
vars = xlsread(wks, 'list', 'N17:AC72');
FRCT1 = xlsread(wks, 'list', 'C17:C72');
FRCT2 = xlsread(wks, 'list', 'D17:D72');
T_annual = xlsread(wks, 'list', 'L3:L14');
directory = sprintf('%s_Results',sheet);
mkdir(directory);
list_file = sprintf('%s\%s\%s_inputs.txt',directory, sheet, sheet);
fid = fopen(list_file, 'wt');
fprintf(fid, 'nRead Fluids Complete
n');
%create all combinations and write to file
count_FLD = size(names,1);
for i_FLD = 1:count_FLD
    %Define Tgeo inputs
    Tgeo_min = vars(i_FLD,1);
    Tgeo_max = vars(i_FLD,2);
    Tgeo_its = iter(1,1);
    if Tgeo_min == Tgeo_max
        Tgeo = Tgeo_max;
    else
        Tgeo = Tgeo_min:(Tgeo_max-Tgeo_min)/Tgeo_its:Tgeo_max;
    end
    %Define Tamb inputs
    if mk_annual == 1
        Tamb = T_annual;'
    elseif mk_annual == 0
        Tamb_min = vars(i_FLD,3);
        Tamb_max = vars(i_FLD,4);
        Tamb_its = iter(1,2);
        if Tamb_min == Tamb_max
            Tamb = Tamb_max;
        else
            Tamb = Tamb_min:(Tamb_max-Tamb_min)/Tamb_its:Tamb_max;
        end
    end
    %Define Ehx inputs
    Ehx_min = vars(i_FLD,6);
    Ehx_max = vars(i_FLD,7);
    Ehx_its = iter(1,4);
    if Ehx_min == Ehx_max
        Ehx = Ehx_max;
    else
        Ehx = Ehx_min:(Ehx_max-Ehx_min)/Ehx_its:Ehx_max;
    end
    %Define Chx inputs
    Chx_min = vars(i_FLD,8);
    Chx_max = vars(i_FLD,9);
    Chx_its = iter(1,5);
    if Chx_min == Chx_max
        %
    end
end
end
end
Chx = Chx_max;
else
Chx = Chx_min:(Chx_max-Chx_min)/Chx_its:Chx_max;
end

%Define ETAC inputs
ETAC_min = vars(i_FLD,10);
ETAC_max = vars(i_FLD,11);
ETAC_its = iter(1.6);
if ETAC_min == ETAC_max
ETAC = ETAC_max;
else
ETAC = ETAC_min:(ETAC_max-ETAC_min)/ETAC_its:ETAC_max;
end

%Define ETAT inputs
ETAT_min = vars(i_FLD,12);
ETAT_max = vars(i_FLD,13);
ETAT_its = iter(1,7);
if ETAT_min == ETAT_max
ETAT = ETAT_max;
else
ETAT = ETAT_min:(ETAT_max-ETAT_min)/ETAT_its:ETAT_max;
end

%Define power inputs
Power_min = vars(i_FLD,14);
Power_max = vars(i_FLD,15);
Power_its = iter(1,8);
if Power_min == Power_max
Power = Power_max;
else
Power = Power_min:(Power_max-Power_min)/Power_its:Power_max;
end

%Define reg inputs
Reg_its = iter(1,9);
if Reg_its > 1
Reg = 0:1:1;
else
Reg = vars(i_FLD,16);
end

%Define P2 inputs
P2_min = refpropm('P', 'T', min(Tamb), 'Q', 0, names{i_FLD}, {FRCT1 FRCT2});
P2_max = vars(i_FLD,5);
P2_its = iter(1,3);
if P2_min >= P2_max
P2 = P2_max;
elseif P2_its > 1
P2 = P2_min:(P2_max-P2_min)/P2_its:P2_max;
elseif P2_its <= 1
P2 = P2_max;
end

count_Tgeo = size(Tgeo,2);
count_Tamb = size(Tamb,2);
count_P2 = size(P2,2);
count_Ehx = size(Ehx,2);
count_Chx = size(Chx,2);
count_ETAC = size(ETAC,2);
count_ETAT = size(ETAT,2);
count_Power = size(Power,2);
count_Reg = size(Reg,2);
for i_Tgeo = 1:count_Tgeo
  for i_Tamb = 1:count_Tamb
    for i_P2 = 1:count_P2
      for i_Ehx = 1:count_Ehx
        for i_Chx = 1:count_Chx
          for i_ETAC = 1:count_ETAC
            for i_ETAT = 1:count_ETAT
              for i_Power = 1:count_Power
                for i_Reg = 1:count_Reg
                  fprintf(fid, names{i_FLD});
                  fprintf(fid, 't');
                  fprintf(fid, num2str(FRCT1(i_FLD)));
                  fprintf(fid, 't');
                  fprintf(fid, num2str(FRCT2(i_FLD)));
                  fprintf(fid, 't');
                  fprintf(fid, num2str(Tgeo(i_Tgeo)));
                  fprintf(fid, 't');
                  fprintf(fid, num2str(Tamb(i_Tamb)));
                  fprintf(fid, 't');
                  fprintf(fid, num2str(P2(i_P2)));
                  fprintf(fid, 't');
                  fprintf(fid, num2str(Ehx(i_Ehx)));
                  fprintf(fid, 't');
                  fprintf(fid, num2str(Chx(i_Chx)));
                  fprintf(fid, 't');
                  fprintf(fid, num2str(ETAC(i_ETAC)));
                  fprintf(fid, 't');
                  fprintf(fid, num2str(ETAT(i_ETAT)));
                  fprintf(fid, 't');
                  fprintf(fid, num2str(Power(i_Power)));
                  fprintf(fid, 't');
                  fprintf(fid, num2str(Reg(i_Reg)));
                  fprintf(fid, 't');
                  % write in Tc, Pc, Ts_Axes, Mol W, Tmax
                  for i_data = 1:8
                    fprintf(fid, num2str(data(i_FLD,i_data)));
                    fprintf(fid, 't');
                  end
                  fprintf(fid, '\n');
                end
              end
            end
          end
        end
      end
    end
  end
end
fprintf('ulist for %s complete\n',names{i_FLD});
end
fclose(fid);
fprintf('ulist for data_%s complete\n',sheet);
fprintf('\nlist complete');
function \{ ETAC, ETAT, FLD, FRCT1, FRCT2, Tgeo, Tamb, P2, Ehx,...
     Chx, power, regen, T5, T1, Tcrit, TS_Smin, TS_Smax,...
     TS_Tmin, TS_Tmax, mmol, Tmax \} = F1b_Inputs(data_line)

data_str = textscan(data_line, '%s', 20, 'delimiter', '\');
data = data_str{:};
FLD = data{1};
FRCT1 = str2double(data{2});
FRCT2 = str2double(data{3});
Tgeo = str2double(data{4}); % K
Tamb = str2double(data{5}); % K
P2 = str2double(data{6}); % kpa user enters P2 if desired, otherwise left as 0.
Ehx = str2double(data{7}); % K
Chx = str2double(data{8}); % K
ETAC = str2double(data{9}); % pump efficiency, 0.xx format
ETAT = str2double(data{10}); % turbine efficiency, 0.xx format
power = str2double(data{11}); % kW
regen = str2double(data{12}); % ---
T5 = Tgeo - Ehx; % K
T1 = Tamb + Chx; % K
Tcrit = str2double(data{13}); % K
Perit = str2double(data{14}); % kpa
TS_Smin = str2double(data{15}); % J/kg-K
TS_Smax = str2double(data{16}); % J/kg-K
TS_Tmin = str2double(data{17}); % K
TS_Tmax = str2double(data{18}); % K
mmol = str2double(data{19}); % kg/mol
Tmax = str2double(data{20}); % K
fprintf('\nStart %s for ', FLD)
fprintf('P2 = %.2f and T5 = %.2f\n', {P2 T5})
function [props,mf,qhx,Scrit,wf_reg_h,wf_reg_c,ETAT,check,error] = F1b_States(T1,P2,T5,ETAC,ETAT,power,Rhx,Pcrit,Tcrit,regen,directory_iter,FLD,FRCT1,FRCT2,mk_plots,mk_loss,mk_tri,mk_phe,plot_colors,RPM_max,its_a,its_Z,Z_min,Z_max,a_II_min,a_II_max,N_noz_its,N_bld_its,power_error)

% P Pressure [kPa]
% T Temperature [K]
% D Density [kg/m3]
% H Enthalpy [J/kg]
% S Entropy [J/(kg*K)]
% U Internal energy [J/kg]
% C Cp [J/(kg K)]
% O Cv [J/(kg K)]
% K Ratio of specific heats (Cp/Cv) [-]
% A Speed of sound [m/s]
% X liquid phase and gas phase composition (mass fractions)
% V Dynamic viscosity [Pa*s]
% L Thermal conductivity [W/(m K)]
% Q Quality (vapor fraction) (kg/kg)
% I Surface tension [N/m]

%determine evap pressure
props = zeros(11,8);  %property matrix [P T Q H S D C K]
check = 0;
error = 0;
mf = 0;
qxh = 0;
wf_reg_h = 0;
wf_reg_c = 0;
Scrit = refpropm('S', 'T', Tcrit, 'P', Pcrit, FLD, {FRCT1 FRCT2});

%pressure check no superheat
if P2<1 && T5 < Tcrit
    P2 = refpropm('P', 'T', T5, 'Q', 1, FLD, {FRCT1 FRCT2});
%pressure check with superheat
elseif P2<1 && T5 > Tcrit
    P2 = 0.95*Pcrit;
%error check I - ensure T5 > sat. liq. temp at P2
elseif P2>0 && P2<Pcrit && T5<Tcrit
    Tcheck = refpropm('T', 'P', P2, 'Q', 0, FLD, {FRCT1 FRCT2});
    if Tcheck > T5
        check = 1;
        fprintf('
');
        fprintf('Error 01: States, Sat. liq. temp. for given P2 > T5
');
        error = 1;
        return
    end
%error check II - ensure T5 is > Tcrit if P2 > Pcrit
elseif P2>Pcrit && T5<Tcrit && mk_tri ~= 1
    Tcheck = refpropm('T', 'P', P2, 'Q', 0, FLD, {FRCT1 FRCT2});
    if Tcheck > T5
        check = 1;
        fprintf('
');
        fprintf('Error 02: States, P2 > Pcrit and T5 < Tcrit
');
        error = 2;
        return
% State Point 1 - pump inlet
% error check III - ensure T1 < Tcrit
if T1 >= Tcrit
    check = 1;
    fprintf('Error 03: States, T1 > Tcrit\n');
    error = 3.0;
    return
end

% error check IV - ensure T1 < T5
if T1 > T5
    check = 1;
    fprintf('Error 04: States, T1 > T5\n');
    error = 4;
    return
end

\{P,T,Q,H,S,D,C,K\} = refpropm('PTQHSDCK', 'T', T1, 'Q', 0, FLD, \{FRCT1 FRCT2\});
props(1,1:8) = \{P,T,Q,H,S,D,C,K\};

% error check V - P2 < P1
if P2 <= 1.2*props(1,1)
    check = 1;
    fprintf('Error 05: States, P2 <= 1.2xP1 for given T1\n');
    error = 5;
    return
end

% Set Cycle Directory For
if mk_plots == 1
    if exist(directory_iter,'dir') == 0
        mkdir(directory_iter);
    end
end

% State Point 2 - pump outlet (reversible)
\{P,T,Q,H,S,D,C,K\} = refpropm('PTQHSDCK', 'P', P2, 'S', props(1,5), FLD, \{FRCT1 FRCT2\});
props(2,1:8) = \{P,T,Q,H,S,D,C,K\};

% State Point 2s - pump outlet (irreversible)
H2s = ((props(2,4)-props(1,4))/\eta) + props(1,4);
\{P,T,Q,H,S,D,C,K\} = refpropm('PTQHSDCK', 'P', P2, 'H', H2s, FLD, \{FRCT1 FRCT2\});
props(2,1:8) = \{P,T,Q,H,S,D,C,K\};

% State Point 3 - for ORC sat liq state in evap
if P2 < Pcrit
    \{P,T,Q,H,S,D,C,K\} = refpropm('PTQHSDCK', 'P', P2, 'Q', 0, FLD, \{FRCT1 FRCT2\});
    props(3,1:8) = \{P,T,Q,H,S,D,C,K\};
    if mk_tri == 1 && T5 < T
        props(3,1:8) = props(2,:);
    end
end
%State Point 4 - for ORC sat vap state in evap
if P2 < Pcrit
    props(4,1:8) = [P,T,Q,H,S,D,C,K];
end

%State Point 5 - turbine inlet
if P2 < Pcrit && abs(T5 - props(4,2)) < 0.0001
    [P,T,Q,H,S,D,C,K] = refpropm('PTQHSDCK', 'T', T5, 'Q', 1, FLD, [FRCT1 FRCT2]);
else
end
props(5,1:8) = [P,T,Q,H,S,D,C,K];

%Correct state point 4 for trilateral cycle
if mk_tri == 1 && props(4,2) > props(5,2)
    props(4,1:8) = props(5,:);
end

%State Point 6 - turbine outlet (reversible)
    [P,T,Q,H,S,D,C,K] = refpropm('PTQHSDCK', 'P', props(1,1), 'S', props(5,5), FLD, [FRCT1 FRCT2]);
    props(6,1:8) = [P,T,Q,H,S,D,C,K];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Determine actual turbine efficiency for given cycle conditions
%Nozzle Negative Gamma Error Check
if mk_loss == 1
    %Nozzle Negative Gamma Error Check
    if mean([props(5,8) props(6,8)]) < 0
        check = 1;
        fprintf('Error 06: Returned a negative Gamma From state 5 to 6\n');
        error = 6;
        return
    end
%Nozzle Negative Sound Speed Error Check
if refpropm('A', 'T', props(6,2), 'P', props(6,1), FLD, [FRCT1 FRCT2]) < 0
    check = 1;
    fprintf('Error 07: Returned a negative Sound Speed at 6\n');
    error = 07;
    return
end
%Nozzle Quality Check
if props(6,3) < 1
    check = 1;
    fprintf('Error 10: Condensing Liquid At Nozzle Exit\n');
    error = 07;
    return
end

%Begin Loss Model While Loop
W_mech = 0;
i_power_attempts = 0;
ETAT_i = 0.0738*log(power)+0.42;
power_i = power/ETAT_i;
while abs(W_mech-power)/power > power_error %Search until W_mech is equal to desired power
% Run Nozzle Function
[V_II,mf,A_s_II_sum,A_II_At] = F1b_Nozzle(power_i,props,...
FLD,FRCT1,FRCT2);
fprintf('--- loss model iteration for power_i: %.0f, mf: %.2f\n', power_i, mf)
% Run axial loss model function for calculated turbine efficiency
{ETAT,W_mech} = F1b_Velocity_Triangle(V_II, mf,A_s_II_sum,...
A_II_At,directory_iter,props,mk_plots,FLD,RPM_max,...
its_a,its_Z,Z_min,Z_max,a_II_min,a_II_max,N_noz_its,N_bld_its);

% error check 12 - ETAT = 0
if ETAT == 0
  check = 1;
  fprintf('\n');
  fprintf('Error 09: ETAT = 0\n');
  error = 09;
  return
end
power_i = power_i+(randi([75 125],1,1)/100)*(power_i-W_mech)/ETAT;
i_power_attempts = i_power_attempts + 1;
% Error Check if no solution is found
if i_power_attempts > 10
  check = 1;
  fprintf('\n');
  fprintf('Error 21: No Loss Model Solution Found\n');
  error = 21;
  return
end
end

%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%
%%% State Point 6s - turbine outlet (irreversible)
H6 = props(5,4)-(ETAT*(props(5,4)-props(6,4)));
{P,T,Q,H,S,D,C,K} = refpropm('PTQHSDCK', 'P', props(1,1), 'H', H6, FLD,
{FRCT1 FRCT2});
props(6,1:8) = {P,T,Q,H,S,D,C,K};

% Define Mass Flow Rate For No loss Model
if mk_loss == 0
  mf = (power_i*1000)/(props(5,4) - props(6,4));
end

%%% State Point 7 - sat vap condensor if turbine expands into vap
if props(6,3)>1
  [P,T,Q,H,S,D,C,K] = refpropm('PTQHSDCK', 'P', props(1,1), 'Q', 1, FLD,
{FRCT1 FRCT2});
  props(7,1:8) = [P,T,Q,H,S,D,C,K];
else
  props(7,:) = props(6,:);
end

% Quality check SC
if P2 > Pcrit && props(6,2) < 0.98*Tcrit
  n=5;
  T_i = props(6,2):(0.98*Tcrit - props(6,2))/(n-1):0.98*Tcrit;
  Sg_check = zeros(size(T_i));
  for k=1:n;
    [S]= refpropm('S', 'T', T_i(k), 'Q', 1, FLD,
{FRCT1 FRCT2});
    Sg_check(k) = S;
  end
Sq_check = max(Sg_check);
if props(5,5) < Sq_check && mk_tri ~= 1;
    check = 1;
    fprintf('%un');
    fprintf('Error 11: States, SC phase change expansion'un');
    error = 11;
    return
end
end

Recuperation Inlet State Points 8 and 10
%State Point 8 (6xi) - high temp. fluid inlet to recuperation
if props(6,2) > props(2,2)+2*Rhx && regen == 1
    props(8,:) = props(6,:);
else
    props(8,:) = 0;
end
%State Point 10 (2xi) - low temp. fluid inlet to recuperation
if props(6,2) > props(2,2)+2*Rhx && regen == 1
    props(10,:) = props(2,:);
else
    props(10,:) = 0;
end
Recuperations Outlet State Points 9 and 11
if props(6,2) > props(2,2)+2*Rhx && regen == 1 && check == 0
    \{qhx_h,qhx_c,~,~,wf_reg_h,wf_reg_c,check,\] = F1b_Pinch_Rhx(mf,Rhx,...
    \{qhx_h,qhx_c,~,~,wf_reg_h,wf_reg_c,check,\[ = F1b_Pinch_Rhx(mf,Rhx,...
    \{P,T,Q,H,S,D,C,K\] = refpropm('PTQHSDCK','P',props(6,1),'H',props(8,4)-Hhx_h,FLD,
    \{P,T,Q,H,S,D,C,K\];
    \{P,T,Q,H,S,D,C,K\] = refpropm('PTQHSDCK','P',props(2,1),'H',props(10,4)+Hhx_c,FLD,
    \{P,T,Q,H,S,D,C,K\];
else
    props(9,:) = 0;
    props(11,:) = 0;
    qhx = 0;
    fprintf('un');
    fprintf('Error 12: Recuperation Not Possible'un');
    check = 1;
end
State Point 3 and 4 - For Supercritical Phase
if P2 > Pcrit && T5 > Tcr
    S2_11 = max(props(11,5),props(2,5));
    if S2_11 < Scrit
        \{S3_2\} = F1b_SC_State3(S2_11,Scrit,P2,FLD);
        \{P,T,Q,H,S,D,C,K\] = refpropm('PTQHSDCK', 'P', P2, 'S', S3_2, FLD,
        \{P,T,Q,H,S,D,C,K\];
    end
end
elseif P2 > Pcrit && T5 < Tcrit && mk_tri == 1
    props(3,:) = props(2,:);
end

if P2 > Pcrit
    if props(5,5)>Scrit
        [S4_2] = F1b_SC_State4(Scrit,P2,FLD,props);
        props(4,1:8) = [P,T,Q,H,S,D,C,K];
    else
        props(4,:) = props(5,:);
    end
end
F1b_Nozzle.m

function [V_II,mf,A_s_II_sum,A_II_At] = F1b_Nozzle(power_i,props,...
   FLD,FRCT1,FRCT2)
   % Convergent - Divergent Nozzle Ideal Gas Calculations
   % Use roman numberal to denote turbine stages
   %I - Nozzles Inlet
   %II - Nozzle Outlet
   %III - Blade Outlet
   %Fluid Properties from props
   K_II = props(6,8);
P_I = props(5,1);
P_II = props(6,1);
T_II = props(6,2);
D_II = props(6,6);
[a_II] = refpropm('A', 'T', T_II, 'P', P_II-
   10, FLD,
   {FRCT1 FRCT2});
   %Nozzle Exit
gamma = K_II;
if gamma <= 1
   gamma = 1.4;
end
n_mach_II = ((1/((P_II/P_I)^((gamma-1)/gamma)))^0.5;   %Nozzle exit mach number
V_II = 0.95*(n_mach_II*a_II);                                           %Nozzle exit velocity, assuming 94% efficiency
mf = power_i/((props(5,4)-props(6,4))/1000);                           %Theoretical Work, kW
%Nozzle Throat (Moran Eq 9.52)
A_II_At = (1/n_mach_II)^((2/(gamma+1))((1+((gamma-1)/2)*n_mach_II^2))^(gamma+1)/(2*(gamma-1)));
%Exit Area
A_s_II_sum = mf/(D_II*V_II);
% fprintf('W_theory: %.2f, n_mach_II: %.2f, V_II: %.0f, mf: %.2f, A_s_II: %.6f, A_II_At: %.2f',
   {W_theory, n_mach_II, V_II,mf, A_s_II_sum,A_II_At})
```matlab
F1b_Velocity_Triangle.m

function [ETAT,W_max] = F1b_Velocity_Triangle(V_II, mf,A_s_II_sum,...
    A_II_At,directory_iter,props,mk_plots,FLD,RPM_max,...
    its_a,its_Z,Z_min,Z_max,a_II_min,a_II_max,N_noz_its,N_bld_its)

if mk_plots == 1
    close(figure(10))  % clear omega versus power plot
    close(figure(11))  % clear omega versus power plot
    close(figure(100)) % clear losses plot
    set(0,'defaultAxesFontName', 'Times New Roman')          % Set Axis Font
    set(0,'defaultTextFontName', 'Times New Roman')           % Set Text Font
    font_size = 10;                                           % Set Font size
end

% Use roman numeral to denote turbine stages
% I - Nozzles Inlet
% II - Nozzle Outlet
% III - Blade Outlet

%% Begin Inputs
%%
% All combinations of nozzle angle a__IIr and speed-work Z
% a_II_range = linspace(a_II_min, a_II_max, its_a);   % Nozzle Absolute Angle, rad...0.8 to 1.4
% Z_range = linspace(Z_min, Z_max, its_Z);        % Speed-work parameter range 0.001 to 0.999
%%%
%% Determine Max Power Condition
i3 = 1;
for i0 = 1:size(a_II_range,2)
    for i1 = 1:size(Z_range,2)
        a_III = a_II_range(i0);
        Zi = Z_range(i1);
        % For an impulse machine only: Constant Axial Velocity, W1=W2, U1=U2, and Reaction = 0
        Vx_II(i3) = V_II*cos(a_IIi);     % Based on trig
        Vx_III(i3) = Vx_II(i3);          % Assumption: Constant axial flow velocity for impulse
        Vu_II(i3) = V_II*sin(a_IIi);
        Vu_III(i3) = (Vu_II(i3)*(Zi-0.5))/(Zi+0.5); % From nasa Velocity Diagram paper (Warren J. Whitney), for zero reaction
        W_II(i3) = ((Vx_II(i3)^2)+(Vu_II(i3)^2+U(i3)^2))^0.5; % Based on trig
        W_III(i3) = W_II(i3);  % Assumption: Impulse machine, no reaction thus relative velocity is constant
        % Based on trig
        W_therm = ((mf*(props(5,4)-props(6,4)))/1000); % From nasa Velocity Diagram paper (Warren J. Whitney), eqs 3-3
        W_noloss(i3) = (mf*U(i3)*delta_Vu(i3))/1000;    % Work, kW
        % fprintf('Vu_II: %.2f, Vu_III: %.2f, dVu: %.2f, W_noloss: %.2f, power: %.2f, Z: %.2f, a_II: %.2f\n',
        Reaction = (W_III(i3)^2.0 - W_II(i3)^2.0)/(W_III(i3)^2.0 - W_II(i3)^2.0 + V_II^2.0); % from Figure 8-3 Glassman
        % Losses Model
        % W_real_max, n_mech_local,omega_local,d_r_m_local,N_noz_local, N_bld_local,....
        % N_noz_local,A_s_tt_local,h_s_tt_local,h_r_tt_local,w_s_tt_local,
        % w_r_tt_local,w_s_in_local,L_s_in_local, noz_arc_local,
        % active_arc_local, noz_space_local, pitch_local, R_ss_local,...
```

R_ml_local,solidity_local,chord_local,l_local,active_blds_local,...
active_fraction_local,tip_ratio_local,te_r_local,zw_local,...
h_w_local,gap_r_s_local,tip_clear_local,P_total_local,...
P_clearance_local,P_discf_local,P_sector_local,P_partial_local,...
P_trail_local,P_incidence_local,P_passage_local] = ...
F1b_Losses(V_II,W_II(i3),W_III(i3),U(i3),W_noloss(i3),Reaction,...
b_II(i3),b_III(i3),a_IIi,mf,A_s_II_sum,A_II_At,Vx_II(i3),...
delta_Vui(i3),props,FLD,N_noz_its,N_bld_its,RPM_max);
n_mech_tmp(i3) = n_mech_local;
d_r_m_tmp(i3) = d_r_m_local;
N_bld_tmp(i3) = N_bld_local;
N_noz_tmp(i3) = N_noz_local;
omega_tmp(i3) = omega_local;
te_r_tmp(i3) = te_r_local;
h_s_tt_tmp(i3) = h_s_tt_local;
h_r_tt_tmp(i3) = h_r_tt_local;
zw_tmp(i3) = zw_local;
h_w_tmp(i3) = h_w_local;
R_ss_tmp(i3) = R_ss_local;
gap_r_s_tmp(i3) = gap_r_s_local;
w_s_tt_tmp(i3) = w_s_tt_local;
w_r_tt_tmp(i3) = w_r_tt_local;
tip_clear_tmp(i3) = tip_clear_local;
noz_arc_tmp(i3) = noz_arc_local;
w_s_in_tmp(i3) = w_s_in_local;
L_s_in_tmp(i3) = L_s_in_local;
active_fraction_tmp(i3) = active_fraction_local;
pitch_tmp(i3) = pitch_local;
l_tmp(i3) = l_local;
chord_tmp(i3) = chord_local;
Z_tmp(i3) = Zi;
a_II_tmp(i3) = a_IIi;
%determine max power
W_real_tmp(i3) = W_real_max;
W_noloss_tmp(i3) = W_noloss(i3);
%Save losses
P_total_tmp(i3) = P_total_local;
P_clearance_tmp(i3) = P_clearance_local;
P_discf_tmp(i3) = P_discf_local;
P_sector_tmp(i3) = P_sector_local;
P_partial_tmp(i3) = P_partial_local;
P_trail_tmp(i3) = P_trail_local;
P_incidence_tmp(i3) = P_incidence_local;
P_passage_tmp(i3) = P_passage_local;
ETAT_tmp(i3) = W_real_tmp(i3)/W_therm;
i3 = i3+1;
end
%Get values for max work condition
[ETAT, index] = max(ETAT_tmp);
W_max = W_real_tmp(index);
d_r_m = d_r_m_tmp(index);
N_bld = N_bld_tmp(index);
N_noz = N_noz_tmp(index);
omega = omega_tmp(index);
te_r = te_r_tmp(index);
h_s_tt = h_s_tt_tmp(index);
h_r_tt = h_r_tt_tmp(index);
zw = zw_tmp(index);
h_w = h_w_tmp(index);
R_ss = R_ss_tmp(index);
gap_r_s = gap_r_s_tmp(index);
w_s_tt = w_s_tt_tmp(index);
w_r_tt = w_r_tt_tmp(index);
tip_clear = tip_clear_tmp(index);
noz_arc = noz_arc_tmp(index);
w_s_in = w_s_in_tmp(index);
L_s_in = L_s_in_tmp(index);
active_fraction = active_fraction_tmp(index);
pitch = pitch_tmp(index);
l = l_tmp(index);
chord = chord_tmp(index);

% W_II = W_II(index);
% W_III = W_III(index);
% U = U(index);
% b_II = b_II(index);
% b_III = b_III

Z = Z_tmp(index);

%% Write Turbine Geometry To File

try
    [gamma, mw] = refpropm('KM', 'T', mean(props(5:6,2)), 'H', mean(props(5:6,4), FLD);
catch ME
    gamma = 999;
    mw = 999;
end

Design_Inputs = {'mf', num2str(mf), 'kg/s'; ...
    'T_I', num2str(props(5,2)-273), 'deg C';...
    'P_I', num2str(props(5,1)*1000), 'Pa';...
    'T_II', num2str(props(6,2)-273), 'deg C';...
    'P_II', num2str(props(6,1)*1000), 'Pa';...
    'T_III', num2str(props(6,2)-273), 'deg C';...
    'P_III', num2str(props(6,1)*1000), 'Pa';...
    'gamma', num2str(gamma), '';
    'mw', num2str(mw), 'kg/kmol';...
    'h_w', num2str(h_w), '';
    'N_noz', num2str(N_noz), '';
    'Ma_Coeff', num2str(0.55), '';
    'Mach Selection', 'Mach';'';
    'Machine', 'Impulse';'';
    'its', '45', '';
    'a_II_min', num2str(a_II), 'rad';...
    'a_II_max', num2str(a_II), 'rad';...
Z_max', num2str(Z), '
',
'U', num2str(U(index)), 'm/s',
'V_I', '10', 'm/s',
'V_II', num2str(V_II), 'm/s',
'b_II', num2str(b_II(index)), 'rad',
'a_III', num2str(a_III(index)), 'rad',
'w_s_in', num2str(w_s_in), 'm',
'L_s_in', num2str(L_s_in), 'm',
'w_s_tt', num2str(w_s_tt), 'm',
'h_s_tt', num2str(h_s_tt), 'm',
'd_s_tp', num2str(d_s_tp), 'm',
'd_s_r', num2str(d_s_r), 'm',
'noz.arc', num2str(noz.arc), 'm',
'Kst', num2str(Kst), '',
'd_r_m', num2str(d_r_m), 'm',
'd_r_r', num2str(d_r_r), 'm',
'pitch', num2str(pitch), 'm',
'chord', num2str(chord), 'm',
'N_bld', num2str(N_bld), '',
'te_r', num2str(te_r), 'm',
'R_ss', num2str(R_ss), 'm',
'zw', num2str(zw), '',
'gap', num2str(gap), 'm',
'tip_clear', num2str(tip_clear), 'm',
'Xli', num2str(Xli), 'm',
'Xlo', num2str(Xlo), 'm',
'Xsl', num2str(Xsl), 'm',
'Xst', num2str(Xst), 'm',
'Kro', num2str(Kro), 'm',

P_total = P_total_tmp(index);
P_clearance = P_clearance_tmp(index);
P_discf = P_discf_tmp(index);
P_sector = P_sector_tmp(index);
P_partial = P_partial_tmp(index);
P_trail = P_trail_tmp(index);
P_passage = P_passage_tmp(index);
P_incidence = P_incidence_tmp(index);

if mk_plots == 1 & W_max > 0

"Write Design Inputs File"
filename = sprintf('%s\Design_Inputs.csv', directory_iter);
 fid = fopen(filename, 'w');
for i = 1:size(Design_Inputs,1)
 for j = 1: size(Design_Inputs,2)
 fprintf(fid, '%s', Design_Inputs{i,j});
if j == size(Design_Inputs,2)
 fprintf(fid,'\n');
else
 fprintf(fid,'\n');
end

186
% Plot Feasible Turbines: Work versus Omega
s_d_r_m_min = min(d_r_m_tmp);
s_d_r_m_max = max(d_r_m_tmp);
s_d_r_m_range = s_d_r_m_max - s_d_r_m_min;

% Set up Omega Color Scale
omega_min = min(omega_tmp);
omega_max = max(omega_tmp);
omega_range = omega_max - omega_min;
omega_colors = colormap('jet');
figure(11)
shading interp; camlight(60,20,'local'); lighting gouraud; camlight left; camlight right;
[x,y,z] = sphere(20);
for i = 1:size(omega_tmp,2)
    if omega_tmp(i) > 0
        if s_d_r_m_range > 0
            s_d_r_m = 10*(d_r_m_tmp(i)/s_d_r_m_range);
        else
            s_d_r_m = 10;
        end
    figure(11)
    d_i = s_d_r_m/200;
    x_i = d_i*x+a_II_tmp(i);
    y_i = d_i*y+Z_tmp(i);
    z_i = d_i*z+W_real_tmp(i);
    omega_color_index = floor(((omega_tmp(i) - omega_min)/omega_range)*(size(omega_colors,1)-1)+1);
    omega_color_i = omega_colors(omega_color_index,:);
    mesh(x_i,y_i,z_i,'FaceColor',omega_color_i,'EdgeColor',omega_color_i,'FaceLighting','phong'); hold on
    sphere centered at origin
    figure(10)
    if W_real_tmp(i) == W_max
        plot(omega_tmp(i), W_real_tmp(i), 'o', 'MarkerFaceColor',omega_color_i,'MarkerEdgeColor', 'k','LineWidth', 2, 'MarkerSize',s_d_r_m); hold on;
    else
        plot(omega_tmp(i), W_real_tmp(i), 'o', 'MarkerFaceColor',omega_color_i,'MarkerEdgeColor',omega_color_i, 'MarkerSize',s_d_r_m); hold on;
    end
end

% Format W vs Omega
figure(10)
plot_title = sprintf('Feasible Turbines (Turbine Shaft Power vs \omega, marker size indicative of D_{r,m})');
title(plot_title)
xlabel('\omega (rad/s)')
ylabel('Turbine Shaft Power (kW)')
anno_xywh = [0.70 0.15 0.215 0.400];
anno_text = sprintf('W_m_{ax}: %.2f kW
D_{r,m}: %.2f m\omega: %.0f rad/s
Z: %.2f na_{I,I}: %.2f na_{II}: %.0f na_{b,l,d}: %.0f na_{o,z}: %.0f \eta_{ETAT}: %.2f \eta_{Admission}: %.2f
{W_{max}, d_{r,m}, omega, Z, a_{II}, N_{bld}, N_{noz}, ETAT, active_fraction});
annotation('textbox',anno_xywh,'String',anno_text,'FontSize',font_size,'EdgeColor','w', 'LineStyle', 'none');
% Save W vs Omega plot
filename10 = sprintf('%s\Feasible-Speed-Work', directory_iter);
saveas(gcf, filename10,'fig');

% Format W vs alpha and Z plot
figure(11)
plot_title = sprintf('Feasible Turbines (Turbine Shaft Power vs \alpha_I vs Z)\nD_r \_m Qualitatively Respresented By Marker Size');
axis([0 1.5 0 1 0 1.1*max(W_real_tmp)]);

title(plot_title)
xlabel('\alpha_I (rad/s)')
ylabel('Z')
zlabel('Turbine Shaft Power (kW)')
anno_xywh = [0.15 0.35 0.215 0.400];
anno_text = sprintf('E_m_a_x: %.2f kW\nD_r \_m: %.2f m\n \omega: %.0f rad/s\nZ: %.2f
na_I_I: %.2f
N_bld:
N_n_o_z: %.0f
ETAT: %.2f
Admission: %.2f',
{W_max, d_r_m, omega, Z, a_II, N_bld, N_noz, ETAT, active_fraction});
annotation('textbox',anno_xywh,'String',anno_text,'FontSize',font_size,'EdgeColor','w','LineStyle', 'none');

% Save W vs alpha and Z plot
filename11 = sprintf('%s
Feasible-Alpha-Z', directory_iter);
saveas(gcf, filename11,'fig');

figure(100)
% Plot losses for all results
 [~,I] = sort(W_real_tmp); % Sort by W_real
W_real_bar = W_real_tmp(:,I);
omega_bar = omega_tmp(:,I);
% ETAT_bar = ETAT_tmp(:,I);
% P_total_bar = P_total_tmp(:,I);
% P_discf_bar = P_discf_tmp(:,I);
% P_sector_bar = P_sector_tmp(:,I);
% P_partial_bar = P_partial_tmp(:,I);
% P_trail_bar = P_trail_tmp(:,I);
% P_incidence_bar = P_incidence_tmp(:,I);
% P_passage_bar = P_passage_tmp(:,I);

n_ideal_bar = (W_noloss(:,I)/W_therm)*100;
ETAT_bar = ETAT_tmp(:,I);
P_total_bar = (W_real_bar:P_total_tmp(:,I))*100;
P_clearance_bar = (W_real_bar:P_clearance_tmp(:,I))*100;
P_discf_bar = (W_real_bar:P_discf_tmp(:,I))*100;
P_sector_bar = (W_real_bar:P_sector_tmp(:,I))*100;
P_partial_bar = (W_real_bar:P_partial_tmp(:,I))*100;
P_trail_bar = (W_real_bar:P_trail_tmp(:,I))*100;
P_incidence_bar = (W_real_bar:P_incidence_tmp(:,I))*100;
P_passage_bar = (W_real_bar:P_passage_tmp(:,I))*100;

y_value = {n_ideal_bar; ETAT_bar*100; P_total_bar; P_clearance_bar; P_discf_bar; P_sector_bar;...
P_partial_bar; P_trail_bar; P_incidence_bar; P_passage_bar;...}

h = bar3(y_value,1.0,'detached');
zlabel_string = sprintf('Loss as Percent of Work\nETAT, \eta_i_d_e_a_l (\%)');

plot_title = sprintf('Individual Turbine Power Losses and ETAT versus Turbine Shaft Power\n(All Feasible Turbines For Cycle)');
title(plot_title)
Power_Labels = '{strcat(Indents,'\eta_i_d_e_a_l');

188
strcat(Indents,'Incidence'); strcat(Indents,'Passage'});

% Power_Labels = {'\eta_ideal'; 'ETAT'; 'Total'; \eta_incidence'; \eta_passage'};

set(gca,'YTickLabel',Power_Labels,'Fontsize',font_size)
rotateyticklabel(gca,90);

W_real_min = min(ceil(W_real_bar));
W_real_max = max(ceil(W_real_bar));
W_real_ran = W_real_max - W_real_min;
W_real_inc = ceil(W_real_ran/8);

W_real_Tick_Labels = W_real_min:W_real_inc:W_real_max;
W_real_Tick = 1:length(h)/size(W_real_Tick_Labels,2):length(h);

shading faceted; camlight(160,20,'local'); lighting gouraud; camlight left; camlight right;

for i = 1:length(h)
    omega_color_index = floor(((omega_bar(i) - omega_min)/omega_range)*(size(omega_colors,1)-1)+1);
    omega_color_i = omega_colors(omega_color_index,:);
    set(h(i), 'facecolor', omega_color_i);
end
set(gca,'XTick', W_real_Tick,'XTickLabel',W_real_Tick_Labels)
view(79,9);
filename100 = sprintf('%s\Losses_All', directory_iter);
saveas(gcf, filename100,'fig');

figure(101)

% Plot losses for Max Result
{i_Wmax} = max(W_real_bar);
y_value = {P_total_bar(i_Wmax); P_clearance_bar(i_Wmax); P_discf_bar(i_Wmax); P_sector_bar(i_Wmax);... P_partial_bar(i_Wmax); P_trail_bar(i_Wmax); P_incidence_bar(i_Wmax); P_passage_bar(i_Wmax)};

omega_color_index = floor(((omega_bar(i_Wmax) - omega_min)/omega_range)*(size(omega_colors,1)-1)+1);
omega_color_i = omega_colors(omega_color_index,:);

bar(y_value,1.0,'facecolor', omega_color_i);
ylabel('Loss as Percent of Work (%)');
plot_title = sprintf('Individual Turbine Power Losses\n(Maximum Turbine Power Case)');
title(plot_title)

Power_Labels = {'Total'; 'Clearance'; 'Windage'; 'Sector'; 'Pumping'; 'Trail'; 'Incidence'; 'Passage'};
set(gca,'XTickLabel',Power_Labels)
rotateyticklabel(gca,30);
grid on
filename101 = sprintf('%s\Losses_Max', directory_iter);
saveas(gcf, filename101,'fig');

figure(102)

% Plot ETAT for all results
bar3(ETAT_bar*100,1.0);
zlabel('ETAT (%)');
ylabel('Turbine Shaft Power (kW)');
plot_title = sprintf('Turbine ETAT versus Turbine Shaft Power\n(All Feasible Turbines For Cycle)');
title(plot_title)
W_real_min = min(ceil(W_real_bar));
W_real_max = max(ceil(W_real_bar));
W_real_ran = W_real_max - W_real_min;
W_real_inc = ceil(W_real_ran/10);
W_real_Tick_Labels = W_real_min:W_real_inc:W_real_max;
W_real_Tick = 1:length(ETAT_bar)/10:length(ETAT_bar);
set(gca,'YTick', W_real_Tick,'YTickLabel',W_real_Tick_Labels)
view(-90,0);

filename102 = sprintf('%s\ETAT_All', directory_iter);
saveas(gcf, filename102,'fig');

%Plot Velocity Triangle
F1b_Plot_VT(a_II, Z, b_II(index), a_III(index), b_III(index), V_II, Vx_II(index), Vu_II(index), W_II, V_III(index),
Vx_III(index), Vu_III(index), W_max, d_r_m, U(index), ETAT, directory_iter)
end
F1b_Losses.m

function \{W_real_max,n_mech_local,omega_local,d_r_m_local,N_bld_local,...
N_noz_local,A_s_tt_local,h_s_tt_local,h_r_tt_local,...
noz_arc_local,active_arc_local,nox_space_local,pitch_local,...
R_ss_local,R_ml_local, solidity_local, chord_local, Llocal,...
active_blds_local,active_fraction_local,tip_ratio_local, te_r,...
zw_h,w_gap_r_s, tip_clear, P_total_local, P_clearance_local,...
P_discf_local, P_sector_local, P_partial_local, P_trail_local,...
P_incidence_local, P_passage_local\} = F1b_Losses(V_II,...
W_II,W_III,U,W_noloss, Reaction, b_II, b_III, a_IIi, mf,...
A_s_II_sum, A_II_At, Vx_II, delta_Vu, props, FLD, N_noz_its, N_bld_its, RPM_max)

P_total = \{\}; P_clearance = \{\}; P_discf = \{\}; P_sector = \{\};
P_partial = \{\}; P_trail = \{\}; P_incidence = \{\}; P_passage = \{\};
geom_check = \{\};

% Use roman numeral to denote turbine stages
% I - Nozzles Inlet
% II - Nozzle Outlet
% III - Blade Outlet
% Loss calculations taken from
% "Turbine Design and Application", Glassman, A.J., NASA (SP-290), 1994
% "Introduction to Turbomachinery", Japikse, D., Baines, N.C., 1994
% User Defaults
% Iterator Inputs
% d_r_m_max = max(0.1,0.3*log(W_noloss));
% d_r_m_max = 3;
% d_r_m_range = 0.05:0.01:d_r_m_max;      %Range of meanline diameters,m
% User Defaults Geometry Ranges
% te_r = 0.001; %Blade leading edge thickness, m
% te_s = 0.001; %Noz trailing edge thickness, m
% tip_clear = 0.00003; %Minimum achievable rotor-housing Clearance, m
% w_r_tt_min = 0.0020; %Minimum blade throat, m
% w_s_tt_min = 0.0020; %Minimum nozzle throat, m
% h_r_tt_min = 0.0020; %Minimum blade height, m
% h_s_tt_min = 0.0020; %Minimum nozzle height, m
% zw = 8.00; % Zweifel coefficient, value based on pp 116 in chapter 4 of NASA report (Glassman)
% gap_r_s = 0.0015; %Gap between nozzle and rotor, m
% h_w = 1.00; %Nozzle h/w ratio
% endurance_limit = 620; %Endurance limit of Inconel 718 at room temperature for 10^8 cycles,
% U_max = 520; %Maximum Tip Speed, m/s (Japikse and Baines, 3-5)
% Loss Coefficients
% X_clearance = 0.79; X_discf = 0.70; X_sector = 1.75; X_partial = 1.25; X_trail = 0.70;
% X_incidence = 0.30; X_passage = 0.35; E_discf = 2.90; E_partial = 2.97;

i_loss = 0;
% fprintf('----------------------------------------
% fprintf('---Run Loss For U: %.2f, a1: %.2f
% %Blade Diameter Iterator
% for i_d_r_m = 1:size(d_r_m_range,2)
% d_r_m = d_r_m_range(i_d_r_m);
% C_r = d_r_m*pi; %Blade tip circumference, m
% for i_d_r_m = 1:size(d_r_m_range,2)
N_bld_max = floor(pi() * d_r_m / (w_r_tt_min + te_r)); % High number of blades
N_bld_range = ceil(logspace(0, log10(N_bld_max), N_bld_its));
N_bld_range = unique(N_bld_range);
for i_N_bld = 1:size(N_bld_range, 2)
    % Number of Nozzles Iterator, a function of d_r_m_range(i_d_r_m)
    noz_arc_min = w_s_tt_min / sin((pi/2) - a_IIi); % Arc length of nozzle exit, m
    N_noz_max = floor(C_r / noz_arc_min);
    N_noz_range = ceil(logspace(0, log10(N_noz_max), N_noz_its));
    N_noz_range = unique(N_noz_range);
    for i_N_noz = 1:size(N_noz_range, 2)
        % Geometry Calculations
        N_bld = N_bld_range(i_N_bld);               % Number of blades
        N_noz = N_noz_range(i_N_noz);               % Number of nozzles
        pitch = C_r / N_bld;                          % Blade space or pitch, m
        w_r_tt = (pitch - te_r) * sin((pi/2) - b_II);   % Distance between suction to pressure surface of blade, m
        A_s_tt = (A_s_II_sum/A_II_At)/N_noz;        % Nozzle throat area per nozzle, m^2
        A_s_II = A_s_II_sum/N_noz;                  % Nozzle Exit Area Per Nozzle, m^2
        h_s_tt = (A_s_tt/h_w)^0.5;                  % Nozzle throat per nozzle, m
        L_s_in = 1*w_s_tt;                          % Nozzle Inlet Length, m
        noz_arc = w_s_tt / sin((pi/2) - a_IIi);         % Arc length of nozzle exit, m
        active_arc = noz_arc * N_noz; % Active circumference of the nozzle, m
        noz_space = C_r / N_noz;                      % Nozzle space or pitch, m
        omega = 2*U/(d_r_m);                        % Rotor angular velocity, rad/s
        RPM = omega * (60/(2*pi));                   % Rotor RPM
        solidity = (2.0/zw) * (cos(-b_III)/cos(b_II)) * sin(b_II + b_III);   % Blade Solidity, Glassman 4-14
        chord = solidity * pitch; % Blade Chord, Glassman 4-3
        R_ml = ((chord/2) * cos((pi/2) - b_II)) + w_r_tt; % Blade meanline radius of curvature, m
        active_blds = ceil(noz_arc / pitch) * N_noz; % Number of active blades, m
        active_fraction = active_arc / C_r; % Active fraction of stator-exit area
        tip_ratio = tip_clear / h_r_tt; % Tip clearance as a fraction of blade height
        fprintf('N_noz: %.0f, N_bld: %.0f, d_r_m: %.2f, C_r: %.2f
', {N_noz, N_bld, d_r_m, C_r})
    end
end

% Strength Calculations:

% Strength Checks: Shaft Strength
Torque = W_noloss/omega; % Shaft torque, N-m
d_shaft = (d_r_m - (2*h_r_tt))/2; % Shaft diameter, m
Ip = pi*((d_shaft^4)/32); % Shaft polar moment of inertia, m^4
shaft_shear = (1/1000000)*((Torque*(d_shaft^2)/Ip); % Shaft shear stress at surface, MPa
shaft_fs = endurance_limit/shaft_shear; % Shaft factor of safety

% Strength Checks: Blade Strength
R_ss = R_ml - w_r_tt/2; % Blade suction surface radius of curvature, m
R_ps = R_ml + w_r_tt/2; % Blade pressure surface radius of curvature, m
bld_thickness = R_ss + pitch - R_ps % Blade thickness at throat, m
bld_force = m*l*delta_Vu/active_blds; % Blade Force Per Blade, N
bld_moment = bld_force*h_r_tt/2; % Blade bending moment, N-m
bld_stress = (bld_moment/((chord/2)*(bld_thickness^2)))/(1/6); % Blade stress, MPa
bld_fs = endurance_limit/bld_stress; % Blade strength factor of safety

192
% Geometry Checks:

geom_check(1,1) = pitch <= te_r;  % Geometry Error Checks: Blade space is less than te_r
geom_check(1,2) = w_r_tt < w_r_tt_min;  % Geometry Error Checks: Blade throat is less than min
geom_check(1,3) = active_fraction > 1;  % Geometry Error Checks: Active fraction is greater than 1
geom_check(1,4) = w_s_tt < w_s_tt_min;  % Geometry Error Checks: Nozzle throat is less than min
geom_check(1,5) = h_s_tt < h_s_tt_min;  % Geometry Error Checks: Nozzle height is less than min
geom_check(1,6) = active_arc >= C_r;  % Geometry Error Check: Nozzle Active Arc greater than C_r
geom_check(1,7) = noz_space <= te_s;  % Geometry Error Checks: Nozzle space is less than te_s
geom_check(1,8) = h_r_tt <= h_r_tt_min;  % Geometry Error Checks: Blade height is less than min
geom_check(1,9) = tip_ratio > 0.4;  % Geometry Error Checks: Tip Ratio is greater than max allowed
geom_check(1,10) = shaft_fs < 2;  % Strength Checks: Shaft Strength
geom_check(1,11) = bld_fs < 2;  % Strength Checks: Blade Strength
geom_check(1,12) = U > U_max;  % Strength Check: U is greater than U_max
geom_check(1,13) = RPM > RPM_max;  % Operation Error Checks: RPM is greater than max allowed

if max(geom_check) ~= 1

%% Begin Loss Model

i_loss = i_loss + 1;
% Tip Clearance Losses (Glassman, 1994 Figure 8-3)
% K_clearance = Ax^2 + Bx + C
% Matrix of Coefficients, m_clear = [A B C tip_ratio]
m_clear = [1.000, 1.000, 1.000, 0.000];
-0.023, -0.002, 0.983, 0.01;
-0.046, -0.005, 0.966, 0.02;
-0.069, -0.008, 0.949, 0.03;
-0.092, -0.011, 0.932, 0.04;
-0.115, -0.014, 0.916, 0.05;
-0.138, -0.017, 0.889, 0.06;
-0.161, -0.020, 0.882, 0.07;
-0.184, -0.023, 0.865, 0.08;
-0.207, -0.026, 0.849, 0.09;
-0.230, -0.029, 0.832, 0.10;
-0.345, -0.044, 0.748, 0.15;
-0.460, -0.058, 0.644, 0.20;
-0.575, -0.073, 0.580, 0.25;
-0.690, -0.088, 0.496, 0.30;
-0.805, -0.102, 0.412, 0.35;
-0.920, -0.117, 0.328, 0.40];

% Bilinear Interpolation Procedure to Calculate K_clearance

% Bilinear Interpolation Procedure to Calculate K_clearance

tip_ratio_index_lo = find(m_clear(:,4) <= tip_ratio,1,'last');
tip_ratio_index_hi = tip_ratio_index_lo + 1;
tip_ratio_lo = m_clear(tip_ratio_index_lo,4);
tip_ratio_hi = m_clear(tip_ratio_index_hi,4);
A_lo = m_clear(tip_ratio_index_lo,1);
B_lo = m_clear(tip_ratio_index_lo,2);
C_lo = m_clear(tip_ratio_index_lo,3);
K_clearance_lo = A_lo*Reaction^2.0 + B_lo*Reaction + C_lo;
A_hi = m_clear(tip_ratio_index_hi,1);
B_hi = m_clear(tip_ratio_index_hi,2);
C_hi = m_clear(tip_ratio_index_hi,3);
K_clearance_hi = A_hi*Reaction^2.0 + B_hi*Reaction + C_hi;
K_clearance = ((tip_ratio_hi - tip_ratio)/(tip_ratio_hi - tip_ratio_lo))*K_clearance_hi ... 
+ ((tip_ratio - tip_ratio_lo)/(tip_ratio_hi - tip_ratio_lo))*K_clearance_lo;
P clearance(i_loss) = X clearance*((1.0 - K_clearance)*W_noloss);  %kW
mf_leak = mf*(P clearance(i_loss)/(mf*props(5,4) - props(6,4)));   %The mass flow in the clearance gap

%Disc Friction Losses (Roelke 1994 and Augnier 2006)################################
%based on no-through flow, Roelke:  eqs 8-9, 8-10, 8-12, 8-14, 8-16
%based on no-through flow, Augnier:  eqs 4-109 to 4-114
if omega > 0.0
  %Windage Disc Friction Losses
  s = gap_r_s;
a = d_r_m/2.0;
s_a = s/a;
D2 = props(6,6);
try
  v2 = refpropm('V', 'D', D2, 'H', props(6,4), FLD);
catch ME
  v2 = refpropm('V', 'T', props(6,2), 'Q', 0, 'Water');
end
Re_disc = (omega*(a^2.0)*D2)/v2;                              %eq 8-10
Cmo_I = (2.0*pi)/(s_a*Re_disc);                              %eq 8-9, Flow Regime I: Laminar, Small Clearance
Cmo_II = (3.70*(s_a^0.10))/Re_disc^0.5;                      %eq 8-12, Flow Regime II: Laminar, Large Clearance
Cmo_III = 0.080 /((s_a^(1.0/6.0))*Re_disc^0.25));       %eq 8-14, Flow Regime III: Turbulent, Small Clearance
Cmo_IV = (0.1020*(s_a^0.10))/(Re_disc^0.2);                  %eq 8-16, Flow Regime IV: Turbulent, Large Clearance
Cmo = max({Cmo_I,Cmo_II,Cmo_III, Cmo_IV});
P_windage = (X_discf*(Cmo*D2*(omega^E_discf)*(a^5.0)/2.0))/1000;     %eq 8-7, Roelke, R.J.W, kW

%Clearance Gap Windage Losses (Augnier 2006, eq 4-119 to 4-123)
if Re_gap < 2000.0
  c_f = 16.0/Re_gap;
else
  c_f = 0.0791/Re_gap^0.25;
end
P_gap = (pi*D2*c_f(a^4.0)*(omega^3.0)*l/4.0)/1000;            %Power lost to shear stresses in the clearance gap, eq. 4-123
P_discf(i_loss) = (P_windage + P_gap);                      %Combined discf loss, kW
else
  P_discf(i_loss) = 0.0;
end

%Partial Admission Sector Losses (Glassman, 1994)
Ks = (1-(pitch/(3*active_arc)));
%Rotord velocity coefficient for sector loss, eq 8-24 (Roelke/Varma)
Ks = (1-(E_sector*active_fraction*pitch/(active_arc)));
%Rotord velocity coefficient for sector loss, eq 8-24 (Roelke modified version a)
\[ K_s = (1 - \text{pitch}/3*\text{active}_\text{arc})*(E_\text{sector}^{\text{active}_\text{fraction}}); \]  
\[ \% \text{Rotor velocity coefficient for sector loss, eq 8-24 (Roelke modified version b)} \]

\[ K_w = \frac{\text{abs}(W_\text{III})}{\text{abs}(W_\text{II})}; \]  
\[ \% \text{Rotor relative-velocity ratio for, W2/W1} \]

\[ K_\text{sector} = U_\text{sector}^{\text{chord}}*U^{0.95}*/(d_{r_m}*\text{active}_\text{fraction}); \]  
\[ \% \text{Aungier, 4-105} \]

\[ P_\text{sector}(i_\text{loss}) = X_\text{sector}(W_\text{noloss} - (K_\text{sector}*(mf-mf_\text{leak})/1000)); \]  
\[ \% \text{Partial Admission Pumping Losses} \]

\[ P_\text{partial}(i_\text{loss}) = (X_\text{partial}*3.63*D^2(U^{\text{E}_\text{partial}})*h_{r_t}^{1.5})/1000; \]  
\[ \% \text{eq 8-23, Roelke} \]

\[ x = \text{te}_r/w_{r_{tt}}; \]
\[ K_\text{trail}_\text{Imp} = 0.275*x^2.0 + 0.080*x; \]  
\[ K_\text{trail}_\text{Axii} = 0.478*x^2.0 + 0.158*x; \]
\[ K_\text{trail} = K_\text{trail}_\text{Imp} + ((b_{II}/b_{III})^2)*(K_\text{trail}_\text{Axii} - K_\text{trail}_\text{Imp}); \]
\[ P_\text{trail}(i_\text{loss}) = (X_\text{trail}*0.5*(mf-mf_\text{leak})*K_\text{trail}*(W_{III}^2))/1000; \]

\[ \% \text{Profile Losses of trailing edge (Japikse and Baines figure 6.62 and eq 6.52)} \]

\[ \text{incidence} = 0; \]
\[ \text{if incidence} < 0.0 \]
\[ n = 2.0; \]
\[ \text{else} \]
\[ n = 3.0; \]
\[ \text{end} \]
\[ \text{incidence}_\text{opt} = 6.0*pi/180.0; \]
\[ \% \text{optimum incidence from Roelke, page 245 chpt 8} \]
\[ K_\text{incidence} = ((W_{II}^2)/(2.0)*(1.0-\cos(\text{incidence}_\text{incidence}_\text{opt})^n)); \]  
\[ \% \text{eq 8-34 in J/kg} \]
\[ P_\text{incidence}(i_\text{loss}) = (X_\text{incidence}*((mf-mf_\text{leak})*K_\text{incidence}))/1000; \]

\[ \% \text{Profile Losses and Secondary Losses} \]
\[ \% \text{Profile Secondary Losses (baines 6.64)} \]
\[ f_{dc} = 0.0334; \]  
\[ \% \text{Correlation for tip clearance to chord} \]
\[ b_{II} = b_{II}; \]  
\[ \% \text{Blade inlet metal angle in tangential reference} \]
\[ b_{1s} = \pi/2 - b_{II}; \]  
\[ \% \text{Gas relative inlet angle in tangential reference} \]
\[ b_{2s} = \pi/2 - b_{III}; \]  
\[ \% \text{Gas relative outlet angle in tangential reference} \]
\[ b_{mean} = \pi/2 - \arctan((1/(tan(b_{1s} + b_{2s}))/2); \]  
\[ \% \text{mean flow angle, Aungier 4-79} \]
\[ \text{sound}_2 = \text{refpropm}(\text{A},\text{T},\text{props}(6,2),\text{P},\text{props}(6,1),\text{FLD}); \]  
\[ \% \text{Sound speed at blade exit, m/s} \]
\[ \text{F}_{ar} = 0.5*(2*(\text{chord}/h_{r_{tt}}))^{0.7}; \]
\[ \% \text{Aspect Ratio Correction, Aungier 4-81} \]
\[ C_L = 2*((1/tan(b_{1s}))/(-1/tan(b_{2s}))); \]
\[ \% \text{Lift coefficient, Aungier 4-77} \]
\[ Z = ((C_L^2)/(2*(\sin(b_{2s})^2)/\sin(b_{mean})^3)); \]
\[ \% \text{Ainley loading factor, 4-78} \]
\[ \text{Re}_c = D^2*W_{III}^2/v_2; \]  
\[ \% \text{Blade chord Reynolds number, Aungier 4-72} \]
\[ \text{Kre} = (\log(10(500000))/\log(10(\text{Re}_c)))^{2.58}; \]  
\[ \% \text{Reynolds correction for turbulent, Aungier 4-74} \]
\[ M1 = W_{II}/\text{sound}_2; \]  
\[ \% \text{Blade inlet relative mach number} \]
\[ M2 = W_{III}/\text{sound}_2; \]  
\[ \% \text{Blade outlet relative mach number} \]
\[ M1_{ave} = (M1-0.566 + \text{abs}(0.566-M1))/2; \]
\[ \% \text{Blade inlet modified mach number, Aungier 4-59} \]
\[ M2_{ave} = (M2+1.000 - \text{abs}(M1-1.000))/2; \]
\[ \% \text{Blade outlet modified may number, Aungier 4-60} \]
\[ X = (2.0*M1_{ave})/(M1_{ave}+M2_{ave}+\text{abs}(M2_{ave}-M1_{ave})); \]
\[ \% \text{Aungier 4-61} \]
K1 = 1.0 - 0.625*(M2_ave - 0.2 + abs(M2_ave - 0.2)); %Aungier 4-62
Kp = 1.0 - (1 - K1)*X^2.0; %Compressibility correction, Aungier 4-63

K_secondary_pre = f_dc*F_ar*Z*sin(b2s)/sin(B_II); %Prelim Secondary Loss Coefficient, Aungier 4-80
Ks = 1 - (1 - Kp)*(chord/h_r_tt)^2/(1 + (chord/h_r_tt)^2); %Modified compressibility factor, Aungier 4-83
(Note that this should be modified if using blades who's axial chord project is different from the chord
K_secondary = K_re*Ks*sqrt((K_secondary_pre^2)/(1 + 7.5*K_secondary_pre^2)); %Secondary loss coefficient, Aungier 4-82
P_secondary = (0.5*(W_III^2)*K_secondary*(mf - mf_leak))/1000; %Secondary power loss

%Profile Passage Losses
[K_profile] = F1b_Profile(w_r_tt, h_r_tt, W_II, W_III,...
R_ml, l, D2, v2, active_blds);
P_profile = K_profile*(mf-mf_leak);
P_passage(i_loss) = X_passage*(P_profile+P_secondary)/1000;
%
fprintf('P_pass: %.0f, P_prof: %.0f, P_secd: %.0f', [P_passage(i_loss), P_profile, P_secondary])

%Sum of Losses in kW
P_total(i_loss) = P_clearance(i_loss) + P_discf(i_loss) + P_sector(i_loss) ...
+ P_partial(i_loss) + P_trail(i_loss) + P_incidence(i_loss) + P_passage(i_loss);
%
fprintf('%.0f %.0f %.0f %.0f %.0f %.0f %.0f
', [P_total(i_loss), P_clearance(i_loss), P_discf(i_loss), P_sector(i_loss), P_partial(i_loss), P_trail(i_loss), P_incidence(i_loss), P_passage(i_loss)]);

%Error Check: Ensure that losses are not greater than work
if P_total(i_loss) < 0 || P_total(i_loss) > W_noloss
P_total(i_loss) = W_noloss; %set losses equal to work
%
fprintf('Error: Set P_total to equal W_ideal\n')
end
%
fprintf('W_noloss: %.2f, P_total: %.2fn', [W_noloss, P_total(i_loss)])
W_real_i(i_loss) = W_noloss - P_total(i_loss);
n_mech_i(i_loss) = W_real_i(i_loss)/W_noloss;

%Save Geometry Values For Loss Solutions
omega_save(i_loss) = omega;
d_r_m_save(i_loss) = d_r_m;
N_bld_save(i_loss) = N_bld;
N_noz_save(i_loss) = N_noz;
A_s_tt_save(i_loss) = A_s_tt;
h_s_tt_save(i_loss) = h_s_tt;
h_r_tt_save(i_loss) = h_r_tt;
w_s_tt_save(i_loss) = w_s_tt;
w_r_tt_save(i_loss) = w_r_tt;
w_s_in_save(i_loss) = w_s_in;
L_s_in_save(i_loss) = L_s_in;
noz_arc_save(i_loss) = noz_arc;
active_arc_save(i_loss) = active_arc;
noz_space_save(i_loss) = noz_space;
pitch_save(i_loss) = pitch;
R_ss_save(i_loss) = R_ss;
R_ml_save(i_loss) = R_ml;
solidity_save(i_loss) = solidity;
chord_save(i_loss) = chord;
l_save(i_loss) = l;
active_blds_save(i_loss) = active_blds;
active_fraction_save(i_loss) = active_fraction;
tip_ratio_save(i_loss) = tip_ratio;
end
end
end
try
    \([W_{\text{real, max}}, \text{index}] = \max(W_{\text{real, i}});\)
    n_mech_local = n_mech_i(index);

    omega_local = omega_save(index);
    d_r_m_local = d_r_m_save(index);
    N_bld_local = N_bld_save(index);
    N_noz_local = N_noz_save(index);
    A_s_tt_local = A_s_tt_save(index);
    h_s_tt_local = h_s_tt_save(index);
    h_r_tt_local = h_r_tt_save(index);
    w_s_tt_local = w_s_tt_save(index);
    w_r_tt_local = w_r_tt_save(index);
    w_s_in_local = w_s_in_save(index);
    L_s_in_local = L_s_in_save(index);
    noz_arc_local = noz_arc_save(index);
    noz_space_local = noz_space_save(index);
    pitch_local = pitch_save(index);
    R_ss_local = R_ss_save(index);
    R_ml_local = R_ml_save(index);
    solidity_local = solidity_save(index);
    chord_local = chord_save(index);
    l_local = l_save(index);
    active_blds_local = active_blds_save(index);
    active_fraction_local = active_fraction_save(index);
    tip_ratio_local = tip_ratio_save(index);

    P_total_local = P_total(index);
    P_clearance_local = P_clearance(index);
    P_discf_local = P_discf(index);
    P_sector_local = P_sector(index);
    P_partial_local = P_partial(index);
    P_trail_local = P_trail(index);
    P_incidence_local = P_incidence(index);
    P_passage_local = P_passage(index);

    catch ME
        fprintf('No Solutionn')
        \([W_{\text{real, max}}, \text{n_mech_local, omega_local, d_r_m_local, N_bld_local, N_noz_local, A_s_tt_local,...}\)
        h_s_tt_local, h_r_tt_local, w_s_tt_local, w_r_tt_local, w_s_in_local, L_s_in_local, noz_arc_local, ...
        noz_space_local, pitch_local, R_ss_local, R_ml_local, ...
        solidity_local, chord_local, l_local, active_blds_local, ...
        active_fraction_local, tip_ratio_local, te_rzw, b_w, gap_r_s, tip_clear] = deal(0);

        \([P_{\text{total, local}}, P_{\text{clearance, local}}, P_{\text{discf, local}}, P_{\text{sector, local}}, P_{\text{partial, local}},...
        P_{\text{trail, local}}, P_{\text{incidence, local}}, P_{\text{passage, local}] = deal(W_{\text{noloss}});\)
    end
function \{ Qhx_h, Qhx_c, Thxo, Tcxo, wf_reg_h, wf_reg_c, check, error \} = F1b_Pinch_Rhx(mf, Rhx, ...
FLD, FRCT1, FRCT2, props, mk_plots, plot_colors, mk_phe, directory_iter, power)

clearvars H6xo Q_x q_x Hhx Hhx Thx Thx Hcx Hcx Tcx Tcx dT
error = 0;
check = 0;
N = 11;  % HX segments
Thxi = props(8,2);
Hhxi = props(8,4);
Tcxi = props(10,2);
Hcxi = props(10,4);
Thxo = Tcxi+(Rhx/2);
dT_pinch = 2;
for i0 = 1:1000
for i1 = 1:N
Hhxo = refpropm('H', 'T', Thxo, 'P', props(8,1), FLD, {...
FRCT1 FRCT2});
Q_x = mf*(Hhxi - Hhxo);
q_x = Q_x/(N-1);
Hhx(i1) = Hhxi - ((i1-1)*q_x)/mf;
Thx(i1) = refpropm('T', 'P', props(8,1), 'H', Hhx(i1), FLD, {...
FRCT1 FRCT2});
Hcx(i1) = Hcxi + ((N-i1)*q_x)/mf;
Tcx(i1) = refpropm('T', 'P', props(10,1), 'H', Hcx(i1), FLD, {...
FRCT1 FRCT2});
dT = Thx(i1) - Tcx(i1);
if i1==N & & i1>1
Thxo = min(Thx);
Tcxo = max(Tcx);
% PHE Evaporator Analysis, friction pumping losses
if mk_phe == 1
\{ wf_reg_h, wf_reg_c, Qphe, check, error \} = F1b_Hx_Phe(FLD,...
FLD, mf, mf, Thxi, Thxo, Tcxi, Tcxo, props(8,1), props(2,1),...)
Q_x, 'Reg', directory_iter, power);
Qhx_h = Q_x;  % W
Qhx_c = Qphe;  % W
% Correct Tcx
for i2 = 1:N
q_x = Qhx_c/(N-1);
Hcx(i2) = Hcx + ((N-i2)*q_x)/mf;
Tcx(i2) = refpropm('T', 'P', props(10,1), 'H', Hcx(i2), FLD, {...
FRCT1 FRCT2});
end
else
\{ wf_reg_h, wf_reg_c \} = deal(0);
Qhx_h = Q_x;  % W
Qhx_c = Q_x;  % W
end
% Plot Pinch Diagram For Regenerator
if mk_plots == 1
figure(2)
if dT<dT_pinch
Thxo = Thxo + 1.0;
if Thxo >= Thxi
    check = 1;
    fprintf('nError in pinch_rege: Thxo > Thxi\n')
    return
end
break
end
end
function [S3_2] = F1b_SC_State3(S2_11,Scrit,P2,FLD)
% global P2 FLD
%State Point 3 - SC
%First derivative for finding 3
n = 50;
S3n = (Scrit - S2_11)/(n-1);
S3i = S2_11:S3n:Scrit;
T3i = zeros(1,size(S3i,2));
m3_1 = zeros(2,n);
fprintf('FLD: %s
', FLD)
for i = 1:n;
    T3i(1,i) = refpropm('T', 'P', P2, 'S', S3i(i), FLD);
end
for i = 1:n-1
    m3_1st = (T3i(i+1)-T3i(i))/(S3i(i+1)-S3i(i));
    m3_1(1,i) = S3i(i);
    m3_1(2,i) = m3_1st;
end
m3_1(1,n) = S3i(n);
m3_1(2,n) = m3_1st;
 [~, ind3_1] = min(m3_1(2,:));

%Second derivative for finding 3
for i = 1:n-1
    m3_2nd = (m3_1(2,i+1)-m3_1(2,i))/(S3i(i+1)-S3i(i));
    m3_2(1,i) = S3i(i);
    m3_2(2,i) = m3_2nd;
end
m3_2(1,n) = S3i(n);
m3_2(2,n) = m3_2nd;
 [~, ind3_2] = min(m3_2(2,:));
S3_1 = m3_1(1,ind3_1);
S3_2 = m3_2(1,ind3_2);
function [S4_2] = F1b_SC_State4(Scrit,P2,FLD,props)
% state point 4 - SC
% first derivative for finding 4
n = 50;
S4n = (props(5,5) - Scrit)/(n-1);
S4i = Scrit:S4n:props(5,5);
T4i = zeros(1,size(S4i,2));
m4_1 = zeros(2,n);

for i = 1:n;
    T4i(1,i) = refpropm('T', 'P', P2, 'S', S4i(i), FLD);
end

for i = 1:n-1
    m4_1st = (T4i(i+1)-T4i(i))/(S4i(i+1)-S4i(i));
    m4_1(1,i) = S4i(i);
    m4_1(2,i) = m4_1st;
end
m4_1(1,n) = S4i(n);
m4_1(2,n) = m4_1st;
[~, ind4_1] = max(m4_1(2,:));

% second derivative for finding 4
for i = 1:n-1
    m4_2nd = (m4_1(2,i+1)-m4_1(2,i))/(S4i(i+1)-S4i(i));
    m4_2(1,i) = S4i(i);
    m4_2(2,i) = m4_2nd;
end
m4_2(1,n) = S4i(n);
m4_2(2,n) = m4_2nd;
[~, ind4_2] = max(m4_2(2,:));

S4_1 = m4_1(1,ind4_1);
S4_2 = m4_2(1,ind4_2);
function [Tha,Thb,mfh,ch,wf_evp_h,wf_evp_c,check,error] = ... 
    F1b_Pinch_Ehx(Ehx,mf,regen,FLD_h,Ph,directory_iter,...
    plot_colors,mk_plots,props,FLD,FRCT1,FRCT2,mk_phe,power)

% global check plot_colors mk_plots props FLD FRCT1 FRCT2 mk_phe
check = 0;
error = 0;
Tha = props(5,2) + Exh;
Hfo = props(5,4);
if regen == 0 || props(11,2) == 0
    Hfi = props(2,4);
    Tfi = props(2,2);
elseif regen == 1 && &props(11,2) > 0
    Hfi = props(11,4);
    Tfi = props(11,2);
end
Q = mf*(Hfo-Hfi);
N = 11; %HX segments
q = Q/(N-1);
dT_pinch = 2;
Thb = Tfi;
for i0 = 1:1000
    Th_ave = 0.5*(Tha+Thbi);
    if strcmp(FLD_h,'Water')==1
        ch = refpropm('C','T',Th_ave,'Q',0,FLD_h);
    elseif strcmp(FLD_h,'Thermia')==1
        ch = 3.75*Th_ave + 776.2;  %J/kg-K
    elseif strcmp(FLD_h,'Ethylene-Glycol')==1
        ch = 2930;
    elseif strcmp(FLD_h,'Exhaust')==1
        ch = 1150;
    end
    mfh = Q/(ch*(Tha-Thbi));
    for i1 = 1:N
        Thi(i1) = Tha - (i1-1)*(q/(mfh*ch)); %Brine temp going from hi to lo
        Hfi(i1) = Hfo - (i1-1)*(q/mf);
        {Tfi(i1)} = refpropm('T', 'P', props(2,1), 'H', Hfi(i1), FLD,
            {FRCT1 FRCT2});
        dT = Thi(i1) - Tfi(i1);
        %Check to see if pinch criteria is failed
        if dT<dT_pinch
            Thbi = Thbi + 1.0;
            if Thbi > Tha
                check = 1;
                fprintf('Error in pinch_evap: Thbi > Tha\n')
                return
            end
            break
        end
    end
    mfh = Q/(ch*(Tha-Thbi));
    for i1 = 1:N
        Thi(i1) = Tha - (i1-1)*(q/(mfh*ch)); %Brine temp going from hi to lo
        Hfi(i1) = Hfo - (i1-1)*(q/mf);
        {Tfi(i1)} = refpropm('T', 'P', props(2,1), 'H', Hfi(i1), FLD,
            {FRCT1 FRCT2});
        dT = Thi(i1) - Tfi(i1);
        %Check to see if pinch criteria is failed
        if dT<dT_pinch
            Thbi = Thbi + 1.0;
            if Thbi > Tha
                check = 1;
                fprintf('Error in pinch_evap: Thbi > Tha\n')
                return
            end
            break
        end
if i1==N && i1>1 %Check that
    %PHE Evaporator Analysis, friction pumping losses
    %hx_phe(FLD_h,FLD_c,mfc,mfh,Thi,Tho,Tci,Tco,Ph,Pc,Q,hx_type)
    if mk_phe == 1
        dQ = 0.0;
        while dQ < 0.90 && check ~=1
            mfh = mfh*(1-dQ);
            Thb = min(Thi);
% \{wf\_evp\_h, \ WF\_evp\_c, Qphe\} =
F1b\_Hx\_Phe(FLD\_h,FLD, mf, Tha, Thb, Tfi(N), Tfi(1), Ph, props(2,1), Q,'Evp');

\{wf\_evp\_h,wf\_evp\_c,Qphe,check,error\} = F1b\_Hx\_Phe(FLD\_h,...
FLD, mf, mf, Tha, Thb, Tfi(N), Tfi(1), Ph, props(2,1),...
Q,'Evp',directory\_iter,\_power);

dQ = Qphe/Q;
end
else
\{wf\_evp\_h,wf\_evp\_c\} = deal(0);
end

% Plot Pinch Diagram For Evaporator
if mk\_plots == 1
  figure(2)
  plot(Thi-273, 'LineStyle', '--','Color', plot\_colors(20,:)); hold on
  plot(Tfi-273, 'LineStyle', '-','Color', plot\_colors(30,:));
  plot(Thi-273, 'LineStyle', '--','Color', 'r'); hold on
  plot(Tfi-273, 'LineStyle', '-','Color', 'r');
end
Thb = min(Thi);
return
end
end
end
function [Tca,Tcb,mfc,cc,Dc,wf_cnd_h,wf_cnd_c,check,error] =...
F1b_Pinch_Chx(Chx,mf,regen,FLD_c,Pc,directory_iter,...
plot_colors,mk_plots,props,FLD,FRCT1,FRCT2,mk_phe,power,Tamb)
% global check plot_colors mk_plots props FLD FRCT1 FRCT2 mk_phe directory_cycle Tamb P2 T5 T1
clearvars Tfi Hfi Tci dT
error = 0;
check = 0;
[cc, Dc] = refpropm('CD','T',Tamb,'P',110+Pc,FLD_c);
Tca = props(1,2) - Chx;
Hfo = props(1,4);
if regen == 0 || props(9,2) == 0
    Hfi = props(6,4);
    Tfi = props(6,2);
elseif regen == 1 && props(9,2) > 0
    Hfi = props(9,4);
    Tfi = props(9,2);
end
Q = mf*(Hfi-Hfo);
N = 11; %HX segments
q = Q/(N-1);
dT_pinch = 2;
Tcbi = Tfi;
for i0 = 1:10000
    mfc = Q/(cc*(Tcbi-Tca));
    for i1 = 1:N
        Tci(i1) = Tca + (i1-1)*(q/(mfc*cc)); %Air temp going from lo to hi
        Hfi(i1) = Hfo + (i1-1)*(q/mf); %working fluid enthalpy going from lo to hi
        if regen == 0 || props(9,2) == 0
            Hfi(i1) = props(6,4);
            Tfi(i1) = props(6,2);
        elseif regen == 1 && props(9,2) > 0
            Hfi(i1) = props(9,4);
            Tfi(i1) = props(9,2);
        end
        Q = mf*(Hfi-Hfo);
        N = 11; %HX segments
        q = Q/(N-1);
        dT = Tfi(i1) - Tci(i1);
        if dT<dT_pinch
            Tcbi = Tcbi - 1.0;
            if Tcbi <= Tca
                check = 1;
                fprintf('Error in pinch cond: Tcbi < Tca\n')
                return
            end
            break
        end
        if i1==N && i1>1
            %PHE Condenser Analysis, friction pumping losses
            if mk_phe == 1
                dQ = 0.0;
                while dQ < 0.90 && check ~= 1
                    mfc = mfc*(1-dQ);
                    Tcb = max(Tci);
                    %
                    [wf_cnd_h, wf_cnd_c, Qphe] = F1b_Hx_Phe(FLD,FLD_c,mfc,mf,Tfi(N),Tfi(1),Tca,Tcb,props(1,1),Pc,Q,'Cnd');
                    [wf_cnd_h,wf_cnd_c,Qphe,check,error] = F1b_Hx_Phe(FLD,...
FLD_c,mfc,mf,Tfi(N),Tfi(1),Tca,Tcb,props(1,1),Pc,Q,...
'Cnd',directory_iter,power);
                    dQ = Qphe/Q;
                    end
            else
                [wf_cnd_h,wf_cnd_c] = deal(0);
            end
        end
    end
end
204
if mk_plots == 1
    figure(2)
    plot(Tfi-273, 'LineStyle', '--', 'Color', plot_colors(20,:)); hold on
    plot(Tci-273, 'LineStyle', '--', 'Color', plot_colors(30,:));
    plot(Tfi-273, 'LineStyle', '-', 'Color', 'b'); hold on
    plot(Tci-273, 'LineStyle', '--', 'Color', 'b');
    X_Tick_Labels = {0:10:100};
    set(gca,'XTickLabel',X_Tick_Labels)
    title('Heat Exchanger Temperature Profiles')
    xlabel('Percent Length of Heat Exchanger (%)')
    ylabel('Temperature (°C)')
    if props(11,4) > 0
        legend('Reg_h', 'Reg_c', 'Evp_h', 'Evp_c', 'Cnd_h', 'Cnd_c')
    else
        legend('Evp_h', 'Evp_c', 'Cnd_h', 'Cnd_c')
    end
    if check == 0;
        filename2 = sprintf('%s\Pinch', directory_iter);
        saveas(figure(2), filename2,'fig');
    end
    Tcb = max(Tci);
    return
end
end
function \{qi,qo,wi,wnet,wf_pipe,wf_hx,wf_total,dht,B,n1st,n2nd,ETAT,...
ncarnot,check,error\} = ...
F1b_Cycle_Calc(props,mf,mfh,mfc,ch,Tca,Tha,FLD,Dc,Pc,wf_evp_h,wf_evp_c,...
wf_cnd_h,wf_cnd_c,wf_reg_h,wf_reg_c,check,regen,mk_pipe,mk_cool,...
mk_phe,ETAT,directory_iter)

% global check regen mk_pipe mk_cool directory_pipe mk_phe error
error = 0;

%Determine Heat Exchange, kw
if props(11,4) > 0
qi = mf*(props(5,4) - props(11,4))/1000;
qo = mf*(props(9,4) - props(1,4))/1000;
else
qi = mf*(props(5,4) - props(2,4))/1000;
qo = mf*(props(6,4) - props(1,4))/1000;
end

%work, kw
wi = (mf*(props(2,4) - props(1,4)))/1000;
wo = (mf*(props(5,4) - props(6,4)))/1000;

%Quality check: Wet Expansion Penalty
if props(6,3)<0.99
    fprintf('Error 15: Wet Expansion Penalty\n');
    wo_dry = wo;
    wo = props(6,3)*(mf*(props(5,4) - props(6,4)))/1000;
    %State Point 6 - turbine outlet (reversible) to Recalc ETAT
    ETAT = wo/wo_dry;
else
    ETAT=ETAT;
end

% try
if mk_pipe == 1
%pipe friction calcs
if regen == 0 || props(11,4) == 0
pipe_states = \{2,5,6,1\};
%Pipe 1: Pump Out to Evaporator In (State 2)
%Pipe 2: Evaporator Out to Expander In (State 5)
%Pipe 3: Expander Out to Condenser In (State 6)
%Pipe 4: Condenser Out to Pump In (State 1)
for i=1:size(pipe_states)
    Loss_Factor = 0.1/size(pipe_states,2);
    \{Pf(i),wf(i),Dh_pipe(i),t(i),Lh(i),Fs(i),Re(i)\} = F1b_Pipe(props.pipe_states(i),mf,FLD,wo,qi,Loss_Factor);
end
elseif regen == 1
%If using Regen
pipe_states = \{2,11,5,6,9,1\};
%Pipe 1: Pump Out to Regen Cold In (State 2)
%Pipe 2: Regen Cold Out To Evaporator In (State 11)
%Pipe 3: Evaporator Out to Expander In (State 5)
%Pipe 4: Expander Out to Regen Hot In (State 6)
%Pipe 5: Regen Hot Out to Condenser In (State 9)
%Pipe 6: Condenser Out to Pump In (State 1)
for i=1:size(pipe_states,2)
    Loss_Factor = 0.1/size(pipe_states,2);
    \{Pf(i),wf(i),Dh_pipe(i),t(i),Lh(i),Fs(i),Re(i)\} = F1b_Pipe(props.pipe_states(i),mf,FLD,wo,qi,Loss_Factor);
    fprintf('state: %.0f,wf: %.2f\n',pipe_states(i), wf(i));
end

end

end
%Coolant Line Pumping Losses
wf_pipe = sum(wf);
%Write Pipe Info To File
filename = sprintf('%s\Pipe.txt', directory_iter);
 fid = fopen(filename, 'w');
 fprintf(fid, 'State\tPf_(kpa)\twf_(kW)\DtDh_pipe_(m)\DtLh_(m)\DtFs\DtRe\n');
 for i = 1:size(Pf,2)
 fprintf(fid, num2str(pipe_states(i)));  
 fprintf(fid, '\t');
 fprintf(fid, num2str(Pf(i)));  
 fprintf(fid, '\t');
 fprintf(fid, num2str(wf(i)));  
 fprintf(fid, '\t');
 fprintf(fid, num2str(Dh_pipe(i)));  
 fprintf(fid, '\t');
 fprintf(fid, num2str(t(i)));  
 fprintf(fid, '\t');
 fprintf(fid, num2str(Lh(i)));  
 fprintf(fid, '\t');
 fprintf(fid, num2str(Fs(i)));  
 fprintf(fid, '\t');
 fprintf(fid, num2str(Re(i)));  
 fprintf(fid, '\t');
 fprintf(fid, '\n');
end
fclose(fid);
else
 wf_pipe = 0;
end
% catch ME
%     check = 1;
%     fprintf('Error 20: REFPROP Error in pipe\n');
%     error = 20;
% end

%Heat Exchanger Pumping Losses
if mk_phe == 1
 %Assumes that geo and amb fluid are to be pumped
 wf_hx = wf_evp_h+wf_evp_c+wf_cnd_h+wf_cnd_c+wf_reg_h+wf_reg_c;
else
 wf_hx = 0;
end

%Total Friction Pumping Losses
if mk_cool == 1
 %User defined Pc and assumed fan efficiency of 50%
 n_fan = 0.50;
 w_cool = ((mfc/Dc)*(Pc-101))/n_fan; %kW
else
 w_cool = 0;
end

wf_total = wf_pipe + wf_hx + w_cool;
wnet = max(0,wo - wi - wf_total);
dht = props(5,4) - props(6,4);
%efficiency
n1st = wnet/qi;
B = wnet/mfh; %specific energy
qs = mfh*ch*(Tha-Tca)/1000;
n2nd = ((1-(Tca/props(5,2)))*qi) / ((1-(Tca/Tha))*qs);  % Moran1999 pg 346
ncarnot = 1 - Tca/Tha;
function [Slo, Shi, Tlo, Thi] = F1b_Plot_TS(props, Pcrit, Tcrit, plot_colors, ...)
    TS_Smin, TS_Smax, TS_Tmin, TS_Tmax, FLD, FRCT1, FRCT2, Tgeo)
    \% plots the TS diagram for the fluid
    n = 20;
    \% line width
    l_w = 1;
    color_ts = plot_colors(50,:);
    \% fs = 12;
    \% temperature range to iterate across for plotting
    T_i = TS_Tmin:(0.98*Ts - TS_Tmin)/(n - 1):0.98*Ts;
    \% f is liquid side of vapor dome and g is gas side of vapor dome
    Tf = zeros(size(T_i));
    Tg = zeros(size(T_i));
    Sf = zeros(size(T_i));
    Sg = zeros(size(T_i));
    for k = 1:n;
        \{S\} = refpropm('S', 'T', T_i(k), 'Q', 0, FLD, \{FRCT1 FRCT2\});
        Sf(k) = S;
        \{S\} = refpropm('S', 'T', T_i(k), 'Q', 1, FLD, \{FRCT1 FRCT2\});
        Sg(k) = S;
    end
    figure(1)
    plot(Sf, T_i - 273, '-', 'Color', color_ts, 'lineWidth', l_w); hold on;
    plot(Sg, T_i - 273, '-', 'Color', color_ts, 'lineWidth', l_w);
    plot(refpropm('S', 'T', Tcrit, 'P', Pcrit, FLD, \{FRCT1 FRCT2\}), Tcrit, 'kX');
    xlabel('Entropy, J/kg\C{\circ}C');
    ylabel('T, ^\circ\C');
    \% set axes
    Tmin = min(props(1,2) - 273, TS_Tmin - 273);
    Tmax = max(props(5,2) - 273 TS_Tmax - 273 Tgeo - 273);
    Smin = min(props(1,5), TS_Smin);
    Smax = max(props(6,5), TS_Smax);
    Ta = (Tmin + Tmax)/2;
    Sa = (Smin + Smax)/2;
    Slo = Smin - 0.10*Sa;
    Shi = Smax + 0.10*Sa;
    Tlo = Tmin - 0.10*Ta;
    Thi = Tmax + 0.10*Ta;
    axis([Slo Shi Tlo Thi]);
    title_text = sprintf('T-S Diagram (Fluid: %s)', FLD);
    title(title_text);
function [fig1] = F1b_Plot_Cycle(Tha,Thb,Tca,Tcb,mf,mfh,mfc,qi,qo,qhx,...
    wi,wo,wp.pipe,wp_hx,wp_total,wnet,Slo,Shi,Tlo,Thi,B,ETAT,...
    P2,props,Pcrit,FLD,FRCT1,FRCT2,regen,plot_colors,font_size)

% text offset
Soff = 0.015*(Shi - Slo);
Toff = 0.020*(Thi - Tlo);

% line width
l_w = 1;

%plot 1 - 2 (ORC isentropic)
figure(1)
fig1 = plot({props(1,5) props(2,5)}, {(props(1,2)-273) (props(2,2)-273)}, '-', 'Color', plot_colors(15,:), 'LineWidth',l_w);

%plot 2 - 3 (ORC isobaric)
if P2 < Pcrit & & props(3,5) ~= props(2,5)
    plot({props(2,5) props(3,5)}, {props(2,2) props(3,2)}, 'b-');
    n = 50;
    Sn = (props(3,5) - props(2,5))/(n-1);
    Si = props(2,5):Sn:props(3,5);
    for i = 1:n-1;
        Ti0 = refpropm('T', 'P', props(2,1), 'S', Si(i), FLD, {FRCT1 FRCT2});
        Ti1 = refpropm('T', 'P', props(2,1), 'S', Si(i+1), FLD, {FRCT1 FRCT2});
        fig1 = plot({Si(i) Si(i+1)}, [Ti0-273 Ti1-273], '-', 'Color', plot_colors(15,:), 'LineWidth',l_w);
    end
end

%plot 3 - 4 (ORC isothermal)
if P2 < Pcrit;
    fig1 = plot({props(3,5) props(4,5)}, {props(3,2)-273 props(4,2)-273}, '-', 'Color', plot_colors(15,:), 'LineWidth',l_w);
end

%plot 4 - 5 (ORC isobaric)
if P2< Pcrit & & abs(props(5,5)-props(4,5))>0.0001
    n = 50;
    Sn = (props(5,5) - props(4,5))/(n-1);
    Si = props(4,5):Sn:props(5,5);
    for i = 1:n-1;
        Ti0 = refpropm('T', 'P', props(2,1), 'S', Si(i), FLD, {FRCT1 FRCT2});
        Ti1 = refpropm('T', 'P', props(2,1), 'S', Si(i+1), FLD, {FRCT1 FRCT2});
        fig1 = plot({Si(i) Si(i+1)}, [Ti0-273 Ti1-273], '-', 'Color', plot_colors(15,:), 'LineWidth',l_w);
    end
end

%plot 2 - 5 (SC isobaric)
if P2 > Pcrit;
    n = 50;
    Tn = (props(5,2) - props(2,2))/(n-1);
    Ti = props(2,2):Tn:props(5,2);
    for i = 1:n-1;
        Si0 = refpropm('S', 'T', Ti(i), 'P', props(2,1), FLD, {FRCT1 FRCT2});
        Si1 = refpropm('S', 'T', Ti(i+1), 'P', props(2,1), FLD, {FRCT1 FRCT2});
        fig1 = plot({Si0 Si1}, [Ti(i)-273 Ti(i+1)-273], '-', 'Color', plot_colors(15,:), 'LineWidth',l_w);
    end
end

%plot 5 - 6 (isentropic)
fig1 = plot({props(5,5) props(6,5)}, {props(5,2)-273 props(6,2)-273}, '-', 'Color', plot_colors(15,:), 'LineWidth',l_w);
%plot 6 - 7 (isobaric)
if props(6,3) > 1
    n = 50;
    Sn = (props(6,5) - props(7,5))/(n-1);
    Si = props(7,5):Sn:props(6,5);
    for i = 1:n-1;
        Ti0 = refpropm('T', 'P', props(7,1), 'S', Si(i), FLD, [FRCT1 FRCT2]);
        Ti1 = refpropm('T', 'P', props(7,1), 'S', Si(i+1), FLD, [FRCT1 FRCT2]);
        fig1 = plot([Si(i) Si(i+1)], [Ti0-273 Ti1-273], '--', 'Color', plot_colors(15,:), 'LineWidth',l_w);
    end
end
%plot 7 - 1 (isothermal)
fig1 = plot([props(7,5) props(1,5)], [props(7,2)-273 props(1,2)-273], '--', 'Color', plot_colors(15,:), 'LineWidth',l_w);

%plot high temp hx Tha Thb
if regen == 1 && props(11,5) ~= 0
    Sha = props(5,5);
    Shb = props(11,5);
else
    Sha = props(5,5);
    Shb = props(2,5);
end
fig1 = plot([Shb Sha], [Thb-273 Tha-273], '--', 'Color', plot_colors(14,:), 'LineWidth',l_w);
    text(Shb-4*Soff,Thb-273+2*Toff,'T_h_o','Rotation',0,'FontSize',font_size, 'BackgroundColor','w');
    text(Sha-4*Soff,Tha-273+2*Toff,'T_h_i','Rotation',0,'FontSize',font_size, 'BackgroundColor','w');

%plot cold temp hx Tca, Tcb
if regen == 1 && props(9,5) ~= 0
    Sca = props(1,5);
    Scb = props(9,5);
else
    Sca = props(1,5);
    Scb = props(6,5);
end
fig1 = plot([Sca Scb], [Tca-273 Tcb-273], '--', 'Color', plot_colors(7,:), 'LineWidth',l_w);
    text(Sca,Tca-273+2*Toff,'T_c_i','Rotation',0,'FontSize',font_size, 'BackgroundColor','w');
    text(Scb,Tcb-273+2*Toff,'T_c_o','Rotation',0,'FontSize',font_size, 'BackgroundColor','w');

%plot high temp recuperation fluid
if props(11,4) > 0 && regen == 1
    n = 50;
    Sn = (props(8,5) - props(9,5))/(n-1);
    Si = props(9,5):Sn:props(8,5);
    for i = 1:n-1;
        Ti0 = refpropm('T', 'P', props(1,1), 'S', Si(i), FLD, [FRCT1 FRCT2]);
        Ti1 = refpropm('T', 'P', props(1,1), 'S', Si(i+1), FLD, [FRCT1 FRCT2]);
        fig1 = plot([Si(i) Si(i+1)], [Ti0-273 Ti1-273], 'x', 'Color', plot_colors(15,:), 'LineWidth',l_w);
        i = i+1;
    end
end

%plot low temp recuperation fluid
n = 50;
Sn = (props(11,5) - props(10,5))/(n-1);
Si = props(10,5):Sn:props(11,5);
for i = 1:n-1;
    Ti0 = refpropm('T', 'P', props(2,1), 'S', Si(i), FLD, [FRCT1 FRCT2]);
    Ti1 = refpropm('T', 'P', props(2,1), 'S', Si(i+1), FLD, [FRCT1 FRCT2]);
end
fig1 = plot([Si(i) Si(i+1)], [Ti0-273 Ti1-273], 'x', 'Color', plot_colors(15,:), 'LineWidth',l_w);
i = i+1;
end

% Add state point labels to plot
% Add cycle information to plot
num_form = '%5.2f';
T_off_it = 2.75;
for i = 1:size(plot_vars,1)
    plot_text = {plot_vars{i,1},plot_vars{i,2},plot_vars{i,3}};
    text(Slo+Soff,Thi-(i*T_off_it)*Toff,plot_text,'FontSize',font_size-1);
end
function [results] = F1b_Store_Results(B,mf,mfh,mfc,qi,qo,wi,wf_total,...
wo,wnet,n1st,n2nd,Tha,Thb,qhx,Tca,Tcb,dht,ETAC,ETAT,...
directory_iter,T1,P2,T5,props,FLD,FRCT1,...
FRCT2,Ehx,Chx,check,error,regen,fig1,mk_plots)

%need to have the same number of columns as outputs to write_file
results = cell(1,34);
if check == 1
    results(1,:) = {0};
    results(1,1) = {sprintf('%s', FLD)};
    results(1,2) = {FRCT1};
    results(1,3) = {FRCT2};
    results(1,4) = {P2};
    results(1,5) = {props(1,1)};
    results(1,34) = {error};
elseif check == 0
%store calcs in matrix results
    results(1,1) = {sprintf('%s', FLD)};
    results(1,2) = {FRCT1};
    results(1,3) = {FRCT2};
    results(1,4) = {P2};
    results(1,5) = {props(1,1)}; %Pressure at state point 1, condenser pressure
    results(1,6) = {T1};
    results(1,7) = {T5};
    results(1,8) = {Ehx};
    results(1,9) = {Chx};
    results(1,10) = {dht};
    results(1,11) = ;
    results(1,12) = {((mf/props(6,6)))); %volumetric flow rate at turbine exit
    results(1,13) = {mfh};
    results(1,14) = {mfc};
    results(1,15) = {qi};
    results(1,16) = {qo};
    results(1,17) = {qhx};
    results(1,18) = {regen};
    results(1,19) = ;
    results(1,20) = {wf_total};
    results(1,21) = {wo};
    results(1,22) = {wnet};
    results(1,23) = {n1st};
    results(1,24) = {n2nd};
    results(1,25) = {B};
    results(1,26) = {ETAC};
    results(1,27) = {ETAT};
    results(1,28) = {props(6,3)}; %quality of fluid at turbine exit
    results(1,29) = {Tha};
    results(1,30) = {Thb};
    results(1,31) = {Tca};
    results(1,32) = {Tcb};
    results(1,33) = {props(6,2)};
    results(1,34) = {error};
end

fprintf('\text{-mf: %.2f, mfh: %.2f, mfc: %.2f \text{\/m}}, \{mf,mfh,mfc\})
fprintf('\text{-qi: %.2f, qo: %.2f, qhx: %.2f \text{\/m}}, \{qi,qo,qhx\})
fprintf('\text{-wi: %.2f, wo: %.2f, wf: %.2f, wnet: %.2f \text{\/m}}, \{wi,wo,wf_total,wnet\})
fprintf('\text{-ETAT: %.2f\text{\/m}}, ETAT)
% save cycle plot
if check == 0
    if mk_plots == 1
        filename1 = sprintf('%s\Cycle', directory_iter);
        saveas(max(fig1), filename1, 'fig');
    end
    close all
else
    close all
end
**F1b_Write_Results**

function F1b_Write_Results(results,directory_log,check)

    t = getCurrentTask();
    if isempty(t) == 0
        tmp_out = sprintf('%s\log.%d.txt',directory_log,t.ID);
    else
        tmp_out = sprintf('%s\log.txt',directory_log);
    end
    fid_out = fopen(tmp_out, 'at');

    %print results to data_file
    if check == 0
        fprintf(fid_out, results{1,1});
        fprintf(fid_out, '
');
        for j = 1:size(results,2)-1
            fprintf(fid_out, num2str(results{1,j+1}));
            fprintf(fid_out, '	');
        end
        fprintf(fid_out, '
');
    else
        fprintf('Do not write results, check = 1\n');
    end
    fclose(fid_out);
function F1b_Gather_Files(directory,directory_log,sheet)

% clc; clear all; close all;
% sheet = 'Top_Cal';
% directory = sprintf('%s\Results',sheet);
% directory_log = sprintf('%s\PP_Logs',directory);

header = {'Fluid', 'FRCT1', 'FRCT2', 'P2,(kpa)', 'P1(kpa)', 'T1(K)', 'T5(K)', 'Ehx(K)', 'Chx(K)', ...}
        'delta_h56(J/kg)', 'mf(kg/s)', 'vf6(m3/s)', 'mfh(kg/s)', ...
        'mfc(kg/s)', 'qin(kw)', 'qout(kw)', 'qhx(kw)', 'wout(kw)' ...
        'Q6', 'Tha(K)', 'Thb(K)', 'Tca(K)', 'Tcb(K)', 'T6', 'error';

%%%print header to data_file
file_out = sprintf('%s\%s_outputs.txt',directory,sheet);
fid_out = fopen(file_out,'wt');
for i = 1:size(header,2)
    fprintf(fid_out, header{1,i});
    fprintf(fid_out, '
');
end
fclose(fid_out);

%Retrieve Log Files
files = dir(directory_log);
log_file = '';
j = 1;
for i = 1:size(files,1)
    if files(i).isdir ~= 1
        fprintf('%s
', files(i).name)
        log_file{j} = sprintf('%s\%s', directory_log,files(i).name);
        j=j+1;
    end
end

for i = 1:size(log_file,2)
    %Open Log File
    fid_in = fopen(log_file{i},'r');
    %Determine Number of Lines in Log File
    lines = 0;
    while (fgets(fid_in) ~= -1),
        lines = lines+1;
    end
    fclose(fid_in);
    %Loop Through Log File and Write To output
    for j = 1:lines
        data = textscan(fid_in, '%s', 34, 'delimiter', '\n');
        data = data{1};
        for k = 1:size(data,1)
            fprintf(fid_out, num2str(data{k}));
            fprintf(fid_out, '\n');
        end
        fprintf(fid_out, '\n');
    end
end
end
fclose('all');
Appendix B.  Axial Code

The program written for designing a single stage axial impulse turbine, analysing flow in the nozzle, and searching for the coefficients of the loss model was written in Python. The first program is the Design module which is used to create the geometry of the stator and rotor passage along with other essential turbine geometry. The process flow of the program is as follow;

- Design.py
  - function_geom07_Read_Case
  - function_geom01_noz.py
  - function_geom01_vlt.py
  - function_geom02_2D_path.py
  - function_geom03_2D_surf.py
  - function_geom04_3D_path.py
  - function_geom05_3D_surf.py
  - function_geom06_CAD

The 3D CFD program written uses the in house CFD code Eilmer3. The code that is original to this thesis is the script used to setup the CFD simulation (3D_Eilmer.py) and the post processing function (3D_Eilmer_Post.py). The process flow of the program is as follows;

- 3D_Run_All.sh
  - 3D_Eilmer_Run.sh
    - e3prep.py
      - 3D_Eilmer.py
        - function_geom08_Read_All
          - e3shared.exe
    - e3post.py
    - 3D_Eilmer_Post.py
      - function_geom08_Read_All

The program written to evaluate the experimental turbine data and to calibrate the loss model has the a process flow as follows;

- Loss_Search.py
  - function_geom08_Real_All.py
  - function_coef_gen.py
  - Loss_Model.py
    - function_passage.py

The code for each of the functions used is placed in this appendix so that the work can be recreated and for others to use and build upon in their research.
Design.py

#!/usr/bin/python
import os, sys, time
from math import pi
from matplotlib.pyplot import show, close
sys.path.append('01_functions/)
import function_geom01_noz as F01a
import function_geom01_vlt as F01b
import function_geom02_2D_path as F02
import function_geom03_2D_surf as F03
import function_geom04_3D_path as F04
import function_geom05_3D_surf as F05
import function_geom06_CAD as F06
# import function_geom
import function_geom07_Read_Case as F07
close('all')

## Default Inputs
show_plots = 'Yes' #Setting To Show Plots (Yes or No)
Design_Case = 'R245fa_Redesign_03' #.csv Design File Located In 02_Designs Folder
Machine = 'Impulse' #Type of Velocity Triangle to Use (Impulse, 0Swirl, Symmetric)
Exit_Crit = 'Mach' #Nozzle Design Selection Criteria (Mach or Thrust)
#Mach for selecting based on Mach or Thrust based on Max Thrust

## Create Directory For Design Plots
Design_Dir = '02_Designs/' + Design_Case + '/'
if not os.path.exists(Design_Dir):
os.makedirs(Design_Dir)

## Read In Design Inputs
[Design, Header, Units] = F07.function_geom07_Read_Case(Design_Case)

## Convergent-Divergent Nozzle Geometry
print "---Make 2D Nozzle Design---"
num = 50 #Number of Characteristic lines
theta_i = 0.0001 #Initial step in theta
plotter = 1 #Set to '1' to plot nozzle
max_iter = 1000000 #Maximum number of iterations
strt_coeff_l = 1.00 #Coefficient for straightening portion length of nozzle, >1.0
strt_coeff_w = 1.00 #Coefficient for straightening portion width of nozzle, >1.0
F01a_Out = F01a.function_geom01_noz(Design, num, theta_i, plotter, max_iter, strt_coeff_l,
strt_coeff_w, Exit_Crit, Design_Case, Design_Dir)

A_II = F01a_Out[0] #Mass flow rate, kg/s
mf_d = F01a_Out[1]*Design['N_noz']
Design['T_II'] = F01a_Out[2]-273 #Nozzle exit velocity, m/s
Design['V_II'] = F01a_Out[3] #Nozzle throat height (from low to high radius), m
Design['h_s_tt'] = F01a_Out[4] #Nozzle throat width (sonic width), m
Design['w_s_tt'] = F01a_Out[5]

219
A_s_tt = F01a_Out[6]  # Nozzle throat area, m^2

print "\nV_II: %.2f, T_II: %.2f, h_s_tt: %.6f, w_s_tt: %.6f \n(mf_d, Design['V_II'], Design['T_II'], Design['h_s_tt'], Design['w_s_tt'])

## Velocity Triangle
## ------------------------------------------------------------------------------------------
print "---Make 2D Velocity Triangle---"
F01b_Out = F01b.function_geom01_vlt(Design, Design_Case, Design_Dir)
Design['a_II'] = F01b_Out[0]  # Blade Inlet Absolute Angle, degrees
Design['a_III'] = F01b_Out[1] # Blade Outlet Absolute Angle, degrees
Design['b_II'] = F01b_Out[2]  # Blade Inlet Relative Angle, degrees
Design['b_III'] = F01b_Out[3] # Blade Outlet Relative Angle, degrees
omega = F01b_Out[4]          # Rotor rotational speed, rad/s
W_max = F01b_Out[5]          # Theoretical Work, W
Design['Z'] = F01b_Out[6]    # Speed Work Parameter
Design['U'] = F01b_Out[7]    # Blade tip speed, m/s
Vx_II = F01b_Out[8]         # Blade Absolute Axial Velocity, m/s
W_III = F01b_Out[9]        # Blade Outlet Relative Velocity, m/s
RPM = omega*(60.0/(2.0*pi)) # Rotor RPM

print "\n\nV_II: %.2f, a_II: %.2f, a_III: %.2f, b_II: %.2f, b_III: %.2f, Z: %.2f, U: %.2f \n(Design['V_II'], Design['a_II'], Design['a_III'], Design['b_II'], Design['b_III'], Design['Z'], Design['U'])

## 2D Path Geometry
## ------------------------------------------------------------------------------------------
print "---Make 2D Geometry Paths---"
F02_Out = F02.function_geom02_2D_path(Design, its, Design_Case, Design_Dir)
Design['d_s_tp'] = Design['d_r_m']+Design['h_s_tt']  # nozzle tip diameter, m
Design['d_r_tp'] = Design['d_s_tp']                 # upper radius of blade, m
its = 50                                            # number of points per side

F02_Out = F02.function_geom02_2D_path(Design, its, Design_Case, Design_Dir)
Design['w_r_tt'] = F02_Out[0]                        # Distance between suction to pressure surface of blade, m
Design['pitch'] = F02_Out[1]                         # Blade space or pitch, m
chord_zw = F02_Out[2]                                # Calculated chord based on Zwiefel, m
solidity = F02_Out[3]                                # Solidity
Design['chord'] = F02_Out[4]                         # Actual chord, m
Design['noz_arc'] = F02_Out[5]                       # Length of nozzle exit, m

print "\n\n\n\nV_II: %.2f, pitch: %.2f, chord: %.2f, noz_arc: %.2f \n(Design['w_r_tt'], Design['pitch'], Design['chord'], Design['noz_arc'])

## 2D Surfaces Geometry
## The surf in F03 need to be in the same order as the surf in F05 for
## the F03_outs to work correctly
## ------------------------------------------------------------------------------------------
print "---Make 2D Surfaces---"
F03_Out = F03.function_geom03_2D_surf(Design_Case, Design_Dir)
no_points = F03_Out[0]
VTK_Dim_1 = F03_out[1]
VTK_Dim_2 = F03_out[2]

## 3D Path Geometry
## ------------------------------------------------------------------------------------------
```
# Blade flow area required, m^2, Kearton, W.J. 1966 pp191
hbc = 1.000
A_r_II = (Design['mf']/Design['N_noz'])*(W_III/Design['Rho_II'])
# Blade height, m, Kearton, W.J. 1966 pp191
h_bc = hbc*(A_r_II/Design['w_r_tt'])*(Design['pitch']/Design['noz_arc'])
# Blade flow area active, m^2, Kearton, W.J. 1966 pp191
A_r_actv = (Design['noz_arc']/Design['pitch'])*Design['h_r_tt']*Design['w_r_tt']
D_s_rt = Design['d_s_tp'] - (2.0*Design['h_s_tt']) # nozzle root diameter, m
D_r_rt = Design['d_s_tp'] - (2.0*Design['h_r_tt']) # blade root diameter, m
its = its/2 # number of points per side for connecting edges
F04_out = F04.function_geom04_3D_path(Design, its, Design_Case, Design_Dir)
Design['l'] = F04_out # Meridional Passage Length

## 3D Surface Geometry
## The surfs in F03 need to be in the same order as the surfs in F05 for the F03_outs to work correctly
F05.function_geom05_3D_surf(Design, no_points, VTK_Dim_1, VTK_Dim_2, Design_Case, Design_Dir)

## Turbine Performance Calculations
psi = W_max/Design['U']**2.0 # Stage loading coefficient, pp. 6-9 eqs 6.37
phi = Vx_II/Design['U'] # Flow coefficient, pp. 6-9 eqs 6.38
Ns = (omega*(Design['mf']/Design['Rho_I'])**0.5)/((Design['H_I']-Design['H_II'])**0.75)

## Write Outputs To Screen
Values = Design.items()
for j in range(len(Values)):
    print Values[j]

## Write Geometry
fw = open("02_Designs"+Design_Case+'.csv', "w")
for k in range(len(Design.keys())):
    fw.write(Header[k] + ',' + str(Design[Header[k]]) + ',' + Units[k] + '
')
fw.close()
```

print "\n---Finish Design---"
if show_plots == 'Yes':
    show()
```
function_geom07_Read_Case

#Function to write geom_config.csv file used by loss_model, 2D_eilmer
import csv

def function_geom07_Read_Case(Design_Case):

    """Read In Conditions From Config File"
    Input_Names = [x[0] for x in csv.reader(open("02_Designs/"+Design_Case+".csv", "r"))]
    Input_Values = [x[1] for x in csv.reader(open("02_Designs/"+Design_Case+".csv", "r"))]
    Input_Units = [x[2] for x in csv.reader(open("02_Designs/"+Design_Case+".csv", "r"))]

    #Define Design Dictionary
    Design ={}
    #Assign Values To Dictionary
    for i in range(len(Input_Values)):
        try:
            Design[Input_Names[i]] = float(Input_Values[i])
        except:
            Design[Input_Names[i]] = Input_Values[i]

    Header = []; Units = []
    for i in range(len(Input_Values)):
        Header.append(Input_Names[i])
        Units.append(Input_Units[i])

    #Print To Screen Dictionary
    Values = Design.items()
    for j in range(len(Values)):
        print Values[j]

    return Design, Header, Units
function_geom01_noz.py

from pylab import *
from scipy import *
from math import pi, atan, asin
from numpy import *
from matplotlib.pyplot import *
import os

def function_geom01_noz(Design, num, theta_i, plotter, max_iter, strt_coeff_l, strt_coeff_w, Exit_Crit, Design_Case, Design_Dir):

    # Plot Setup
    fig_size_set = (7, 4)
    rc('font', **{'family': 'sans-serif', 'sans-serif': ['Times New Roman']})
    rc('font', size=10)
    fig_size_set = (7, 4)
    u_mew = 0.45
    dpi_set = 300
    
    R = 8314.0/Design['mw']  # J/kg-K
    T_I = Design['T_I'] + 273.0
    T_II = Design['T_II'] + 273.0
    
    n_mach_II = (((Design['P_II']/Design['P_I'])**((Design['gamma']-1.0)/(Design['gamma']))-1.0)**(2.0/(Design['gamma']-1.0)))**0.5  # Nozzle exit mach number, eq 8.42
    V_II = n_mach_II*(T_II*Design['gamma']*R)**0.5  # Nozzle exit velocity, eq 8.25 based
    Rho_II = Design['P_II']/(R * T_II)  # Nozzle exit density, eq
    A_II = (Design['mf']/(Design['N_noz'])*(Rho_II*V_II)
    
    A_II_At = ((1.0/n_mach_II**2.0) * (2.0/(Design['gamma']+1.0)) * (1.0+((Design['gamma']-1)/2)*n_mach_t)**2)**0.5  # eq 10.32
    A_t = A_II/A_II_At
    n_mach_t = 1.0
    T_I_Tt = 1.0/(1.0+((Design['gamma']-1)/2)*n_mach_t)  # eq 8.40
    Tt = T_I*T_I_Tt  # temperature at throat
    P_I_Pt = 1.0/(1.0+((Design['gamma']-1)/2)*n_mach_t)**2*(Design['gamma']/(Design['gamma']-1))  # eq 8.40
    Pt = Design['P_I']*P_I_Pt  # pressure at throat
    At = (Tt*Design['gamma']*R)**0.5  # speed of sound at throat
    u_t = n_mach_t*At  # velocity at throat
    
    A_I = (T_I*Design['gamma']*R)**0.5
    Rho_I = Design['P_I']/(R * T_I)
    u_I = A_I*0.1
    n_mach_I = u_I/A_I
    A_I = (Design['mf']/(Design['N_noz'])*(Rho_I*u_I)
    
    h_th = (A_t/Design['h_w'])**0.5
    width = A_t/h_th
    dh = h_th/num

    # Part A: Determine Theoretical Exit conditions for nozzle, find where P becomes u
h = [ ]; Ae = [ ]; A_ratio = [ ]; Ma = [ ]; P = [ ]; Te = [ ]; Vt = [ ]; Ve = [ ]; rhot = [ ]; mdot = [ ]; TT = [ ]

A_star = h_th*width
M = 1.0
dM1 = 0.1
for i in range(max_iter):
    h.append(h_th+i*dh)
    Ae.append(h[i]*width)
    A_Asq = (Ae[i]/A_star)**2.0
    A_ratio.append(sqrt(A_Asq))

# Newton Rhapsdon on Eq. 5.20 - Anderson text
res = 1
if i > 0:
    M = Ma[i-1]
while res > .001:
    M2 = M + dM1
    funa1 = -A_Asq + (1.0 / M ** 2.0) * ((2.0 / (Design['gamma'] + 1.0)) * (1.0 + (Design['gamma'] - 1.0) * M ** 2.0 / 2.0)) ** ((Design['gamma'] + 1.0) / (Design['gamma'] - 1.0))
    funa2 = -A_Asq + (1.0 / M2 ** 2.0) * ((2.0 / (Design['gamma'] + 1.0)) * (1.0 + (Design['gamma'] - 1.0) * M2 ** 2.0 / 2.0)) ** ((Design['gamma'] + 1.0) / (Design['gamma'] - 1.0))
    dv_dm = (funa2 - funa1) / dM1
    M = M - funa1 / dv_dm
    res = abs(funa1)
Ma.append(M)
# Find Pressure
P.append(Design['P_I']*(1+(Design['gamma']-1.0)*Ma[i]**2.0/2.0)**(-Design['gamma']/(Design['gamma']-1.0)))
# Find thrust for each point
Te.append(T_I/(1+(Design['gamma']-1.0)*Ma[i]**2.0/2.0))
Tt.append(T_I/(1+(Design['gamma']-1.0)/2.0))
Ve.append(sqrt(Te[i]*Design['gamma']*R))
Vt.append(sqrt(Tt[i]*Design['gamma']*R))
rhot.append(P[i]/(R*Te[i]))
mdot.append(rhot[i]*Ve[i]*Ae[i])
TT.append(mdot[i]*Ve[i]+(P[i]-Design['P_II'])*Ae[i])

if P[i] < Design['P_II']:
    #Calculate the pressure if shock wave exists at the exit plane
    P_exit = P[i]*(1.0+(Design['gamma']*2.0/(Design['gamma']+1.0))*(Ma[i]**2.0-1.0))
    if P_exit <= Design['P_II']:
        P.append(P_exit)
break

# Part C. Method of Characteristics

if Exit_Crit == 'Mach':
a = Design['Ma_Coeff']*max(Ma)
b = Ma.index(a)

if Exit_Crit == 'Thrust':
a = max(TT)
b = TT.index(a)

M_e = Ma[b] #Mach number at ideal exit
A_max = Ae[b] #Nozzle Exit area at max mach

#Find theta_max by using equation 9.42, the prandtl-meyer function
theta_max = (180.0/pi)*((sqrt((Design['gamma'] + 1.0)/(Design['gamma'] - 1.0)))*atan((sqrt((Design['gamma'] - 1.0))*(M_e**2.0 - 1.0))/((Design['gamma'] + 1.0)))-atan(sqrt(M_e**2.0 - 1.0)))/2.0

#D_theta for each char line
del_theta = (theta_max - theta_i) / (num - 1.0)

# Find
theta = zeros((num,num+1))
M_ex = zeros((num+1,num+1))
nu = zeros((num+1,num+1))
mu = zeros((num+1,num+1))
K_m = zeros((num+1,num+1))
K_p = zeros((num+1,num+1))

for i in range(num):
    for j in range(num):
        if i == 0:
            #Theta for each line (first lines) and the Characteristic Line Constants and
            theta[i,j] = theta_i + del_theta*(j)
            nu[i, j] = theta[i, j]
            K_m[i, j] = theta[i, j] + nu[i, j]  #Minus Characteristic Line Constant, eq 13.17
        elif i > 0:
            K_p[i, j] = -K_m[0, i]

        # Find Thetas
        if j >= i:
            theta[i, j] = del_theta * (j-i)
        else:
            theta[i, j] = theta[j, i]
        nu[i, j] = theta[i, j] - K_p[i, j]
        K_m[i, j] = theta[i, j] + nu[i, j]

        # Prandtl-Meyer function (using Newton Rhapson)
dM = 0.10  # Leave at about .1
    if j == 0:
        M_ex[i,j] = 1.00
    else:
        M_ex[i,j] = M_ex[i, j-1]
        M = M_ex[i, j]

res = 1
while res > .01:
    M2 = M + dM
    # Prandtl-Meyer Function
    funv1 = (-nu[i, j] * (pi / 180.0) + (sqrt((Design['gamma'] + 1.0) / (Design['gamma'] - 1.0)) * atan((sqrt((Design['gamma'] - 1.0) * (M ** 2 - 1.0) / (Design['gamma'] + 1)))) - atan(sqrt(M ** 2 - 1)) - atan((sqrt((Design['gamma'] - 1.0) * (M2 ** 2 - 1.0) / (Design['gamma'] + 1)))) - atan(sqrt(M2 ** 2 - 1))))
    funv2 = (-nu[i, j] * (pi / 180.0) + (sqrt((Design['gamma'] + 1.0) / (Design['gamma'] - 1.0)) * atan((sqrt((Design['gamma'] - 1.0) * (M2 ** 2 - 1.0) / (Design['gamma'] + 1)))) - atan(sqrt(M2 ** 2 - 1))))
    dy_dm = (funv2 - funv1) / dM
\[ M = M - \text{funv1} / dv_{dm} \]
\[ \text{res} = \text{abs} (\text{funv1}) \]
\[ M_{\text{ex}}[i, j] = M \]

# Find the angle \( \mu \), Local Mach Angle (noted on page 731 in edt 4)
\[ \text{mu}[i, j] = (180 / \pi) * \text{asin}(1 / M_{\text{ex}}[i, j]) \]

# Add last point to char line
\[ \text{theta}[i, \text{num}] = \text{theta}[i, \text{num}-1] \]
\[ \text{nu}[i, \text{num}] = \text{nu}[i, \text{num}-1] \]
\[ K_m[i, \text{num}] = K_m[i, \text{num}-1] \]
\[ K_p[i, \text{num}] = K_p[i, \text{num}-1] \]

\[ \text{char} = \text{zeros}((\text{num}+1, \text{num}+2, 2)) \]
\[ \text{test} = \text{zeros}((\text{num}+1,\text{num}+1)) \]
\[ \text{testpty} = \text{zeros}((\text{num}+1,\text{num}+1)) \]
\[ \text{iterm} = \text{zeros}(1,2) \]

for \( i \) in range(\text{num}):
    for \( j \) in range(\text{num}+1):
        # Draw points of intersection
        # Point 1 of all char lines
        if \( j == 0 \):
            \( \text{char}[i, j, 0] = 0 \)
            \( \text{char}[i, j, 1] = h_{th} / 2 \)
        # Where first line hits the symmetry line
        if \( i == 0 \) and \( j == 1 \):
            \( \text{char}[i, j, 0] = (-h_{th} / 2) / \text{tan}((\pi / 180) * (\text{theta}[0, j-1] - \text{mu}[0, j-1])) \)
            \( \text{char}[i, j, 1] = 0 \)
        # Where all other lines hit the symmetry line
        if \( j == i+1 \) and \( j > 1 \):
            \( \text{char}[i, j, 0] = -\text{char}[i-1, j, 1] / \text{tan}((\pi / 180) * (.5 * \text{theta}[i, j-2] - .5 * (\text{mu}[i, j-2] + \text{mu}[i, j-1]))) + \text{char}[i-1, j, 0] \)
            \( \text{char}[i, j, 1] = 0 \)
        \( \text{test}[i, j] = (\text{theta}[i, j-2] - .5 * (\text{mu}[i, j-2] + \text{mu}[i, j-1])) \)
        \( \text{testpty}[i, j] = \text{char}[i-1, j, 1] \)
        \( \text{testptx}[i, j] = \text{char}[i-1, j, 0] \)

# All other data points for char 1 calculated
if \( i == 0 \) and \( j > 1 \) and \( j != i+1 \):
    \( C_p = \text{tan}(\pi / 180) * (.5 * (\text{theta}[i, j-2] + \text{theta}[i, j-1]) + .5 * (\text{mu}[i, j-2] + \text{mu}[i, j-1]))) \)
    \( C_m = \text{tan}((\pi / 180) * (.5 * (\text{theta}[j-1, 0] + \text{theta}[i, j-1]) - .5 * (\text{mu}[j-1, 0] + \text{mu}[i, j-1]))) \)
    \( A = \text{mat}([[1, -C_m], [1, -C_p]]) \)
    \( B = \text{mat}([[\text{char}[0, 0, 1] - \text{char}[0, 0, 0] * C_m], [\text{char}[0, j-1, 1] - \text{char}[0, j-1, 0] * C_p]]) \)
    \( c = \text{inv}(A) * B \)
    \( c = c.T \)
    \( \text{iterm} = c[0, :] \)
    \( \text{char}[i, j, 0] = \text{iterm}[0, 1] \)
    \( \text{char}[i, j, 1] = \text{iterm}[0, 0] \)

# All other points for all char lines calculated
if \( i > 0 \) and \( j := i+1 \) and \( j > 1 \):
    \( C_p = \text{tan}(\pi / 180) * (.5 * (\text{theta}[i, j-2] + \text{theta}[i, j-1]) + .5 * (\text{mu}[i, j-2] + \text{mu}[i, j-1]))) \)
    \( C_m = \text{tan}((\pi / 180) * (.5 * (\text{theta}[i-1, j-1] + \text{theta}[i, j-1]) - .5 * (\text{mu}[i-1, j-1] + \text{mu}[i, j-1]))) \)
    \( A = \text{mat}([[1, -C_m], [1, -C_p]]) \)
B = mat([[char(i - 1, j, 1) - char[i - 1, j, 0] * C_m], [char[i, j-1, 1] - char[i, j-1, 0] * C_p]])
c = inv(A) * B
c = c.T
iterm = c[0, :]
char[i, j, 0] = iterm[0, 1]
char[i, j, 1] = iterm[0, 0]

# Fill in similar points (where char lines share points)
for i in range(1, num):
    for j in range(1, num):
        char[j, i, 0] = char[i - 1, j + 1, 0]
        char[j, i, 1] = char[i - 1, j + 1, 1]

## Make the nozzle shape and extend the char lines to wall
## Initial start point of the nozzle (at throat)
if strt_coeff_l == 1.0 or strt_coeff_w == 1.0:
    print "No, nozzle straightening portion added"
    its_addition = 0  # nozzle extension points
else:
    print "nozzle straightening portion added"
    its_addition = 10  # nozzle extension points
noz = zeros((num+its_addition+1, 2))  # change from zeros((num+1,2)) to allow for one last straight outlet portion
noz[0, 0] = 0
noz[0, 1] = h_th / 2

## Find all the points of the nozzle
for i in range(1, num):
    # Find different slopes and points to intersect
    m1 = tan((pi / 180) * (theta[i - 1, num - 1] + mu[i - 1, num - 1]))
    if i == 1:
        m2 = (pi / 180) * theta_max
    else:
        m2 = ((pi / 180) * (theta[i - 1, num]))
    m3 = ((pi / 180) * (theta[i - 1, num - 1]))
    m4 = tan((m2 + m3) / 2)
    A = mat([[1, -m4], [1, -m1]])
    B = mat([[noz[i - 1, 1] - noz[i - 1, 0] * m4], [char[i - 1, num, 1] - char[i - 1, num, 0] * m1]])
c = inv(A) * B
nc = c.T
iterm = c[0, :]
noz[0, 0] = iterm[0, 1]
noz[0, 1] = iterm[0, 0]

# Extend char lines to wall
char[i - 1, num + 1, 0] = noz[i, 0]
char[i - 1, num + 1, 1] = noz[i, 1]

# Last line
m1 = tan((pi / 180) * (theta[num - 1, num - 1] + mu[num - 1, num - 1]))
m2 = ((pi / 180) * (theta[num - 2, num - 1]))
m3 = ((pi / 180) * (theta[num - 1, num]))
m4 = tan((m2 + m3) / 2)
\[ A = \text{mat}([[1, -m4], [1, -m1]]) \]
\[ B = \text{mat}([[[\text{noz}[\text{num}-1, 1] - \text{noz}[\text{num}-1, 0] * m4], [	ext{char}[\text{num}-1, 1] - \text{char}[\text{num}-1, 0] * m1]]) \]
\[ c = \text{inv}(A) * B \]
\[ c = c.T \]
\[ \text{iterm} = c[0,:] \]
\[ \text{noz}[\text{num}, 0] = \text{iterm}[0, 1] \]
\[ \text{noz}[\text{num}, 1] = \text{iterm}[0, 0] \]

```
# Add last point for straightening portion of nozzle
l_addition = (\text{strt}\_\text{coeff}\_l*\text{noz}[\text{num},0] - \text{noz}[\text{num},0])/(\text{its}\_\text{addition})
w_addition = (\text{strt}\_\text{coeff}\_w*\text{noz}[\text{num},1] - \text{noz}[\text{num},1])/(\text{its}\_\text{addition})
```

```python
if \text{strt}\_\text{coeff}\_l == 1.0 or \text{strt}\_\text{coeff}\_w == 1.0:
    print "No, nozzle straightening portion added"
else:
    l_addition = (\text{strt}\_\text{coeff}\_l*\text{noz}[\text{num},0] - \text{noz}[\text{num},0])/(\text{its}\_\text{addition})
w_addition = (\text{strt}\_\text{coeff}\_w*\text{noz}[\text{num},1] - \text{noz}[\text{num},1])/(\text{its}\_\text{addition})
    for i in range(\text{its}\_\text{addition}):
        print "nozzle straightening portion added"
        \text{noz}[\text{num}+1+i, 0] = \text{noz}[\text{num},0]+(i+1)*l_addition
        \text{noz}[\text{num}+1+i, 1] = \text{noz}[\text{num},1]+(i+1)*w_addition
```

```
# Extend Char Lines to Wall
\text{char}[\text{num}-1, \text{num} + 1, 0] = \text{noz}[\text{num}, 0]
\text{char}[\text{num}-1, \text{num} + 1, 1] = \text{noz}[\text{num}, 1]
```

```
if \text{plotter} == 1:
    # Plot the Nozzle Shape (Top Side)
    \text{fig1} = \text{figure}(1, \text{figsize}=\text{fig}\_\text{size}\_\text{set}); \text{clf()}
    \text{rect = fig1}.\text{patch}; \text{rect}.\text{set}\_\text{facecolor}('white')
    \text{ax = subplot}(2,1,1,\text{axisbg}='w', \text{frameon}=\text{True})
    \text{a} = \text{max}(\text{noz}[:,0])
    \text{plot}(\text{noz}[:,0], \text{noz}[:,1], 'k', \text{mec}='k', \text{color}='k', \text{ms} = 5.0, \text{mew}=u\_\text{mew})  # Nozzle Hi Side
    \text{plot}(\text{a}, \text{A}\_\text{max}/\text{width}/2.0, 'o', \text{mec}='k', \text{color}='k', \text{ms} = 5.0, \text{mew}=u\_\text{mew})  # Nozzle Lo Side
    \text{plot}(\text{a}, \text{A}\_\text{max}/\text{width}/2.0, 'o', \text{mec}='w', \text{color}='w', \text{ms} = 5.0, \text{mew}=u\_\text{mew})  # Char Points Hi Side
    \text{plot}(\text{a}, -\text{A}\_\text{max}/\text{width}/2.0, 'o', \text{mec}='w', \text{color}='w', \text{ms} = 5.0, \text{mew}=u\_\text{mew})  # Char Points Lo Side
    \text{ax}.\text{get}\_\text{xaxis}().\text{tick}\_\text{bottom}()
    \text{ax}.\text{axes}.\text{get}\_\text{xaxis}().\text{set}\_\text{visible}(\text{False})
    \text{# Plot Char Lines}
    \text{for i in range(\text{num})}:
        \text{plot}(\text{char}[i, :, 0], \text{char}[i, :, 1], 'k', \text{mec}='k', \text{color}='grey', \text{ms} = 5.0, \text{mew}=u\_\text{mew})
        \text{plot}(\text{char}[i, :, 0], -\text{char}[i, :, 1], 'k', \text{mec}='k', \text{color}='grey', \text{ms} = 5.0, \text{mew}=u\_\text{mew})

    \text{title}('Minimum Length Nozzle Design')
    \text{ylabel}('Nozzle Width (m)')
```

```
# Find errors in A/A* and Mexit
\text{error}\_\text{Area} = 100*(\text{width}^2*\text{noz}[\text{num}-1,1] - \text{A}\_\text{max})/(\text{A}\_\text{max})
\text{error}\_\text{Mach} = 100*(\text{M}_e - \text{M}_\text{ex}[\text{num}-1,\text{num}-1])/(\text{M}_e)
```

```
# Plot Mach Number and pressure through nozzle using the quasi-1D
# area relations. (Isentropic expansion through nozzle)
\text{Mnoz} = \text{zeros}(\text{1, num}+\text{its}\_\text{addition}+1))
\text{Pnoz} = \text{zeros}(\text{1, num}+\text{its}\_\text{addition}+1))
\text{Mnoz}[0,0] = 1.0  # Choked Flow
\text{M} = \text{Mnoz}[0,0]
```
```
\[ A_{e[i]} = 2 \times \text{noz}[i,1] \times \text{width} \]
\[ A_{\text{Asq}} = \frac{A_{e[i]}}{A_{\text{star}}} \]
\[ A_{\text{ratio[i]}} = \sqrt{A_{\text{Asq}}} \]

# Newton Rhapson on Eq. 5.20 - Anderson text

res = 1.0
if i > 0:
    M = Mnoz[0,i-1]
    while res > 0.001:
        M2 = M + dM1
        funa1 = -A_{\text{Asq}} \quad + \quad \left( \frac{1}{M^2} \right)^2 \frac{1}{2} \left( 1 + \frac{\gamma - 1}{2} M^2 \right) \frac{1}{\gamma - 1}
        funa2 = -A_{\text{Asq}} \quad + \quad \left( \frac{1}{M^2} \right)^2 \frac{1}{2} \left( 1 + \frac{\gamma - 1}{2} M^2 \right) \frac{1}{\gamma - 1}
        dv_{dm} = \frac{funa2 - funa1}{dM1}
        M = M - funa1/dv_{dm}
        res = abs(funa1)

Mnoz[0,i] = M

# Find Pressure
Pnoz[0,i] = Design[\text{'P_I'}] \times \left( 1 + \frac{\gamma - 1}{2} Mnoz[0,i]^2 \right)^{-\frac{1}{\gamma - 1}}

```
ax = subplot(2,1,2,axisbg='w', frameon=True)
lns1 = plot(noz[:,0],Mnoz[0,:], 'o', mec='k', color='w', ms = 5.0, mew=0.5, label=r'$M$')
ylabel('Mach Number')
twinx()
lns2 = plot(noz[:,0],Pnoz[0,:]/Design['P_II'], 'x', mec='k', color='w', ms = 5.0, mew=0.5, label=r'$P$')
xlabel('Nozzle length (m)')
ylabel('Pressure (MPa)'); grid('on')
lns = lns1+lns2
labs = [l.get_label() for l in lns]
legend(lns, labs, loc='lower right', shadow=True)
setp(legend.text, fontsize=8)
fig1.savefig(Design_Dir + Design_Case + '-MOC.svg', dpi=dpi_set,format='svg')
fig1.savefig(Design_Dir + Design_Case + '-MOC.png', dpi=dpi_set,format='png')
```

Directory = Design_Dir + '00_Nozzle/'
if not os.path.exists(Directory):
    os.makedirs(Directory)

# write nozzle y,z coords to txt files, BC
fw = open(Directory + 'input_geom_BC.dat', "w")
count = noz.shape[0]
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (0.0, -noz[i,1], noz[i,0]))
fw.close()

# write nozzle x,y coords to txt files, ON
fw = open(Directory + 'input_geom_ON.dat', "w")
count = noz.shape[0]
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (0.0, noz[i,1], noz[i,0]))
fw.close()
V_exit = Ve[b] #Nozzle exit velocity at max mach
print "Finished Making Nozzle"
return A_exit, mf_exit, T_exit, V_exit, width, h_th, A_t
from scipy import *
import matplotlib.cm as cm
from matplotlib.pyplot import *
def function_geom01_vlt(Design, Design_Case, Design_Dir):
    """Plot Setup"""
    fig_size_set = (7,4)
    rc('font',**{'family':'sans-serif','sans-serif':['Times New Roman']})
    rc("font", size=10)
    fig_size_set = (7,4)
    u_mew = 0.45
    dpi_set = 300
    a_II_range = linspace(Design['a_II_min'], Design['a_II_max'], Design['its'])  #Nozzle Absolute Angle, rad...0.8 to 1.4
    b_III_range = linspace(0.00, 1.40, Design['its'])                              #Nozzle Absolute Angle, rad...0.8 to 1.4
    Z_range = linspace(Design['Z_min'], Design['Z_max'], Design['its'])      #Speed-work parameter range   0.001 to 0.999
    Vu_II = []; Vx_II = []; Vu_III = []; U = []; W_II = []; Wu_II = []; Wx_II = []
    Vx_III = []; W_III = []; Wu_III = []; V_III = []; Wx_III = []; delta_Vu = [];
    Wu_III = []; a_II = []; a_III = []; b_III = []; b_j = []; n_static = []; Z = []
    W_iso = 0.5 * Design['mf'] * Design['V_II']**2.0
    if Design['Machine'] == "Impulse":
        count3 = max(1,len(Z_range))
        count4 = max(1,len(a_II_range))
        for i3 in range(count3):
            for i4 in range(count4):
                a_II.append(a_II_range[i4])
                Z.append(Z_range[i3])
                #For an impulse machine only: Constant Axial Velocity, W1=W2, U1=U2, and Reaction = 0
                Vu_II.append(Design['V_II']*[sin(a_II[-1])])                 #Based on trig
                Vx_II.append(Design['V_II']*[cos(a_II[-1])])                 #Based on trig
                U.append(Z[-1]*(Vu_II[-1]-Vx_II[-1]))          #From nasa Velocity Diagram paper (Warren J. Whitney), eqs
                Wx_II.append(Vx_II[-1])                       #Base
                Wu_II.append(Vu_II[-1])               #Based on trig
                V_III.append((Vu_III[-1]**2+Wx_III[-1]**2)**0.5)Based on trig
                W_III.append(Wx_III[-1])#Assumption: Impulse machine, no reaction thus relative velocity is constant
    Kw = W_III[i3]/W_II[i3]                        #rotor relative-velocity ratio for
    #Static Stage Efficiency
\[
\text{Lst} = \text{Design['Kst']} \times (\text{Design['V_II']}^2 + \text{Design['V_II']}^2)/2 \quad \# \text{Stator Loss, From nasa Velocity Diagram paper (Warren J. Whitney), eqs 3-15a}
\]

\[
\text{Lro} = \text{Design['Kro']} \times (\text{W_II[1]-1}^2 + \text{W_III[1]-1}^2)/2 \quad \# \text{Rotor Loss, From nasa Velocity Diagram paper (Warren J. Whitney), eqs 3-15b}
\]

\[
\text{n_static}_i = (\text{W_noloss[1]-1} - \text{Lst} - \text{Lro}) / \text{W}_\text{iso} \quad \# \text{Static Efficiency, based on eq 3-13 and 3-12}
\]

\[
\text{n_static}_.append(\text{n_static}_i)
\]

if Design['Machine'] == "0Swirl":
    count3 = len(Z_range)
    count4 = len(a_II_range)
    for i3 in range(count3):
        for i4 in range(count4):
            b_2ri = 0.00
            b_2ri_running = True
            while b_2ri_running:
                Zi = Z[i3]
                a_1ri = a_II[i4]
                b_2ri = b_2ri + 0.01
                # For a zero exit swirl machine: Constant Axial Velocity, Reaction = 1 - 1/2Zi, U1=U2
                Reaction = 1 - (1/(2*Zi))  # From nasa Velocity Diagram paper (Warren J. Whitney), eqs 3-6
                Vu_1i = Design['V_II'*sin(a_1ri)]  # Based on trig
                Vx_1i = Design['V_II'*cos(a_1ri)]  # Based on trig
                V_2i = (Vu_2i)**2 + (Vx_2i)**2 + 0.5  # Based on trig
                W_1i = (2*Zi*W_1i)**2 + 0.5  # Based on trig
                W_2i = W_2i*sin(b_2ri)  # Based on trig
                Wx_2i = W_2i*cos(b_2ri)  # Based on trig
                U_2i = Vu_2i + Vx_2i  # Based on trig
                if b_2ri > a_1ri:
                    b_2ri_running = False
                if abs((U_1i - U_2i)/U_1i) < 0.001:
                    Z_plot.append(Zi)
                    Vu_II_.append(Vu_1i)
                    Vx_II_.append(Vx_1i)
                    U_.append(U_1i)
                    W_II_.append(W_1i)
                    Wx_II_.append(Wx_1i)
                    W_III_.append(W_2i)
                    W_III_.append(Wx_2i)
                    Vx_III_.append(V_2i)
                    Vu_III_.append(Vu_2i)
                    V_III_.append(V_2i)
                    a_II_.append(a_1ri)  # Based on trig
                    a_III_.append(arctan(Vu_III[-1]/Vx_III[-1]))  # Based on trig
                    b_II_.append(arccos(Vx_II[-1]/W_II[-1]))  # Based on trig
                    b_III_.append(b_2ri)  # Based on trig
                    b_j_.append(U[-1]/Design['V_II'])  # blade speed to absolute jet speed ratio
                    delta_Vu_.append(Vu_II[-1] - Vu_III[-1])  # Total difference in tangential velocities at blade inlet and exit
                    W_noloss_.append(Design['mf']*U[-1]*delta_Vu[-1])  # Work, Watts
                    # Static Stage Efficiency
Lst = Design['Kst'] * ((Design['V_I']**2) + (Design['V_II']**2))/2               #Stator Loss, From nasa Velocity Diagram paper (Warren J. Whitney), eqs 3-15a

Lro = Design['Kro'] * (W_I - W_II)**2 + W_III**2)/2       #Rotor Loss, From nasa Velocity Diagram paper (Warren J. Whitney), eqs 3-15b

n_static_i = (W_noloss - Lst - Lro) / W_iso  #Static Efficiency, based on eq 3-13 and 3-12

if Design['Machine'] == "Symmetric":
    count3 = len(Z)
    for i3 in range(count3):
        a_1ri = 0.00
        a_1ri_running = True
        while a_1ri_running:
            Zi = Z[i3]
            a_1ri = a_1ri + 0.005
            #For a symmetric machine: V1 = W2 and V2 = W1, and Reaction = 0.5 eqs 3-9a and 3-9b
            Vu_1i = Design['V_II']*sin(a_1ri)              #Based on trig
            Vx_1i = Design['V_II']*cos(a_1ri)              #Based on trig
            Vx_2i = Vx_1i                                  #Assumption: Constant axial flow velocity
            U_1i = Zi*(Vu_1i - Vx_2i)                     #From nasa Velocity Diagram paper (Warren J. Whitney), eqs 3-3
            Wu_1i = Vu_1i - U_1i                         #Based on trig
            Wx_1i = Vx_1i                                 #Based on trig
            W_1i = (Wu_1i**2 + Wx_1i**2)**0.5             #Based on trig
            W_2i = Design['V_II']                          #Based on assumption for symmetry that V1 = W2
            Wx_2i = Vx_2i                                 #Based on trig
            Vu_2i = U_1i - Vx_2i                         #Based on trig
            V_2i = W_1i                                  #Based on assumption for symmetry that V2 = W1
            U_2i = Wu_2i + Vu_2i                         #Based on trig
            Reaction_i = 1 - (W_1i**2)/(W_2i**2)         #Based on eqs 2-41 and eqs 3-5 states R should equal 0.5 for symmetry
            if a_1ri > alph_max:
                a_1ri_running = False
                if abs((U_1i - U_2i)/U_1i) < 0.01 and abs(0.50 - Reaction_i)/0.50 < 0.005:
                    Z_plot.append(Zi)
                    Vu_II.append(Vu_1i)
                    Vx_II.append(Vx_1i)
                    U.append(U_1i)
                    Wu_II.append(Wu_1i)
                    Wx_II.append(Wx_1i)
                    W_II.append(W_1i)
                    W_III.append(W_2i)
                    Wu_III.append(Wu_2i)
                    Wx_III.append(Wx_2i)
                    Vx_III.append(Vx_2i)
                    Vu_III.append(Vu_2i)
                    V_III.append(V_2i)
            a_II.append(a_1ri)                           #Based on trig
            a_III.append(arctan(Vu_III[-1]/Vx_III[-1]))   #Based on trig
            b_II.append(arccos(Vx_II[-1]/W_II[-1]))       #Based on trig
            b_III.append(arctan(Wu_III[-1]/Wx_III[-1]))   #Based on trig
            b_j.append(U[-1]/Design['V_II'])             #blade speed to absolute jet speed ratio
            delta_Vu.append(Vu_II[-1] + Vx_III[-1])      #Total difference in tangential velocities at blade inlet and exit

W_noloss.append(Design['mf']*U[-1]*delta_Vu[-1])       #Work, Watts
#Static Stage Efficiency
Lst = Design['Kst'] * ((Design['V_I']**2) + (Design['V_II']**2))/2               #Stator Loss, From nasa Velocity Diagram paper (Warren J. Whitney), eqs 3-15a
Lro = Design['Kro'] * (W_II[-1]**2 + W_III[-1]**2)/2  # Rotor Loss, From nasa Velocity Diagram paper (Warren J. Whitney), eqs 3-15b
n_static_i = (W_noloss[-1] - Lst - Lro) / W_iso  # Static Efficiency, based on eq 3-13 and 3-12
n_static.append(n_static_i)

determine max power and omega for desired rotor diameter
W_max = max(W_noloss)
i_maxw = W_noloss.index(W_max)

# Calculate Torque for range of speed-work parameter
count5 = len(Z)
omega = []
Torque = []
for i5 in range(count5):
    omega.append((2.0*U[i5]/Design['d_r_m'])/Design['d_r_m'])
    Torque.append(W_noloss[i5]/omega[i5])

# Plot Work and Torque vs Speed-Work Parameter
fig2 = figure(2,figsize=fig_size_set)
rect = fig2.patch; rect.set_facecolor('white')
subplot(1,1,1,axisbg='w', frameon=True)
lns1 = plot(Z,W_noloss, 'o', mec='k', color='k', ms = 3.0, mew=0.25, label=r'$W$')
lns2 = plot(Z,omega, 'x', mec='k', color='k', ms = 3.0, mew=0.25, label=r'$\omega$')
xlabel('Speed-Work Parameter'); ylabel('Work (W) and Omega (rad/s)'); grid('on')
twinx()
lns3 = plot(Z, Torque, 's', mec='k', color='w', ms = 5.0, mew=u_mew, label=r'$\text{Torque}$')
ylabel('Torque (N-m)')
lns = lns1+lns2+lns3
labs = [l.get_label() for l in lns]
legend(lns, labs, loc='upper left', shadow=True)
fig2.savefig(Design_Dir + Design_Case + '-Speed-Work.svg', dpi=dpi_set,format='svg')
fig2.savefig(Design_Dir + Design_Case + '-Speed-Work.png', dpi=dpi_set,format='png')

# Plot Velocity Triangle
fig3 = figure(3,figsize=fig_size_set)
rect = fig3.patch; rect.set_facecolor('white')
subplot(1,1,1,axisbg='w', frameon=True)
xoff = 0.05*Vu_II[i_maxw]
yoff = 0.10*(Vx_II[i_maxw]+Vx_III[i_maxw]+w_rotor)
ptA = mat([0,0])
ptB = mat([L*sin(a_II[i_maxw]),L*cos(a_II[i_maxw])])
pt1 = mat([ptB[0,0],ptB[0,1]+2*yoff])
pt2 = mat([pt1[0,0]+U[i_maxw],pt1[0,1]])
pt3 = mat([pt1[0,0]+Vu_II[i_maxw],pt1[0,1]])
pt4 = mat([pt3[0,0],pt3[0,1]+Vx_II[i_maxw]])
ptC = mat([pt4[0,0],pt4[0,1]+2*yoff])
ptD = mat([ptC[0,0],ptC[0,1]+w_rotor])
pt5 = mat([ptD[0,0],ptD[0,1]+2*yoff])
pt6 = mat([pt5[0,0]-U[i_maxw],pt5[0,1]])
pt7 = mat([pt6[0,0]+Vu_III[i_maxw],pt6[0,1]])
pt8 = mat([pt7[0,0],pt7[0,1]+Vx_III[i_maxw]])
t_x = min([ptA[0,0], ptB[0,0], ptC[0,0], ptD[0,0], pt1[0,0], pt2[0,0], pt3[0,0], pt4[0,0], pt5[0,0], pt6[0,0], pt7[0,0], pt8[0,0]])
 t_y = min([ptA[0,1], ptB[0,1], ptC[0,1], ptD[0,1], pt1[0,1], pt2[0,1], pt3[0,1], pt4[0,1], pt5[0,1], pt6[0,1], pt7[0,1], pt8[0,1]])

#Plot U1
plot(array([pt1[0,0], pt2[0,0]]), array([pt1[0,1], pt2[0,1]]), 'k')
text(mean([pt1[0,0], pt2[0,0]]),mean([pt1[0,1], pt2[0,1]])-yoff,'U')

#Plot Vu_II
plot(array([pt3[0,0], pt2[0,0]]), array([pt3[0,1], pt2[0,1]]), 'k')
text(mean([pt3[0,0], pt2[0,0]]),pt2[0,1]-yoff,'Vu_II')

#Plot Vx_II
plot(array([pt3[0,0], pt4[0,0]]), array([pt3[0,1], pt4[0,1]]), 'k')
text(mean([pt3[0,0], pt4[0,0]]),mean([pt3[0,1], pt4[0,1]])-xoff,'Vx_II')

#Plot V_II
plot(array([pt1[0,0], pt4[0,0]]), array([pt1[0,1], pt4[0,1]]), 'k')
text(mean([pt1[0,0], pt4[0,0]])-xoff,mean([pt1[0,1], pt4[0,1]]),'V_II')

#Plot W_II
plot(array([pt2[0,0], pt4[0,0]]), array([pt2[0,1], pt4[0,1]]), ':k')
text(mean([pt2[0,0], pt4[0,0]])-xoff,mean([pt2[0,1], pt4[0,1]]),'W_II')

#Plot U2
plot(array([pt6[0,0], pt5[0,0]]), array([pt6[0,1], pt5[0,1]]), 'k')
text(mean([pt5[0,0], pt6[0,0]]),mean([pt5[0,1], pt6[0,1]])-yoff,'U')

#Plot Vu_III
plot(array([pt6[0,0], pt7[0,0]]), array([pt6[0,1], pt7[0,1]]), 'k')
text(mean([pt6[0,0], pt7[0,0]]),pt7[0,1]-yoff,'Vu_III')

#Plot Vx_III
plot(array([pt7[0,0], pt8[0,0]]), array([pt7[0,1], pt8[0,1]]), 'k')
text(mean([pt7[0,0], pt8[0,0]])-2*xoff,mean([pt7[0,1], pt8[0,1]]),'Vx_III')

#Plot V_III
plot(array([pt6[0,0], pt8[0,0]]), array([pt6[0,1], pt8[0,1]]), 'k')
text(mean([pt6[0,0], pt8[0,0]])+xoff,mean([pt6[0,1], pt8[0,1]]),'V_III')

#Plot W_III
plot(array([pt5[0,0], pt8[0,0]]), array([pt5[0,1], pt8[0,1]]), ':k')
text(mean([pt5[0,0], pt8[0,0]]),mean([pt5[0,1], pt8[0,1]])','W_III')

# Add cycle information to plot
# cx = -90.0
# cy = 11.5
# text(cx, 1*cy,'$W_{noloss[i_maxw]}$: %.0f m$^2$/s' % (W_noloss[i_maxw]), fontsize = 9)
# text(cx, 2*cy,'$\alpha_{II}$: %.2f$^\circ$' % (alpha_II[i_maxw]), fontsize = 9)
# text(cx, 3*cy,'$\beta_{III}$: %.2f$^\circ$' % (beta_III[i_maxw]), fontsize = 9)
# text(cx, 4*cy,'$D_m$: %.2f m' % (Design\['d_r_m'\]), fontsize = 9)
# text(cx, 5*cy,'$U$: %.0f m/s' % (U[i_maxw]), fontsize = 9)
# text(cx, 6*cy,'$U/V$: %.2f' % (Z[i_maxw]), fontsize = 9)
# text(cx, 7*cy,'$V_{u,II}$: %.2f' % (Vu_II[i_maxw]), fontsize = 9)
# text(cx, 8*cy,'$V_{u,III}$: %.2f' % (Vu_III[i_maxw]), fontsize = 9)
# text(cx, 9*cy,'$V_{x,III}$: %.2f' % (Vx_III[i_maxw]), fontsize = 9)
# text(cx, 10*cy,'$W_{u,II}$: %.2f' % (Wu_II[i_maxw]), fontsize = 9)
# text(cx, 11*cy,'$W_{u,III}$: %.2f' % (Wu_III[i_maxw]), fontsize = 9)
# text(cx, 12*cy,'$W_{II}$: %.2f' % (W_II[i_maxw]), fontsize = 9)
def function_geom02_2D_path(Design, its, Design_Case, Design_Dir):

    """
    This python script creates nozzle and blade geometry for a supersonic impulse turbine
    The nozzle geometry is generated from a 2D matlab nozzle CD nozzle program, nozzle.m. The geometry is then
    rotated in this script to the specified absolute nozzle inlet angle. From the rotated nozzle geometry
    the blade passage is generated from the given blade dimensions generated from axial.m file.
    """

    ### Plot Setup #################################################################
    fig_size_set = (7,4)
    rc('font', **{'family': 'sans-serif', 'sans-serif': ['Times New Roman']})
    rc("font", size=10)
    fig_size_set = (7,4)
    u_mew = 0.45
    dpi_set = 300

    ###################################################################
    Oz = 0.10 # Origin of nozzle dim rotation x coordinate, m, arbitrary just need to keep the points out of the negative
    Oy = 0.10 # Origin of nozzle dim rotation y coordinate, m
    ###################################################################

    """
    line segment BC. Low side of nozzle supersonic region R2
    input horizontal x,y,z geometry of nozzle and convert to x,y,z coordinates
    shifted by angle alpha, where alpha is the absolute jet angle
    """

    fname = Design_Dir + '00_Nozzle/input_geom_BC.dat'
    fr = open(fname, "r")
    z_bc = []
    y_bc = []
    zi_bc = []
    yi_bc = []

    for line in fr.readlines():
        tks = line.split()
        if len(tks) == 0:
            continue
        x = float(tks[0])
        try:
            y = float(tks[1])
        except:
            y = 0.0
        try:
            z = float(tks[2])
        except:
            z = 0.0
        # translate wrt to an origin of 0
        z_tl = z - Oz
        z_bc.append(z_tl)
        y_bc.append(y)
        zi_bc.append(x)
        yi_bc.append(0.0)
\[ y_{tl} = y - Oy \]
\[ x_{tl} = x \]

#rotate the line geometry about point 0,0,0 in the z,y plane
\[ z_r = z \times \cos(Design['a_II']) - y \times \sin(Design['a_II']) \]
\[ y_r = z \times \sin(Design['a_II']) + y \times \cos(Design['a_II']) \]
\[ x_r = x \]

#translate back to original origin
\[ z_1 = z_r + Oz \]
\[ y_1 = y_r + Oy \]
\[ x_1 = x \]

zi_bc.append(z)
yi_bc.append(y)
zi_bc.append(z_1)
yi_bc.append(y_1)

fr.close()

hy_bc = y  #y coordinate for calculating nozzle exit width, last point of BC
#z and y coords used for rotational calculations of nozzle geometry of ON.
zt_bc = z_1
yt_bc = y_1

"""
line segment AB.  Lo side of nozzle inlet subsonic region R1
"""

z_ab = linspace(z_bc[0] - Design['L_s_in'],z_bc[0],its)
y_ab = linspace(y_bc[0],y_bc[0],its)
count = len(z_ab)

zi_ab = []
yi_ab = []

#for i in range(0,2.1):
for i in range(count):
    #translate wrt to an origin of 0
    z_tl = z_ab[i] - Oz
    y_tl = y_ab[i] - Oy

#rotate the line geometry about point 0,0,0 in the z,y plane
\[ z_r = z_ab[i] \times \cos(Design['a_II']) - y_ab[i] \times \sin(Design['a_II']) \]
\[ y_r = z_ab[i] \times \sin(Design['a_II']) + y_ab[i] \times \cos(Design['a_II']) \]

#translate back to original origin
\[ z_1 = z_r + Oz \]
\[ y_1 = y_r + Oy \]

zi_ab.append(z_1)
yi_ab.append(y_1)

"""
line segment ON.  Hi side of nozzle region R2
"""

fname = Design_Dir + '00_Nozzle/input_geom_ON.dat'
fr = open(fname, "r")
z_on = []
y_on = []
zi_on = []
yi_on = []

for line in fr.readlines():
    tks = line.split()
    if len(tks) == 0:
        continue
    x = float(tks[0])
    try:
        y = float(tks[1])
    except:
        y = 0.0
    try:
        z = float(tks[2])
    except:
        z = 0.0

    #translate wrt to an origin of 0
    z_tl = z - Oz
    y_tl = y - Oy
    x_tl = x

    #rotate the line geometry about point 0,0,0 in the z,y plane
    z_r = z * cos(Design['a_II']) - y * sin(Design['a_II'])
y_r = z * sin(Design['a_II']) + y * cos(Design['a_II'])
x_r = x

    #translate back to original origin
    z_1 = z_r + Oz
    y_1 = y_r + Oy
    x_1 = x

    z_on.append(z)
y_on.append(y)
zi_on.append(z_1)
yi_on.append(y_1)

hy_on = y # y coordinate at the nozzle exit (end point of ON)
noz_h = hy_on - hy_bc # nozzle height at exit before rotation of points
zt_on_rot = zt_bc # last z coordinate on rotated BC line, used as last z point on the straight of ON
yt_on_rot = yt_bc + noz_h/cos(Design['a_II']) # last y coordinate to be used for rotated ON. Based on the nozzle angle
bld_C = pi * Design['d_r_tp'] # circumference of upper radius of blade
pitch = bld_C / Design['N_bld']
w_r_tt = (pitch - Design['r_r']) * sin((pi/2.0) - Design['b_II'])
# pitch = te_rhroat - te_r
solidity = (2.0 / Design['zw']) * (cos(-Design['b_II']) / cos(Design['b_II'])) * sin(Design['b_II'] + Design['b_III'])
# Glassman 4-14
chord_zw = solidity * pitch # Glassman 4-3
r_u = Design['R_ss'] + w_r_tt

# create z,y points for the straight line portion of ON at the nozzle exit
zi_on_strts = linspace(zi_on[-1], zt_on_rot, its)
yi_on_strts = linspace(yi_on[-1], yt_on_rot, its)
count = len(zi_on_strts)
for i in range(1, count):
    zi_on.append(zi_on_strts[i])
yi_on.append(yi_on_strts[i])

fr.close()
```python
##make z and y points of PO such that when rotated, AP forms a straight horizontal line

\[
\begin{align*}
\text{z}_{\text{po}} &= \text{linspace}(\text{z}_{\text{ab}}[0]-\text{Design}'w_{\text{s\_in}}'*\sin((\pi/2)-\text{Design}'a_{\text{II}}'),\text{z}_{\text{on}}[0],\text{its}) \\
\text{y}_{\text{po}} &= \text{linspace}(\text{y}_{\text{ab}}[0]+\text{Design}'w_{\text{s\_in}}'*\cos((\pi/2)-\text{Design}'a_{\text{II}}'),\text{y}_{\text{on}}[0],\text{its})
\end{align*}
\]

\[
\text{count} = \text{len(}z_{\text{po}}\text{)}
\]

\[
\begin{align*}
\text{zi}_{\text{po}} &= [] \\
\text{yi}_{\text{po}} &= []
\end{align*}
\]

For \( i \) in range(count):
#translate wrt to an origin of 0
\[
\begin{align*}
\text{z}_{\text{tl}} &= z_{\text{po}}[i] - O_z \\
\text{y}_{\text{tl}} &= y_{\text{po}}[i] - O_y
\end{align*}
\]

#rotate the line geometry about point 0,0,0 in the z,y plane
\[
\begin{align*}
\text{z}_{\text{r}} &= z_{\text{po}}[i] * \cos(\text{Design}'a_{\text{II}}') - y_{\text{po}}[i] * \sin(\text{Design}'a_{\text{II}}') \\
\text{y}_{\text{r}} &= z_{\text{po}}[i] * \sin(\text{Design}'a_{\text{II}}') + y_{\text{po}}[i] * \cos(\text{Design}'a_{\text{II}}')
\end{align*}
\]

#translate back to original origin
\[
\begin{align*}
\text{z}_1 &= z_{\text{r}} + O_z \\
\text{y}_1 &= y_{\text{r}} + O_y
\end{align*}
\]

\[
\begin{align*}
\text{zi}_{\text{po}}.append(z_1) \\
\text{yi}_{\text{po}}.append(y_1)
\end{align*}
\]

##line segment CD. Lo side of entry gap between nozzle and blade region R3

\[
\begin{align*}
\text{z}_{\text{cd}} &= \text{linspace}(\text{zi}_{\text{bc}}[-1], \text{zi}_{\text{bc}}[-1]+\text{Design}'\text{gap}', \text{its}) \\
\text{c}_{\text{d\_offset}} &= \text{Design}'\text{gap}'*\tan(\text{Design}'a_{\text{II}}')
\end{align*}
\]

\[
\begin{align*}
\text{y}_{\text{cd}} &= \text{linspace}(\text{yi}_{\text{bc}}[-1], \text{yi}_{\text{bc}}[-1]+\text{c}_{\text{d\_offset}}', \text{its})
\end{align*}
\]

#Define start point of blade inlet on the lower arc
\[
\begin{align*}
\text{z}_{\text{d}} &= z_{\text{cd}}[-1] \\
\text{y}_{\text{d}} &= y_{\text{cd}}[-1]
\end{align*}
\]

##Blade Geometry

#Define Lower Arc Points

\[
\text{i\_c} = 10000
\]

\[
\text{psi\_c} = 0.01 \ #\text{psi correction step, m}
\]

For \( i \) in range(i\_c):

#Define Point E
\[
\begin{align*}
\text{z}_{\text{e}} &= \text{Design}'X_{\text{li}}'\*\text{chord}_{zw} + z_{\text{d}} \\
\text{y}_{\text{e}} &= y_{\text{d}} + (z_{\text{e}}-z_{\text{d}})*\tan(\text{Design}'b_{\text{II}}')
\end{align*}
\]

#Define center of arc curvature, point Q
\[
\begin{align*}
\text{z}_{\text{q}} &= z_{\text{e}} + \text{Design}'R_{ss}'\*\cos(\pi/2-\text{Design}'b_{\text{II}}') \\
\text{y}_{\text{q}} &= y_{\text{e}} - \text{Design}'R_{ss}'\*\sin(\pi/2-\text{Design}'b_{\text{II}}')
\end{align*}
\]

#Define Point F
\[
\begin{align*}
\text{z}_{\text{f}} &= z_{\text{q}} + \text{Design}'R_{ss}'\*\cos(\pi/2-\text{Design}'b_{\text{III}}') \\
\text{y}_{\text{f}} &= y_{\text{q}} + \text{Design}'R_{ss}'\*\sin(\pi/2-\text{Design}'b_{\text{III}}')
\end{align*}
\]

#Define Point G
\[
\begin{align*}
\text{z}_{\text{g}} &= z_{\text{f}} + \text{Design}'X_{\text{lo}}'\*\text{chord}_{zw} \\
\text{y}_{\text{g}} &= y_{\text{f}} - \text{Design}'X_{\text{lo}}'\*\text{chord}_{zw}\*\tan(\text{Design}'b_{\text{III}}')
\end{align*}
\]

#Define Upper Arc Points

# h = r_u - Design['R_{ss}'] # blade passage width, perpendicular distance between upper and lower surfaces of blade
```
# Define Point M
\[ z_m = z_d \]
\[ y_m = y_d + \text{Design}'Xsl'*h/cos(\text{Design}'b_{II}') \]

# Define Point L
\[ z_l = z_e - (h*\text{Design}'Xsl'\cdot sin(\text{Design}'b_{II}') \]
\[ y_l = y_e + (h*\text{Design}'Xsl'\cdot cos(\text{Design}'b_{II}') \]

# Define Point K
\[ z_k = z_f + (h*\text{Design}'Xst'\cdot sin(\text{Design}'b_{III}') \]
\[ y_k = y_f + (h*\text{Design}'Xst'\cdot cos(\text{Design}'b_{III}') \]

# Define Point J
\[ z_j = z_g \]
\[ y_j = y_g + (\text{Design}'Xst'\cdot h)/cos(\text{Design}'b_{III}') \]

if \[ z_l \leq z_d \]
\[ \text{Design}'Xli' = \text{Design}'Xli' + \psi_c \]

if \[ z_g \leq z_k \]
\[ \text{Design}'Xlo' = \text{Design}'Xlo' + \psi_c \]

# Define Lower Arc Lines

# Define Line DE
\[ z_{de} = \text{linspace}(z_d, z_e, its) \]
\[ y_{de} = \text{linspace}(y_d, y_e, its) \]

# Define Arc EF
\[ z_{ef} = \text{linspace}(z_e, z_f, its) \]
\[ \text{count} = \text{len}(z_{ef}) \]
\[ \text{ARC}_{ef} = [] \]

for \[ i1 \] in range(\[ \text{count} \]):
\[ \text{ARC}_{ef}.append((\text{Design}'R_{ss}'\cdot 2 - (z_{ef}[i1] - z_q)\cdot 2)**.5 + y_q) \]

# Define Line FG
\[ z_{fg} = \text{linspace}(z_f, z_g, its) \]
\[ y_{fg} = \text{linspace}(y_f, y_g, its) \]

# Define Upper Arc Lines
\[ z_{ml} = \text{linspace}(z_m, z_l, its) \]
\[ y_{ml} = \text{linspace}(y_m, y_l, its) \]

# Define Arc LK
\[ z_{lk} = \text{linspace}(z_l, z_k, its) \]
\[ r_{ui} = \text{linspace}(h*\text{Design}'Xsl'+\text{Design}'R_{ss}',h*\text{Design}'Xst'+\text{Design}'R_{ss}',its) \]
\[ r_{ui} = \text{linspace}(w_{r_{tt}}*\text{Design}'Xsl'+\text{Design}'R_{ss}',w_{r_{tt}}*\text{Design}'Xst'+\text{Design}'R_{ss}',its) \]
\[ \text{count} = \text{len}(z_{lk}) \]

# Define Line KJ
\[ z_{kj} = \text{linspace}(z_k, z_j, its) \]
\[ y_{kj} = \text{linspace}(y_k, y_j, its) \]

Connecting Lines between regions...

# Define line NM
z_nm = linspace(zi_on[-1],z_ml[0],its)
y_nm = linspace(yi_on[-1],y_ml[0],its)

# Define line AP
z_pa = linspace(zi_po[0],zi_ab[0],its)
y_pa = linspace(yi_po[0],yi_ab[0],its)

# Define line OB
z_ob = linspace(zi_on[0],zi_bc[0],its)
y_ob = linspace(yi_on[0],yi_bc[0],its)

# Define line NC
z_nc = linspace(z_nm[0],z_cd[0],its)
y_nc = linspace(y_nm[0],y_cd[0],its)

# Define line MD
z_md = linspace(z_ml[0],z_de[0],its)
y_md = linspace(y_ml[0],y_de[0],its)

# Define line JG
z_jg = linspace(z_kj[-1],z_fg[-1],its)
y_jg = linspace(y_kj[-1],y_fg[-1],its)

# Define line LE
z_le = linspace(z lk[0],z_ef[0],its)
y_le = linspace(ARC lk[0],ARC_ef[0],its)

# Define line KF
z_kf = linspace(z_kj[0],z_fg[0],its)
y_kf = linspace(y_kj[0],y_fg[0],its)

"""
Adjust so that PO is at origin
"""
  z_off = zi_po[0]
y_off = yi_po[0]
zi_bc = zi_bc - z_off
yi_bc = yi_bc - y_off
zi_on = zi_on - z_off
yi_on = yi_on - y_off
zi_ab = zi_ab - z_off
yi_ab = yi_ab - y_off
zi_po = zi_po - z_off
yi_po = yi_po - y_off
z_cd = z_cd - z_off
y_cd = y_cd - y_off
z_nm = z_nm - z_off
y_nm = y_nm - y_off
z_de = z_de - z_off
y_de = y_de - y_off
z_fg = z_fg - z_off
y_fg = y_fg - y_off
z_ef = z_ef - z_off
ARC Ef = ARC_ef - y_off
z_ml = z_ml - z_off
y_ml = y_ml - y_off
z_kj = z_kj - z_off
y_kj = y_kj - y_off
z lk = z lk - z off
ARC lk = ARC lk - y off
z pa = z pa - z off
y pa = y pa - y off
z ob = z ob - z off
y ob = y ob - y off
z nc = z nc - z off
y nc = y nc - y off
z md = z md - z off
y md = y md - y off
z ig = z ig - z off
y ig = y ig - y off
z le = z le - z off
y le = y le - y off
z kf = z kf - z off
y kf = y kf - y off

Write data to .dat files for use in Eilmer simulation
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
This is changed from the 3D version: In this version 0.00 is placed
in the z column because 2D Eilmer only allows for X and Y
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Directory = Design_Dir + '00_2D_Paths/
if not os.path.exists(Directory):
    os.makedirs(Directory)

#line BC .dat
BC = vstack((zi_bc, yi_bc))
BC = BC.T
fw = open(Directory + "2D_BC.dat", "w")
count = len(zi_bc)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (BC[i,0], BC[i,1], 0.00))
fw.close()

#line AB .dat
AB = vstack((zi_ab, yi_ab))
AB = AB.T
fw = open(Directory + "2D_AB.dat", "w")
count = len(zi_ab)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (AB[i,0], AB[i,1], 0.00))
fw.close()

#line ON .dat
ON = vstack((zi_on, yi_on))
ON = ON.T
fw = open(Directory + "2D_ON.dat", "w")
count = len(zi_on)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (ON[i,0], ON[i,1], 0.00))
fw.close()

#line PO .dat
PO = vstack((zi_po, yi_po))

243
PO = PO.T
fw = open(Directory + "2D_PO.dat", "w")
count = len(zi_po)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (PO[i,0], PO[i,1], 0.00))
fw.close()

#line CD .dat
CD = vstack((z_cd,y_cd))
CD = CD.T
fw = open(Directory + "2D_CD.dat", "w")
count = len(z_cd)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (CD[i,0], CD[i,1], 0.00))
fw.close()

#line NM .dat
NM = vstack((z_nm,y_nm))
NM = NM.T
fw = open(Directory + "2D_NM.dat", "w")
count = len(z_nm)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (NM[i,0], NM[i,1], 0.00))
fw.close()

#line DE .dat
DE = vstack((z_de,y_de))
DE = DE.T
fw = open(Directory + "2D_DE.dat", "w")
count = len(z_de)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (DE[i,0], DE[i,1], 0.00))
fw.close()

###line FG .dat
FG = vstack((z_fg,y_fg))
FG = FG.T
fw = open(Directory + "2D_FG.dat", "w")
count = len(z_fg)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (FG[i,0], FG[i,1], 0.00))
fw.close()

###line EF .dat
EF = vstack((z_ef,ARC_ef))
EF = EF.T
fw = open(Directory + "2D_EF.dat", "w")
count = len(z_ef)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (EF[i,0], EF[i,1], 0.00))
fw.close()

###line ML .dat
ML = vstack((z_ml,y_ml))
ML = ML.T
fw = open(Directory + "2D_ML.dat", "w")
count = len(z_ml)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (ML[i,0], ML[i,1], 0.00))
fw.close()
fw.close()

###line KJ .dat
KJ = vstack((z_kj,y_kj))
KJ = KJ.T
fw = open(Directory + "2D_KJ.dat", "w")
count = len(z_kj)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (KJ[i,0], KJ[i,1], 0.00))
fw.close()

###line LK .dat
LK = vstack((z_lk,ARC_lk))
LK = LK.T
fw = open(Directory + "2D_LK.dat", "w")
count = len(z_lk)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (LK[i,0], LK[i,1], 0.00))
fw.close()

###Line PA
PA = vstack((z_pa,y_pa))
PA = PA.T
fw = open(Directory + "2D_PA.dat", "w")
count = len(z_pa)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (PA[i,0], PA[i,1], 0.00))
fw.close()

###Line OB
OB = vstack((z_ob,y_ob))
OB = OB.T
fw = open(Directory + "2D_OB.dat", "w")
count = len(z_ob)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (OB[i,0], OB[i,1], 0.00))
fw.close()

###Line NC
NC = vstack((z_nc,y_nc))
NC = NC.T
fw = open(Directory + "2D_NC.dat", "w")
count = len(z_nc)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (NC[i,0], NC[i,1], 0.00))
fw.close()

###Line MD
MD = vstack((z_md,y_md))
MD = MD.T
fw = open(Directory + "2D_MD.dat", "w")
count = len(z_md)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (MD[i,0], MD[i,1], 0.00))
fw.close()

###Line JG
JG = vstack((z_jg,y_jg))
JG = JG.T
fw = open(Directory + "2D_JG.dat", "w")
count = len(z_jg)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (JG[i,0], JG[i,1], 0.00))
fw.close()

###Line LE
LE = vstack((z_le,y_le))
LE = LE.T
fw = open(Directory + "2D_LE.dat", "w")
count = len(z_le)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (LE[i,0], LE[i,1], 0.00))
fw.close()

###Line KF
KF = vstack((z_kf,y_kf))
KF = KF.T
fw = open(Directory + "2D_KF.dat", "w")
count = len(z_kf)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (KF[i,0], KF[i,1], 0.00))
fw.close()

chord = z_fg[-1] - z_de[0]
arc_s = yi_on[-1] - yi_bc[-1]

```python
# Plot results
fig4 = figure(4,figsize=fig_size_set)
rect = fig4.patch; rect.set_facecolor('white')
subplot(1,1,1,axisbg='w', frameon=True)
plot(zi_bc,yi_bc, '-k')
hold("on"); axis('equal')
plot(zi_on,yi_on, '-k')
plot(zi_ab,yi_ab, '-k')
plot(zi_po,yi_po, '-k')
plot(z_cd,y_cd, 'k--')
plot(z_nm,y_nm, 'k--')
plot(z_de,y_de, '-k')
plot(z_fg,y_fg, '-k')
plot(z_q-z_off, y_q-y_off, 'kx')
plot(z_ef,ARC_ef, '-k')
plot(z_ml,y_ml, '-k')
plot(z_kj,y_kj, '-k')
plot(z lk,ARC lk, '-k')
plot(z_pa,y_pa, 'k:')
plot(z_ob,y_ob, 'k:')
plot(z_nc,y_nc, 'k:')
plot(z_md,y_md, 'k:')
plot(z_jg,y_jg, 'k:')
plot(z_le,y_le, 'k:')
plot(z_kf,y_kf, 'k:')
ylabel('y-axis (m)')
xlabel('z-axis (m)')
title('2D Paths For Single Stator and Rotor Passage')
fig4.savefig(Design_Dir + Design_Case + '-2D-Paths.svg', dpi=dpi_set, format='svg')
```
fig4.savefig(Design_Dir + Design_Case + ' - 2D_Paths.png', dpi=dpi_set, format='png')

return w_r_tt, pitch, chord_zw, solidity, chord, arc_s
```python
#!/usr/bin/python
import os
from pylab import *
from scipy import *
from math import pi
from numpy import *
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt

def function_geom03_2D_surf(Design_Case, Design_Dir):
    ###Plot Setup#################################################################
    fig_size_set = (7,4)
    rc('font',**{'family':'sans-serif','sans-serif':['Times New Roman']})
    rc("font", size=10)
    fig_size_set = (7,4)
    u_mew = 0.45
    dpi_set = 300
    
    ***
    Used to created Structured 2D VTK Files
    ***
    """
    Used to created Structured 2D VTK Files
    """
    geom_dir = Design_Dir + '00_2D_Paths'
    ################################################################
    #Path Order (South, North, West, East)
    geoms = mat([["2D_PO.dat", '2D_AB.dat', '2D_PA.dat', '2D_OB.dat'],
                 ["2D_ON.dat", '2D_BC.dat', '2D_OB.dat', '2D_NC.dat'],
                 ["2D_NM.dat", '2D_CD.dat', '2D_NC.dat', '2D_MD.dat'],
                 ["2D_ML.dat", '2D_DE.dat', '2D_MD.dat', '2D_LE.dat'],
                 ["2D_LK.dat", '2D_EF.dat', '2D_LE.dat', '2D_KF.dat'],
                 ["2D_FG.dat", '2D_KJ.dat', '2D_KF.dat', '2D_JG.dat']])
    surfs = mat(["2D_OBAP.vtk","2D_NCBO.vtk","2D_MDCN.vtk","2D_LEDM.vtk","2D_KFEL.vtk","2D_JGFK.vtk"])
    modes = mat(["SN", 'SN', 'SN', 'SN', 'WE', 'SN'])
    """"""
    ################################################################
    x_p = []
    y_p = []
    z_p = []
    x_s = []
    y_s = []
    z_s = []
    no_points_out = []
    VTK_Dim_1_out = []
    VTK_Dim_2_out = []
    """"""
    for j in range(geoms.shape[0]): #rows in geoms
        row_count = []
        for temp in range(geoms.shape[1]): #columns in geoms
            fname = geom_dir+"/"+geoms[j,temp]
            fr = open(fname, "r")
            row_count.append(len(fr.readlines()))
        rows = max(row_count)
        x_pi = zeros((rows,4)) #x's in order of S,N,W,E
        y_pi = zeros((rows,4)) #y's in order of S,N,W,E
    """"""`
```
\[ z_{\pi} = \text{zeros}(\text{rows}, 4) \] #z's in order of S,N,W,E
\[ x_{\pi} = [] \]
\[ y_{\pi} = [] \]
\[ z_{\pi} = [] \]

#read in all data points from the four paths defining the surface
for i in range(geoms.shape[1]): #columns in geoms
    fname = geom_dir + "/" + geoms[j, i]
    fr = open(fname, "r")
    temp = -1
    for line in fr.readlines():
        tks = line.split()
        if len(tks) == 0:
            continue
        x = float(tks[0])
        try:
            y = float(tks[1])
        except:
            y = 0.0
        try:
            z = float(tks[2])
        except:
            z = 0.0
        temp = temp + 1
        x_pi[temp, i] = x
        y_pi[temp, i] = y
        z_pi[temp, i] = z
    x_p.append(x)       #x's stored for all surfaces for plotting at the end
    y_p.append(y)       #y's stored for all surfaces for plotting at the end
    z_p.append(z)       #z's stored for all surfaces for plotting at the end

if modes[0, j] == 'SN':
    for i_str in range(row_count[2]): #streamwise counter (west path count)
        x_bld = linspace(x_pi[i_str, 2], x_pi[i_str, 3], row_count[0]) #hub to shroud
        z_bld = linspace(z_pi[i_str, 2], z_pi[i_str, 3], row_count[0]) #streamwise
        y_bld = linspace(y_pi[i_str, 2], y_pi[i_str, 3], row_count[0]) #blade to blade
    for i_bld in range(row_count[0]): #blade to blade counter (south path count)
        x_si.append(x_bld[i_bld])
        y_si.append(y_bld[i_bld])
        z_si.append(z_bld[i_bld])

if modes[0, j] == 'WE':
    for i_str in range(row_count[2]): #streamwise counter (west path count)
        x_bld = linspace(x_pi[i_bld[0], i_bld[1], row_count[2]]) #hub to shroud
        z_bld = linspace(z_pi[i_bld[0], i_bld[1], row_count[2]]) #streamwise
        y_bld = linspace(y_pi[i_bld[0], i_bld[1], row_count[2]]) #blade to blade
    for i_bld in range(row_count[0]): #blade to blade counter (south path count)
        x_si.append(x_bld[i_str])
        y_si.append(y_bld[i_str])
        z_si.append(z_bld[i_str])

Directory = Design_Dir + '00_2D_Surfaces/
if not os.path.exists(Directory):
    os.makedirs(Directory)
no_cells = (row_count[0]-1)*(row_count[2]-1)
no_points = len(x_si)
VTK_Dim_1 = row_count[0]
VTK_Dim_2 = row_count[2]
fw = open(Directory+surfs[0, j], "w")
fw.write("#VTK DataFile Version 2.0\n")
fw.write("Surface Data File For " + surfs[0, j] + "\n")
fw.write("ASCII\n")
fw.write("\n")
fw.write("DATASET STRUCTURED GRID\n")
fw.write("DIMENSIONS %.f %.f %.f\n" % (VTK_Dim_1, VTK_Dim_2))
fw.write("POINTS %.f %f %f\n" % (no_points))

for i in range(len(x_si)):
    x_s.append(x_si[i])
    y_s.append(y_si[i])
    z_s.append(z_si[i])
    fw.write(" %.7f %.7f %.7f\n" % (x_si[i], y_si[i], z_si[i]))

fw.close()

no_points_out.append(no_points)
VTK_Dim_1_out.append(VTK_Dim_1)
VTK_Dim_2_out.append(VTK_Dim_2)

fig5 = figure(5, figsize=fig_size_set)
rect = fig5.patch; rect.set_facecolor('white')
subplot(1,1,1, axisbg='w', frameon=True)
ax = Axes3D(fig5)
ms_its = 50/rows
ax.plot(x_p, y_p, z_p, 'o', mec='k', color='w', ms = ms_its, mew=u_mew)  # Plot Path Points
ax.plot(x_s, y_s, z_s, ',', mec='k', color='k', ms = ms_its, mew=u_mew)  # Plot Surface Points
ax.set_xlabel('x-axis')
ax.set_ylabel('y-axis')
ax.set_zlabel('z-axis')
fig5.savefig(Design_Dir + Design_Case + '-2D-Surfaces.svg', dpi=dpi_set, format='svg')
fig5.savefig(Design_Dir + Design_Case + '-2D_Surfaces.png', dpi=dpi_set, format='png')

return no_points_out, VTK_Dim_1_out, VTK_Dim_2_out
show()
def function_geom04_3D_path(Design, its, Design_Case, Design_Dir):

    ###Plot Setu
    p#################################################################
    fig_size_set = (7,4)
    rc('font',**{'family':'sans-serif','sans-serif':['Times New Roman']})
    rc("font", size=10)
    fig_size_set = (7,4)
    u_mew = 0.45
dpi_set = 300

    ****
    This python script transforms 2D geometry to 3D geometry projected onto an arc of user defined radius without
    distorting the geometry (meaning the cross-sectional areas of the geometry of the nozzle and blade are preserved).
    ****

    #####################################################################
    # geom_dir = "00_geom_2D_path" #directory that contains 2D geometry
    geoms = mat([os.listdir(geom_dir)]) #list all files in the 2D geometry directory
    count0 = geoms.shape[1]
    X_plot = []
    Y_plot = []
    Z_plot = []
    Xr_plot = []
    Yr_plot = []
    Zr_plot = []

    #####################################################################
    #define rotational offset (default is based on the mid point of the nozzle exit
fname = geom_dir+"/2D_NC.dat"
fr = open(fname, "r")

X_nc = []
Y_nc = []
Z_nc = []

nc_x0 = []
nc_y0 = []
nc_z0 = []
nc_x1 = []
nc_y1 = []
nc_z1 = []
md_x0 = []
md_y0 = []
md_z0 = []
md_x1 = []
md_y1 = []
md_z1 = []

for line in fr.readlines():
    tks = line.split()
    if len(tks) == 0:
        continue
    x = float(tks[0])
    try:
        y = float(tks[1])
    except:
        y = 0.0
    try:
        z = float(tks[2])
    except:
        z = 0.0
    #Convert 2D (x,y,z) geom to 3D (z,y,x) required by Eilmer
    X_nc.append(z)
    Y_nc.append(y)
    Z_nc.append(x)

x_off = mean(mat([X_nc[0], X_nc[-1]]))
y_off = mean(mat([Y_nc[0], Y_nc[-1]]))
z_off = mean(mat([Z_nc[0], Z_nc[-1]]))

#read in geometry from 2D files and apply coordmap function to each data point and write to SE geom file
for i0 in range(count0): #counter for geometry file
    X0 = []
    Y0 = []
    Z0 = []

    X1 = []
    Y1 = []
    Z1 = []

    r_name = mat(['lo', 'hi'])
    if parts[0,i0] == 'n':
        r = mat([[Design['d_s_rt']/2.0), (Design['d_s_tp']]/2.0))
        r_av = mean(r)
        count1 = r.shape[1]
    if parts[0,i0] == 'b':
        r = mat([[Design['d_r_rt']]/2.0), (Design['d_r_tp']]/2.0])

r_av = mean(r)
count1 = r.shape[1]

for i1 in range(count1): #counter for radius hi and lo for each geometry file
    fname = geom_dir+"/"+geoms[0,i0]
    fr = open(fname, "r")
    X = []
    Y = []
    Z = []
    Xr = []
    Yr = []
    Zr = []

    for line in fr.readlines():
        tks = line.split()
        if len(tks) == 0:
            continue
        x = float(tks[0])
        try:
            y = float(tks[1])
        except:
            y = 0.0
        try:
            z = float(tks[2])
        except:
            z = 0.0
        z = x #set 2D x value to z
        x = r[0,i1] #set 3D x to radius to be used in Coordmap.py

        t = crd.coordmap(x, y, z, r_av)
        M = crd.coordmap(X[i2], Y[i2], Z[i2], r[0,i1])
        Xr.append(M[0,0])
        Yr.append(M[0,1])
        Zr.append(M[0,2])

    data_out = vstack((Xr, Yr, Zr))
    data_out = data_out.T
    geoms_3D = geoms[0,i0].replace('2D', '3D')
fw = open(Directory+r_name[0,i1]+"_"+geom_3D, "w")
count = len(Zr)
for i in range(count):
    fw.write("%.7e %.7e %.7e\n" % (data_out[i,0], data_out[i,1], data_out[i,2]))
fw.close()

#Store beginning and end points for connecting lines
X0.append(data_out[0,0])
Y0.append(data_out[0,1])
Z0.append(data_out[0,2])
X1.append(data_out[-1,0])
Y1.append(data_out[-1,1])
Z1.append(data_out[-1,2])

#Store beginning and end points for gap: Order is '2D_NC.dat', '2D_MD.dat'
if geom[0,i0] == '2D_NC.dat':
    nc_x0.append(data_out[0,0]) #point N
    nc_y0.append(data_out[0,1])
    nc_z0.append(data_out[0,2])
if geom[0,i0] == '2D_MD.dat':
    md_x0.append(data_out[0,0]) #point M
    md_y0.append(data_out[0,1])
    md_z0.append(data_out[0,2])

##Create Connecting Lines
X0_ct = linspace(X0[0],X0[-1],its)
Y0_ct = linspace(Y0[0],Y0[-1],its)
Z0_ct = linspace(Z0[0],Z0[-1],its)
fw = open(Directory+"ct0_"+geom_3D, "w")
count = len(Z0_ct)
for i in range(count):
    Xct_plot.append(X0_ct[i])
    Yct_plot.append(Y0_ct[i])
    Zct_plot.append(Z0_ct[i])
fw.write("%.7e %.7e %.7e\n" % (X0_ct[i], Y0_ct[i], Z0_ct[i]))
fw.close()

X1_ct = linspace(X1[0],X1[-1],its)
Y1_ct = linspace(Y1[0],Y1[-1],its)
Z1_ct = linspace(Z1[0],Z1[-1],its)
fw = open(Directory+"ct1_"+geom_3D, "w")
count = len(Z1_ct)
for i in range(count):
    Xct_plot.append(X1_ct[i])
    Yct_plot.append(Y1_ct[i])
    Zct_plot.append(Z1_ct[i])
fw.write("%.7e %.7e %.7e\n" % (X1_ct[i], Y1_ct[i], Z1_ct[i]))
fw.close()

##Create lines for gap between nozzle exit and blade inlet, lo_3D_cd
x_cd_lo = linspace(nc_x1[0],md_x1[0],its)
y_cd_lo = linspace(nc_y1[0],md_y1[0],its)
z_cd_lo = linspace(nc_z1[0], md_z1[0], its)

fw = open(Directory+"lo_3D_CD.dat", "w")
count = len(z_cd_lo)
for i in range(count):
    Xr_plot.append(x_cd_lo[i])
    Yr_plot.append(y_cd_lo[i])
    Zr_plot.append(z_cd_lo[i])
    fw.write(\"%.7e %.7e %.7e\n\" % (x_cd_lo[i], y_cd_lo[i], z_cd_lo[i]))
fw.close()

##Create lines for gap between nozzle exit and blade inlet, hi_3D_cd
x_cd_hi = linspace(nc_x1[-1], md_x1[-1], its)
y_cd_hi = linspace(nc_y1[-1], md_y1[-1], its)
z_cd_hi = linspace(nc_z1[-1], md_z1[-1], its)

fw = open(Directory+"hi_3D_CD.dat", "w")
count = len(z_cd_hi)
for i in range(count):
    Xr_plot.append(x_cd_hi[i])
    Yr_plot.append(y_cd_hi[i])
    Zr_plot.append(z_cd_hi[i])
    fw.write(\"%.7e %.7e %.7e\n\" % (x_cd_hi[i], y_cd_hi[i], z_cd_hi[i]))
fw.close()

##Create lines for gap between nozzle exit and blade inlet, lo_3D_nm
x_nm_lo = linspace(nc_x0[0], md_x0[0], its)
y_nm_lo = linspace(nc_y0[0], md_y0[0], its)
z_nm_lo = linspace(nc_z0[0], md_z0[0], its)

fw = open(Directory+"lo_3D_NM.dat", "w")
count = len(z_nm_lo)
for i in range(count):
    Xr_plot.append(x_nm_lo[i])
    Yr_plot.append(y_nm_lo[i])
    Zr_plot.append(z_nm_lo[i])
    fw.write(\"%.7e %.7e %.7e\n\" % (x_nm_lo[i], y_nm_lo[i], z_nm_lo[i]))
fw.close()

##Create lines for gap between nozzle exit and blade inlet, hi_3D_nm
x_nm_hi = linspace(nc_x0[-1], md_x0[-1], its)
y_nm_hi = linspace(nc_y0[-1], md_y0[-1], its)
z_nm_hi = linspace(nc_z0[-1], md_z0[-1], its)

fw = open(Directory+"hi_3D_NM.dat", "w")
count = len(z_nm_hi)
for i in range(count):
    Xr_plot.append(x_nm_hi[i])
    Yr_plot.append(y_nm_hi[i])
    Zr_plot.append(z_nm_hi[i])
    fw.write(\"%.7e %.7e %.7e\n\" % (x_nm_hi[i], y_nm_hi[i], z_nm_hi[i]))
fw.close()

##Determine Blade Passage Meridional Length
geom_dir = Design_Dir + '00_3D_Paths'
geoms = mat(['hi_3D_ML.dat', 'hi_3D_LK.dat', 'hi_3D_KJ.dat',
             'lo_3D_ML.dat', 'lo_3D_LK.dat', 'lo_3D_KJ.dat',
             'hi_3D_DE.dat', 'hi_3D_EF.dat', 'hi_3D_FG.dat',
             'lo_3D_DE.dat', 'lo_3D_EF.dat', 'lo_3D_FG.dat'])
count = geoms.shape[1]
Lsum = 0
for i in range(count):
    fname = geom_dir+'/'+geoms[0,i]
    fr = open(fname, "r")
    i1 = 0
    for line in fr.readlines():
        i1 = i1+1
        tks = line.split()
        if len(tks) == 0:
            continue
        try:
            x = float(tks[0])
        except:
            x = 0.0
        try:
            y = float(tks[1])
        except:
            y = 0.0
        try:
            z = float(tks[2])
        except:
            z = 0.0
        if i1 == 1:
            x0 = x; x1 = x
            y0 = y; y1 = y
            z0 = z; z1 = z
        elif i1 > 1:
            x0 = x1; x1 = x
            y0 = y1; y1 = y
            z0 = z1; z1 = z
        L = ((x0-x1)**2.0 + (y0-y1)**2.0 + (z0-z1)**2.0)**0.5
        Lsum = L + Lsum

l = Lsum/4.0

##Plot original and transformed data points
ms_its = 1
fig6 = figure(6,figsize=fig_size_set)
rect = fig6.patch; rect.set_facecolor('white')
subplot(1,1,1,axisbg='w', frameon=True)
ax = Axes3D(fig6)
ax.plot(Xr_plot,Yr_plot,Zr_plot, 'o', mec='k', color='k', ms = ms_its, mew=u_mew)
ax.plot(Xct_plot,Yct_plot,Zct_plot, 'o', mec='k', color='w', ms = ms_its, mew=u_mew)
ax.set_xlabel('x-axis');      ax.set_ylabel('y-axis');      ax.set_zlabel('z-axis')
ax.set_title('3D Paths For Single Stator and Rotor Passage')
fig6.savefig(Design_Dir + Design_Case + '-3D_Paths.svg', dpi=dpi_set,format='svg')
fig6.savefig(Design_Dir + Design_Case + '-3D_Paths.png', dpi=dpi_set,format='png')
# plt.figure(7)
# subplot(221)
# plot(X_plot,Y_plot, 'ro', mec='r', ms = ms_p, mew=0.5)
# plot(Xr_plot,Yr_plot, 'bo', mec='b', ms = ms_p, mew=0.5)
# xlabel('x-axis')
# ylabel('y-axis')
# axis('equal')
# grid('on')
# title('X-Y view')
# subplot(222)
# plot(Y_plot, Z_plot, 'ro', mec='r', ms = ms_p, mew=0.5)
# plot(Yr_plot, Zr_plot, 'bo', mec='b', ms = ms_p, mew=0.5)
# xlabel('y-axis')
# ylabel('z-axis')
# axis('equal')
# grid('on')
# title('Y-Z view')

# subplot(223)
# plot(X_plot, Z_plot, 'ro', mec='r', ms = ms_p, mew=0.5)
# plot(Xr_plot, Zr_plot, 'bo', mec='b', ms = ms_p, mew=0.5)
# xlabel('x-axis')
# ylabel('z-axis')
# axis('equal')
# grid('on')
# title('X-Z view')
# savefig("00_descriptives/geom_3D_rot")

return l
function_geom05_3D_surf.py

#!/usr/bin/python
import os
from pylab import *
from scipy import *
from math import pi
from numpy import *
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import coordmap as crd

def function_geom05_3D_surf(Design, no_points, VTK_Dim_1, VTK_Dim_2, Design_Case, Design_Dir):
    fig_size_set = (7,4)
    rc('font',**{'family':'sans-serif','sans-serif':['Time New Roman']})
    rc("font", size=10)
    fig_size_set = (7,4)
    u_mew = 0.45
    dpi_set = 300
    Directory = Design_Dir + '00_3D_Surfaces/
    if not os.path.exists(Directory):
        os.makedirs(Directory)

    """
    This python script transforms 2D geometry to 3D geometry projected onto an arc of user defined radius without 
distorting the geometry (meaning the cross-sectional areas of the geometry of the nozzle and blade are preserved).
    """
    
    #define rotational offset (default is based on the mid point of the nozzle exit
    fname = Design_Dir + '00_2D_Paths/2D_NC.dat'
    fr = open(fname, "r")
    X nc = []
    Y nc = []
    Z nc = []
    for line in fr.readlines():
        tks = line.split()
        if len(tks) == 0:
            continue
        x = float(tks[0])
        try:
y = float(tks[1])
except:
y = 0.0
try:
z = float(tks[2])
except:
z = 0.0

#Convert 2D (x,y,z) geom to 3D (z,y,x) required by Eilmer
X_nc.append(z)
Y_nc.append(y)
Z_nc.append(x)

X_off = mean(mat([[X_nc[0], X_nc[-1]]]))
Y_off = mean(mat([[Y_nc[0], Y_nc[-1]]]))
Z_off = mean(mat([[Z_nc[0], Z_nc[-1]]]))

#read in geometry from 2D surface files and apply coordmap function to each data point and write to VTK file
for i0 in range(count0):  #counter for geometry file
    r_name = mat(['lo', 'hi'])
    if parts[0,i0] == 'n':
        r = mat([(Design['d_s_rt']/2.0), (Design['d_s_tp']/2.0)])
        r_av = mean(r)
    if parts[0,i0] == 'b':
        r = mat([(Design['d_r_rt']/2.0), (Design['d_r_tp']/2.0)])
        r_av = mean(r)

    for i1 in range(2):  #counter for radius hi and lo for each geometry file
        fr = open(fname, "r")
        X = []
        Y = []
        Z = []
        Xr = []
        Yr = []
        Zr = []
        i = 0 #initial counter to skip VTK header info
        for line in fr.readlines():
            if i > 6:  #skip past the VTK header information
                tks = line.split()
                if len(tks) == 0:
                    continue
                x = float(tks[0])
                try:
                    y = float(tks[1])
                except:
                    y = 0.0
                try:
                    z = float(tks[2])
                except:
                    z = 0.0

                z = x #set 2D x value to z
                x = r[0,i1] #set 3D x to radius to be used in Coordmap.py
                X.append(x-x_off)
                Y.append(y-y_off)
                Z.append(z-z_off)
                X_plot.append(X[-1])
Y_plot.append(Y[-1])
Z_plot.append(Z[-1])
i = i+1 #counter for VTK header skip

##Rotate geometry using coordmap.py
count2 = len(Z)
for i2 in range(count2):
    M = crd.coordmap(X[i2], Y[i2], Z[i2], r_av)
    Xr.append(M[0,0])
    Yr.append(M[0,1])
    Zr.append(M[0,2])
    Xr_plot.append(M[0,0])
    Yr_plot.append(M[0,1])
    Zr_plot.append(M[0,2])

##Write rotated geometry data to file
data_out = vstack((Xr,Yr, Zr))
data_out = data_out.T
geoms_3D = geoms[0,i0].replace('2D', '3D')
fw = open(Directory+r_name[0,i1]+"_"+geoms_3D, "w")
fw.write("#VTK DataFile Version 2.0\n")
fw.write("Surface Data File For " + geoms_3D + "\n")
fw.write("ASCII\n")
fw.write("\n")
fw.write("DATASET STRUCTURED GRID\n")
fw.write("DIMENSIONS %.f %.f %.f\n (VTK_Dim_1[i0], VTK_Dim_2[i0]) #need to be able to get points and cell data for VTK")
fw.write("POINTS %.f %.f %.f\n (no_points[i0])")

for i in range(count):
    fw.write(" %.7f %.7f %.7f\n" % (data_out[i,0], data_out[i,1], data_out[i,2]))
fw.close()

#Retrieve 3D paths to plot with 3D surfs
for i in range(count):
    #fname = "00_geom_3D_path/"+geoms[0,i]
    fname = Design_Dir + "00_3D_Paths/"+geoms[0,i]
    fr = open(fname, "r")
    for line in fr.readlines():
        tks = line.split()
        if len(tks) == 0:
            continue
        x = float(tks[0])
        try:
            y = float(tks[1])
        except:
            y = 0.0
        try:
            z = float(tks[2])
        except:
            z = 0.0
        X_p.append(x)
Y_p.append(y)
Z_p.append(z)
##Plot original and transformed data points
fig7 = figure(7,figsize=fig_size_set)
rect = fig7.patch; rect.set_facecolor('white')
subplot(1,1,1,axisbg='w', frameon=True)
ax = Axes3D(fig7)
ms_its = 1
ax.plot(Xr_plot, Yr_plot, Zr_plot, 'o', mec='k', color='w', ms = ms_its, mew=u_mew)        #Plot Surface Points
ax.plot(X_p, Y_p, Z_p, ',', mec='k', color='k', ms = ms_its, mew=u_mew)        #Plot Path Points
ax.set_xlabel('x-axis')
ax.set_ylabel('y-axis')
ax.set_zlabel('z-axis')
fig7.savefig(Design_Dir + Design_Case + '-3D-Surfaces.svg', dpi=dpi_set,format='svg')
fig7.savefig(Design_Dir + Design_Case + '-3D_Surfaces.png', dpi=dpi_set,format='png')
show()
function_geom06_CAD

#!/usr/bin/python
import os
##from pylab import *
##from scipy import *
##from math import pi
from numpy import *
##from mpl_toolkits.mplot3d import Axes3D
#import matplotlib.pyplot as plt
##import coordmap as crd
from shutil import copyfile
def function_geom06_CAD(Design_Case, Design_Dir):
    r = mat(["lo_3D_", 'hi_3D_'])
    geom_copy = "00_geom_3D_path/"
    geom_copy = Design_Dir + "00_3D_Paths/"
    # geom_past = "00_geom_CAD/Nozzle/"
    geom_past = Design_Dir + "CAD_Nozzle/"
    if not os.path.exists(geom_past):
        os.makedirs(geom_past)
    ##List files to turn into .txt files that CAD can read
    geoms = mat(["AB", 'BC', 'NC', 'ON', 'PO', 'PA",])
    for k in range(r.shape[1]):
        for j in range(geoms.shape[1]):
            #.dat to .csv
            copyfile(geom_copy + r[0,k] + geoms[0,j] + '.dat',
                     geom_past + r[0,k] + geoms[0,j] + '.txt',)
            #.dat to .xls in mm
            fname = geom_copy + r[0,k] + geoms[0,j] + '.dat'
            fr = open(fname, "r")
            X = []
            Y = []
            Z = []
            for line in fr.readlines():
                tks = line.split()
                if len(tks) == 0:
                    continue
                x = float(tks[0])
                try:
                    y = float(tks[1])
                except:
                    y = 0.0
                try:
z = float(tks[2])
except:
z = 0.0
X.append(x * 1000)
Y.append(y * 1000)
Z.append(z * 1000)

XYZ = vstack((X,Y,Z))
XYZ = XYZ.T
fw = open(geom_past + r[0,k] + geoms[0,j] + '.xls', "w")
count = len(XYZ)
for i in range(count):
    fw.write("%.7e\t%.7e\t%.7e\n" % (XYZ[i,0], XYZ[i,1], XYZ[i,2]))
fw.close()

# geom_past = "00_geom_CAD/Blade/"
#Create Blade CAD Geometry Directory In Design_Dir
geom_past = Design_Dir + 'CAD_Blade/
if not os.path.exists(geom_past):
    os.makedirs(geom_past)
###List files to turn into .txt files that CAD can read
geoms = mat(['DE', 'EF', 'FG', 'ML', 'LK', 'KJ', 'MD', 'JG',])

for k in range(r.shape[1]):
    for j in range(geoms.shape[1]):
        ####.dat to .csv
        copyfile(geom_copy + r[0,k] + geoms[0,j] + '.dat',
                geom_past + r[0,k] + geoms[0,j] + '.txt',)
        ####.dat to .xls in mm
        fname = geom_copy + r[0,k] + geoms[0,j] + '.dat'
        fr = open(fname, "r")
        X = []
        Y = []
        Z = []

for line in fr.readlines():
    tks = line.split()
    if len(tks) == 0:
        continue
    x = float(tks[0])
    try:
        y = float(tks[1])
        except:
            y = 0.0
    try:
        z = float(tks[2])
        except:
            z = 0.0
    X.append(x * 1000)
    Y.append(y * 1000)
    Z.append(z * 1000)

XYZ = vstack((X,Y,Z))
XYZ = XYZ.T
fw = open(geom_past + r[0,k] + geoms[0,j] + '.xls', "w")
count = len(XYZ)
for i in range(count):
    fw.write("%.7e\t%.7e\t%.7e\n" % (XYZ[i,0], XYZ[i,1], XYZ[i,2]))
fw.close()
3D_Run_All

#!/bin/sh
#3D_eilmer_run.sh
reset

echo "
-----Begin 3D Eilmer------
"
\ 3D_Eilmer_Run.sh
echo "
-----Finish 3D Eilmer------
"
sleep 2

echo "
-----Begin 3D Eilmer Post------
"
python 3D_Eilmer_Post.py
echo "
-----Finish 3D Eilmer Post------
"

3D_Eilmer_Run.sh

#!/bin/sh
#3D_eilmer_run.sh
reset

sleep 1
#Run Eilmer
e3prep.py --job=3D_Eilmer --do-vrml

echo "
-----e3prep.py finished------
"
time e3shared.exe --job=3D_Eilmer --run

echo "
-----e3shared.exe finished------
"
e3post.py --job=3D_Eilmer --tindx=all --vtk-xml --add-mach
sleep 2

echo "
-----e3post.py finished------
"

#Create Sol_Directory for Solution
while read line
do
data_path=$line
echo "$line"
done < "TempFile.txt"
data_file=$line
echo "data_path:  $data_path"
echo "data_file:  $data_file"

c=0
Sol_Dir=$data_path$data_file'/Solution_3D'
echo "Sol_Dir:  $Sol_Dir"
while [ $c -le 1 ]; do
  if [-d $Sol_Dir ]; then
    echo "$Sol_Dir exists"
a=`echo $Sol_Dir|awk -F"3D" '{print $NF}"
    echo "Sa"
a=$((a+1))
    Sol_Dir=$data_path$data_file'/Solution_3D'$a
  else  
    c=100
  fi
done

echo "
-----$Sol_Dir-----
"

mkdir ${Sol_Dir}

#Make backups of geom_config.csv and 3D_Eilmer.py and save in solution directory
sleep 2
#cp $data_path"$data_file'/'geom_config.csv $Sol_Dir'/'geom_config_back.csv
#mv geom_config_back.csv ${Sol_Dir}
#Move 3D_Eilmer.py $Sol_Dir'/'3D_Eilmer_BU.py
#mv 3D_eilmer_back.py ${Sol_Dir}
#Clean up files and place in correct Sol_Directories
sleep 5
mv plot ${Sol_Dir}
mv hist ${Sol_Dir}
mv flow ${Sol_Dir}
mv grid ${Sol_Dir}
mv heat ${Sol_Dir}
mv block_labels.list ${Sol_Dir}
mv e3shared.log ${Sol_Dir}
mv gas-model.lua ${Sol_Dir}
mv 3D_Eilmer.config ${Sol_Dir}
mv 3D_Eilmer.control ${Sol_Dir}
mv 3D_Eilmer.finish ${Sol_Dir}
mv 3D_Eilmer.times ${Sol_Dir}
mv 3D_Eilmer.fstc_times ${Sol_Dir}
mv 3D_Eilmer.wrl ${Sol_Dir}
mv nohup.out ${Sol_Dir}

echo "
-----All Done-----
"
3D_Eilmer.py

# 3D_eilmer.py
# Local, Run Command in Terminal: \ 3D_eilmer_run.sh
# Mango, Run Command in Terminal to put on Mango and run in background: nohup \3D_eilmer_run.sh &
# To view progress in terminal use and sssc_turb directory: tail -f nohup.out
# To back out just use control c
# To ensure shell file is executable use: chmod +x 3D_eilmer_run.sh

from numpy import mean
from math import pi
import sys, csv
sys.path.append('01_Functions/
import function_geom08_Read_All as F08
gdata.title = "3D Flow through a single stage axial turbine stator and rotor set."
print gdata.title

#### Begin Inputs ###################################################################################
#### For Noz_Fill: No solution directory required
#### For Blade_Fill: Solution_3D_Noiz_Fill directory required
#### For Blade_Pass: Solution_3D_Bld_Fill directory required
data_path = '00_data/02_Nozzles_R245/'
#Source directory (i.e. '00_data/02_Nozzles/' or '')
data_file = '20120601_R245_089_140/'
#Data File (i.e. '20120420_R245_100_125' or '')
Design_Case = 'Base_Case_2Noz'
fw_temp_w = open("TempFile.txt", "w")
fw_temp_w.write(data_path+data_file)
fw_temp_w.close()

Solution = "Noz_Fill"  #Noz_Fill, Blade_Fill, Blade_Pass
Mesh_L = "Coarse"  #Fine or Coarse
select_gas_model(fname="03_Gas_Models/R245fa_150_LUT.lua.gz")
print "uGas Model Selected"

#### End Inputs ###################################################################################

"Read In Conditions From Config File"
[All, Header] = F08.function_geom08_Read_All(data_path+data_file)
mf = mean(All['mf_dq'])  #mass flow rate, kg/s
T_I = mean(All['T_I_dq'])+273.15  #Stator inlet temperature, deg K
T_II = mean(All['T_II_dq'])  #Stator inlet pressure, Pa
Rho_I = mean(All['Rho_I_dq'])  #Stator Inlet Density, kg/m3
P_II = mean(All['P_II_dq'])  #Stator outlet pressure, Pa
T_III = mean(All['T_III_dq'])+273.15  #Rotor outlet temperature, deg K
P_III = mean(All['P_III_dq'])  #Rotor outlet pressure, Pa
d_r_m = mean(All['d_r_m'])  #Rotor mean diameter, m
omega = 2.0*pi*mean(All['RPM_dq'])/60.0  #Rotor radial velocity, rad/s
N_noz = mean(All['N_noz'])  #Number of Stator Nozzles
h_s_tt = mean(All['h_s_tt'])  #Stator Nozzle Height (from lo to hi radius), m
w_s_in = mean(All['w_s_in'])  #Stator Nozzle Inlet Width, m
noz_arc = mean(All['noz_arc'])  #Stator Nozzle Arc Length, m
active_arc = noz_arc*N_noz  #Nozzle active arc length, m
U = (omega * d_r_m)  #Rotor Blade Tip Speed, m/s

#Use Coefficient to allow eilmer to settle desired inlet pressure
A_s_in = w_s_in*h_s_tt  #Stator Inlet Area, m2
V_I = (mf/N_noz)/(Rho_I*A_s_in)
P_I_total = 1.15*(P_I + 0.5*Rho_I*(V_I**2.0))

print "mf: ", mf

267
print "V_I: ", V_I
print "T_I: ", T_I
print "P_I: ", P_I
print "Rho_I: ", Rho_I
print "P_I_total: ", P_I_total
print "P_II: ", P_II
print "T_III: ", T_III
print "P_III: ", P_III
print "d_r_m: ", d_r_m
print "omega: ", omega
print "N_noz: ", N_noz
print "h_s_tt: ", h_s_tt
print "active_arc: ", active_arc

path_dir = '02_Designs/' + Design_Case + '/00_3D_Paths/'  # Directory for path geometry
surf_dir = '02_Designs/' + Design_Case + '/00_3D_Surfaces/'  # Directory for surface geometry

### Finish Inputs  #############################################################################

"Define Mesh"
if Mesh_L == "Coarse":
    nx = 15  # mesh elements along the i axis, hub to shroud
    ny = 15  # mesh elements along the j axis, streamline
    nz = 15  # mesh elements along the k axis, blade to blade

if Mesh_L == "Fine":
    nx = 15  # mesh elements along the i axis, hub to shroud
    ny = 25  # mesh elements along the j axis, streamline
    nz = 50  # mesh elements along the k axis, blade to blade

Define Flow Conditions

if Solution == "Blade_Fill":
    Old_Solution = "03_Solution_Noz_Fill/"  # Directory for old solution
    Blade_Fill = ExistingSolution('3D_eilmer', Old_Solution, 2, 9999, dimensions=3,
                               assume_same_grid=1, zipFiles=1)

if Solution == "Blade_Pass":
    Old_Solution = "03_Solution_Bld_Fill/"  # Directory for old solution
    Blade_Pass = ExistingSolution('3D_eilmer', Old_Solution, 3, 9999, dimensions=3,
                               assume_same_grid=1, zipFiles=1)

state_I = FlowCondition(p=P_I_total, T=T_I)  # Stator Inlet Condition, P(Pa) and T(K)
state_II = FlowCondition(p=P_II, T=T_III)   # Stator Outlet Condition, P(Pa) and T(K)
state_III = FlowCondition(p=P_III, v=U, T=T_III)  # Rotor Outlet Condition, P(Pa) and T(K)

if Solution == "Noz_Fill":
    r1_fill = state_I
    r2_fill = state_II
    print "Noz_Fill Flow Solution"

if Solution == "Blade_Fill":
    r1_fill = Blade_Fill
    r2_fill = Blade_Fill
    r3_fill = state_2_bld_o
    r4_fill = state_2_bld_o
    r5_fill = state_2_bld_o
    r6_fill = state_2_bld_o
    print "Blade Fill Flow Solution"
if Solution == "Blade_Pass":
    r4_fill = Blade_Pass
    r5_fill = Blade_Pass
    r6_fill = Blade_Pass
    print "Blade Pass Flow Solution"

if Solution == "Noz_Fill" or Solution == "Blade_Fill":
    # Region 1: Subsonic Nozzle Entry
    p03_r1 = Spline2(path_dir + "lo_3D_PO.dat")
    p32_r1 = Spline2(path_dir + "ct1_3D_PO.dat")
    p12_r1 = Spline2(path_dir + "hi_3D_PO.dat")
    p01_r1 = Spline2(path_dir + "ct0_3D_PO.dat")
    p47_r1 = Spline2(path_dir + "lo_3D_AB.dat")
    p76_r1 = Spline2(path_dir + "ct1_3D_AB.dat")
    p56_r1 = Spline2(path_dir + "hi_3D_AB.dat")
    p45_r1 = Spline2(path_dir + "ct0_3D_AB.dat")
    p04_r1 = Spline2(path_dir + "lo_3D_PA.dat")
    p37_r1 = Spline2(path_dir + "lo_3D_OB.dat")
    p26_r1 = Spline2(path_dir + "hi_3D_OB.dat")
    p15_r1 = Spline2(path_dir + "hi_3D_PA.dat")
    p32_r1 = Spline2(path_dir + "ct1_3D_PO.dat")
    p76_r1 = Spline2(path_dir + "ct1_3D_AB.dat")
    p56_r1 = Spline2(path_dir + "hi_3D_AB.dat")
    p45_r1 = Spline2(path_dir + "ct0_3D_AB.dat")
    p04_r1 = Spline2(path_dir + "lo_3D_PA.dat")
    p37_r1 = Spline2(path_dir + "lo_3D_OB.dat")
    p26_r1 = Spline2(path_dir + "hi_3D_OB.dat")
    p15_r1 = Spline2(path_dir + "hi_3D_PA.dat")
    #CoonsPatch(pS; pN; pW; pE)
    SN_r1 = CoonsPatch(p32_r1, p76_r1, p37_r1, p26_r1)
    SS_r1 = CoonsPatch(p01_r1, p45_r1, p04_r1, p15_r1)
    ST_r1 = CoonsPatch(p45_r1, p76_r1, p47_r1, p56_r1)
    SB_r1 = CoonsPatch(p01_r1, p32_r1, p03_r1, p12_r1)
    SW_r1 = MeshPatch(surf_dir + "lo_3D_OBAP.vtk")
    SE_r1 = MeshPatch(surf_dir + "hi_3D_OBAP.vtk")
    print "
    Region 1 Surfaces
    
    # Region 1: Subsonic Nozzle Entry
    r1_volume = ParametricVolume(SN_r1, SE_r1, SS_r1, SW_r1, ST_r1, SB_r1)
    r1_blk = Block3D(label ="r1-block ", nni =nx, nnj =2*ny, nnk =nz,
                    parametric_volume = r1_volume,
                    fill_condition = r1_fill,
                    hcell_list =[(nx/nx, ny/2, nz/2)])
    # Region 2: Supersonic Nozzle Exit
    p03_r2 = Spline2(path_dir + "lo_3D_ON.dat")
    p32_r2 = Spline2(path_dir + "ct1_3D_ON.dat")
    p12_r2 = Spline2(path_dir + "hi_3D_ON.dat")
    p01_r2 = Spline2(path_dir + "ct0_3D_ON.dat")
    p47_r2 = Spline2(path_dir + "lo_3D_BC.dat")
    p76_r2 = Spline2(path_dir + "ct1_3D_BC.dat")
    p56_r2 = Spline2(path_dir + "hi_3D_BC.dat")
    p45_r2 = Spline2(path_dir + "ct0_3D_BC.dat")
    p04_r2 = Spline2(path_dir + "lo_3D_OB.dat")
    p37_r2 = Spline2(path_dir + "lo_3D_NC.dat")
    p26_r2 = Spline2(path_dir + "hi_3D_NC.dat")
    p15_r2 = Spline2(path_dir + "hi_3D_OB.dat")
#CoonsPatch(pS; pN; pW; pE)
SN_r2 = CoonsPatch(p32_r2, p76_r2, p37_r2, p26_r2)
SS_r2 = CoonsPatch(p01_r2, p45_r2, p04_r2, p15_r2)
ST_r2 = CoonsPatch(p45_r2, p76_r2, p47_r2, p56_r2)
SB_r2 = CoonsPatch(p01_r2, p32_r2, p03_r2, p12_r2)
SW_r2 = MeshPatch(surf_dir + "lo_3D_NCBO.vtk")
SE_r2 = MeshPatch(surf_dir + "hi_3D_NCBO.vtk")
print "\n\nRegion 2 Surfaces\n"

# Region 2: Supersonic Nozzle Exit
r2_volume = ParametricVolume(SN_r2, SE_r2, SS_r2, SW_r2, ST_r2, SB_r2)
r2_blk = Block3D(label ="r2-block ", nni=nx, nnj=ny, nnk=nz,
                 parametric_volume=r2_volume,
                 fill_condition=r2_fill,
                 hcell_list=[(nx/nx, ny/2, nz/2)])

if Solution == "Blade_Fill":
  # Region 3: Gap between nozzle exit and blade entry
  p03_r3 = Spline2(path_dir + "lo_3D_NM.dat")
p32_r3 = Spline2(path_dir + "ct0_3D_MD.dat")
p12_r3 = Spline2(path_dir + "hi_3D_NM.dat")
p01_r3 = Spline2(path_dir + "ct0_3D_NC.dat")
p47_r3 = Spline2(path_dir + "lo_3D_CD.dat")
p76_r3 = Spline2(path_dir + "ct1_3D_MD.dat")
p56_r3 = Spline2(path_dir + "hi_3D_CD.dat")
p45_r3 = Spline2(path_dir + "ct1_3D_NC.dat")
p04_r3 = Spline2(path_dir + "lo_3D_NC.dat")
p37_r3 = Spline2(path_dir + "lo_3D_MD.dat")
p26_r3 = Spline2(path_dir + "hi_3D_MD.dat")
p15_r3 = Spline2(path_dir + "hi_3D_NC.dat")
print "\n\nRegion 3 Surfaces\n"

# Region 3: Gap between nozzle exit and blade entry
r3_volume = WireFrameVolume(p01_r3, p12_r3, p32_r3, p03_r3, p45_r3, p56_r3, p76_r3,
p47_r3, p04_r3, p15_r3, p26_r3, p37_r3)
r3_blk = Block3D(label ="r3-block ", nni=nx, nnj=ny, nnk=nz,
                 parametric_volume=r3_volume,
                 fill_condition=r3_fill,
                 hcell_list=[])}

if Solution == "Blade_Fill" or Solution == "Blade_Pass":
  # Region 4: Blade inlet
  p03_r4 = Spline2(path_dir + "lo_3D_ML.dat")
p32_r4 = Spline2(path_dir + "ct1_3D_ML.dat")
p12_r4 = Spline2(path_dir + "hi_3D_ML.dat")
p01_r4 = Spline2(path_dir + "ct0_3D_ML.dat")
p47_r4 = Spline2(path_dir + "lo_3D_DE.dat")
p76_r4 = Spline2(path_dir + "ct1_3D_DE.dat")
p56_r4 = Spline2(path_dir + "hi_3D_DE.dat")
p45_r4 = Spline2(path_dir + "ct0_3D_DE.dat")
p04_r4 = Spline2(path_dir + "lo_3D_DE.dat")
p37_r4 = Spline2(path_dir + "lo_3D_LE.dat")
p26_r4 = Spline2(path_dir + "hi_3D_LE.dat")
p15_r4 = Spline2(path_dir + "hi_3D_MD.dat")

#CoonsPatch(pS; pN; pW; pE)
```
SN_r4 = CoonsPatch(p32_r4, p76_r4, p37_r4, p26_r4)
SS_r4 = CoonsPatch(p01_r4, p45_r4, p04_r4, p15_r4)
ST_r4 = CoonsPatch(p45_r4, p76_r4, p47_r4, p56_r4)
SB_r4 = CoonsPatch(p01_r4, p32_r4, p03_r4, p12_r4)
SW_r4 = MeshPatch(surf_dir + "lo_3D_LEDM.vtk")
SE_r4 = MeshPatch(surf_dir + "hi_3D_LEDM.vtk")
print "\nRegion 4 Surfaces\n"
# Region 4: Blade inlet
r4_volume = ParametricVolume(SN_r4, SE_r4, SS_r4, SW_r4, ST_r4, SB_r4)
r4_blk = Block3D("r4-block", nni=nx, nnj=ny, nnk=nz,
    parametric_volume = r4_volume,
    fill_condition = r4_fill,
    hcell_list = [(nx/nx, ny/2, nz/2)],
    omegaz = omega)

# Region 5: Blade Radius
p03_r5 = Spline2(path_dir + "lo_3D_LK.dat")
p32_r5 = Spline2(path_dir + "ct1_3D_LK.dat")
p12_r5 = Spline2(path_dir + "hi_3D_LK.dat")
p01_r5 = Spline2(path_dir + "ct0_3D_LK.dat")
p47_r5 = Spline2(path_dir + "lo_3D_EF.dat")
p76_r5 = Spline2(path_dir + "ct1_3D_EF.dat")
p56_r5 = Spline2(path_dir + "hi_3D_EF.dat")
p45_r5 = Spline2(path_dir + "ct0_3D_EF.dat")
p04_r5 = Spline2(path_dir + "lo_3D_LE.dat")
p37_r5 = Spline2(path_dir + "lo_3D_LE.dat")
p26_r5 = Spline2(path_dir + "hi_3D_LK.dat")
p15_r5 = Spline2(path_dir + "hi_3D_LE.dat")

# Region 5: Blade Radius
SN_r5 = CoonsPatch(p32_r5, p76_r5, p37_r5, p26_r5)
SS_r5 = CoonsPatch(p01_r5, p45_r5, p04_r5, p15_r5)
ST_r5 = CoonsPatch(p45_r5, p76_r5, p47_r5, p56_r5)
SB_r5 = CoonsPatch(p01_r5, p32_r5, p03_r5, p12_r5)
SW_r5 = MeshPatch(surf_dir + "lo_3D_KFEL.vtk")
SE_r5 = MeshPatch(surf_dir + "hi_3D_KFEL.vtk")
print "\nRegion 5 Surfaces\n"
# Region 5: Blade Radius
r5_volume = ParametricVolume(SN_r5, SE_r5, SS_r5, SW_r5, ST_r5, SB_r5)
r5_blk = Block3D("r5-block", nni=nx, nnj=ny, nnk=nz,
    parametric_volume = r5_volume,
    fill_condition = r5_fill,
    hcell_list = [],
    omegaz = omega)

# Region 6: Blade Outlet
p03_r6 = Spline2(path_dir + "lo_3D_KJ.dat")
p32_r6 = Spline2(path_dir + "ct1_3D_KJ.dat")
p12_r6 = Spline2(path_dir + "hi_3D_KJ.dat")
p01_r6 = Spline2(path_dir + "ct0_3D_KJ.dat")
p47_r6 = Spline2(path_dir + "lo_3D_FG.dat")
p76_r6 = Spline2(path_dir + "ct1_3D_FG.dat")
p56_r6 = Spline2(path_dir + "hi_3D_FG.dat")
p45_r6 = Spline2(path_dir + "ct0_3D_FG.dat")
p04_r6 = Spline2(path_dir + "lo_3D_KF.dat")
```
p37_r6 = Spline2(path_dir + "lo_3D_JG.dat")
p26_r6 = Spline2(path_dir + "hi_3D_JG.dat")
p15_r6 = Spline2(path_dir + "hi_3D_KF.dat")

# CoonsPatch(pS; pN; pW; pE)
SN_r6 = CoonsPatch(p32_r6, p76_r6, p37_r6, p26_r6)
SS_r6 = CoonsPatch(p01_r6, p45_r6, p04_r6, p15_r6)
ST_r6 = CoonsPatch(p45_r6, p76_r6, p47_r6, p56_r6)
SB_r6 = CoonsPatch(p01_r6, p32_r6, p03_r6, p12_r6)
SW_r6 = MeshPatch(surf_dir + "lo_3D_JGFK.vtk")
SE_r6 = MeshPatch(surf_dir + "hi_3D_JGFK.vtk")
print "\n\nRegion 6 Surfaces\n"

# Region 6: Blade Outlet
r6_volume = ParametricVolume(SN_r6, SE_r6, SS_r6, SW_r6, ST_r6, SB_r6)

r6_blk = Block3D(label ="r6-block ", nni =nx, nnj =ny, nnk =nz,
parametric_volume = r6_volume,
fill_condition = r6_fill,
hcell_list = [(nx, ny/2, nz/2)],
omegaz = omega)

identify_block_connections()

Define boundary conditions - Stationary

if Solution == "Noz_Fill":
    # Region 1: Subsonic Nozzle Inlet
    r1_blk.bc_list[SOUTH] = SubsonicInBC(state_I)
    r1_blk.bc_list[TOP] = AdiabaticBC()
    r1_blk.bc_list[BOTTOM] = AdiabaticBC()
    r1_blk.bc_list[EAST] = AdiabaticBC()
    r1_blk.bc_list[WEST] = AdiabaticBC()
    # Region 2: Supersonic Nozzle Outlet
    r2_blk.bc_list[NORTH] = FixedPOutBC(P_II)
    r2_blk.bc_list[TOP] = AdiabaticBC()
    r2_blk.bc_list[BOTTOM] = AdiabaticBC()
    r2_blk.bc_list[EAST] = AdiabaticBC()
    r2_blk.bc_list[WEST] = AdiabaticBC()

if Solution == "Blade_Fill":
    # Region 1: Subsonic Nozzle Entry
    r1_blk.bc_list[SOUTH] = SubsonicInBC(state_I)
    r1_blk.bc_list[TOP] = AdiabaticBC()
    r1_blk.bc_list[BOTTOM] = AdiabaticBC()
    r1_blk.bc_list[EAST] = AdiabaticBC()
    r1_blk.bc_list[WEST] = AdiabaticBC()
    # Region 2: Supersonic Nozzle Outlet
    r2_blk.bc_list[NORTH] = FixedPOutBC(P_II)
    r2_blk.bc_list[TOP] = AdiabaticBC()
    r2_blk.bc_list[BOTTOM] = AdiabaticBC()
    r2_blk.bc_list[EAST] = AdiabaticBC()
    r2_blk.bc_list[WEST] = AdiabaticBC()
    # Region 3: Gap between nozzle exit and blade entry
    r3_blk.bc_list[TOP] = FixedPOutBC(P_II)
    r3_blk.bc_list[BOTTOM] = FixedPOutBC(P_II)
    # Region 4: Blade Inlet
    r4_blk.bc_list[TOP] = AdiabaticBC()
    r4_blk.bc_list[BOTTOM] = AdiabaticBC()
    r4_blk.bc_list[EAST] = AdiabaticBC()
r4_blk.bc_list[WEST] = AdiabaticBC()

# Region 5: Blade Arc
r5_blk.bc_list[TOP] = AdiabaticBC()
# Region 5: Blade Arc
r5_blk.bc_list[BOTTOM] = AdiabaticBC()
# Region 5: Blade Arc
r5_blk.bc_list[EAST] = AdiabaticBC()
# Region 5: Blade Arc
r5_blk.bc_list[WEST] = AdiabaticBC()

# Region 6: Blade Outlet
r6_blk.bc_list[NORTH] = FixedPOutBC(P_III)
# Region 6: Blade Outlet
r6_blk.bc_list[TOP] = AdiabaticBC()
# Region 6: Blade Outlet
r6_blk.bc_list[BOTTOM] = AdiabaticBC()
# Region 6: Blade Outlet
r6_blk.bc_list[EAST] = AdiabaticBC()
# Region 6: Blade Outlet
r6_blk.bc_list[WEST] = AdiabaticBC()

if Solution == "Blade_Pass":

    # Region 4: Blade Inlet
    #r4_blk.bc_list[SOUTH] = SupInBC(state_II)
    r4_blk.bc_list[SOUTH] = SubsonicInBC(state_II)
    #r4_blk.bc_list[SOUTH] = FixedPOutBC(P_II)
    #r4_blk.bc_list[SOUTH] = ExtrapolateOutBC()
    # Region 4: Blade Inlet
    #r4_blk.bc_list[TOP] = AdiabaticBC()
    # Region 4: Blade Inlet
    #r4_blk.bc_list[BOTTOM] = AdiabaticBC()
    # Region 4: Blade Inlet
    #r4_blk.bc_list[EAST] = AdiabaticBC()
    # Region 4: Blade Inlet
    #r4_blk.bc_list[WEST] = AdiabaticBC()

    # Region 5: Blade Arc
    #r5_blk.bc_list[TOP] = AdiabaticBC()
    # Region 5: Blade Arc
    #r5_blk.bc_list[BOTTOM] = AdiabaticBC()
    # Region 5: Blade Arc
    #r5_blk.bc_list[EAST] = AdiabaticBC()
    # Region 5: Blade Arc
    #r5_blk.bc_list[WEST] = AdiabaticBC()

    # Region 6: Blade Outlet
    r6_blk.bc_list[NORTH] = FixedPOutBC(P_III)
    # Region 6: Blade Outlet
    #r6_blk.bc_list[TOP] = AdiabaticBC()
    # Region 6: Blade Outlet
    #r6_blk.bc_list[BOTTOM] = AdiabaticBC()
    # Region 6: Blade Outlet
    #r6_blk.bc_list[EAST] = AdiabaticBC()
    # Region 6: Blade Outlet
    #r6_blk.bc_list[WEST] = AdiabaticBC()

""

Global Data Settings
""
gdata.dimensions = 3

gdata.axisymmetric_flag = 0

gdata.flux_calc = ADAPTIVE

if Solution == "Noz_Fill":
    t_max = 1.0e-3

if Solution == "Blade_Fill":
    t_max = (2.0 * h_noz)/(omega * bld_d_mean)

if Solution == "Blade_Pass":
    t_max = (2.0 * pi) / (omega * n_noz)

gdata.max_time = t_max

gdata.max_step = 1.0e8

gdata.dt = 1.0e-8

gdata.dt_plot = t_max/25.0

gdata.dt_history = t_max/25.0

gdata.cfl = 0.20

gdata.viscous_flag = 0
def function_geom08_Read_All(directory):
    # directory = '00_data/' + '01_Nozzles_R134/' + '20121219_R134_035_144/'
    Data = csv.reader(open(directory + 'All_Data.csv', 'r'))
    # Define Header
    for row in Data:
        Header = row
        break
    del Header[-1]
    # print Header

    # Define and Populate All Dictionary
    All = collections.defaultdict(list)
    for row in Data:
        for i in range(len(row)-1):
            try:
                All[Header[i]].append(float(row[i]))
            except:
                All[Header[i]].append(row[i])

    return All, Header
# 3D_Eilmer_Post.py

```python
#!/usr/bin/env python
#3D_mach.py
#To run in Terminal: python 3D_mach.py
#Post processing file for calculating nozzle exit velocity, mass flow, and mach

import sys, os, string
sys.path.append(os.path.expandvars("$HOME/e3bin")) # installation directory
sys.path.append("") # so that we can find user's scripts in working directory
from e3_grid import StructuredGrid #located in HOME/e3bin
from e3_flow import StructuredGridFlow #located in HOME/e3bin
from libprep3 import *
from gzip import GzipFile
from pylab import *
from scipy import *
import numpy as np
from math import *
import csv
import matplotlib.pyplot as plt
from matplotlib.pyplot import *
sys.path.append('01_Functions/)
import function_geom08_Read_All as F08

close('all')

rc('font',**{
    'family':'sans-serif',
    'sans-serif': ['Times New Roman']
})
rc("font", size=8)
fig_size_set = (7,4)
dpi_set = 300
ms_p = 1

#### End Inputs ----

####
data_path = '00_data/02_Nozzles_R245/'    #Source directory (i.e. '00_data/02_Nozzl

data_file = '20120601_R245_089_140/'  #Data run (i.e. '20120420_R245_100_125' or ")

"""Read In Conditions From All_Data ""
[All, Header] = F08.function_geom08_Read_All(data_path+data_file)

d_r_m = mean(All['d_r_m'])                  #Rotor mean diameter, m
omega = 2.0*pi*mean(All['RPM_dq'])/60.0     #Rotor radial velocity, rad/s
N_noz = mean(All['N_noz'])                  #Number of Stator Nozzles
h_s_tt = mean(All['h_s_tt'])                #Stator Nozzle Height (from lo to hi radius), m

"\nSuper Sonic Convergent
"

Time_plt = []
```

275
for i1 in range(count1):
    '''Region 0 - Nozzle Inlet'''
    fileName = sol_dir+'grid/t0000/3D_eilmer.grid.b0000.t0000.gz'
    fin = GzipFile(fileName, "rb")
    grd = StructuredGrid()
    grd.read(f=fin)
    fin.close()
    print "Read grid: ni=", grd.ni, "nj=", grd.nj, "nk=", grd.nk, "n"

    fileName = sol_dir+'flow/+times[0,i1]+/3D_eilmer.flow.b0000.'+times[0,i1]+'.gz'
    fin = GzipFile(fileName, "rb")
    soln = StructuredGridFlow()
    soln.read(fin)
    fin.close()
    ni = soln.ni; nj = soln.nj; nk = soln.nk
    print "Read solution: ni=", ni, "nj=", nj, "nk=", nk, "n"

    j = 0 #Set j to nj-1 to take values from volume exit, north face
    p_Sum_R0 = []
    T_Sum_R0 = []
    rho_Sum_R0 = []
    Mass_Flow_R0 = 0
    for i in range(ni):
        for k in range(nk):
            p0,p1,p2,p3,p4,p5,p6,p7 = grd.get_vertex_list_for_cell(i,j,k)
            # The north cell face has p3, p2, p6, p7 as corners.
            p = soln.data["p"][i][j][k]             #pressure, Pa
            T = soln.data["T[0]"][i][j][k]          #temperature, deg K
            p_Sum_R0.append(p)
            T_Sum_R0.append(T)
            surface_area = quad_area(p3, p2, p6, p7)
            vel_x = soln.data["vel.x"][i][j][k]    #velocity vector in x direction, m/s
            vel_y = soln.data["vel.y"][i][j][k]    #velocity vector in y direction, m/s
            vel_z = soln.data["vel.z"][i][j][k]    #velocity vector in z direction, m/s
            rho = soln.data["rho"][i][j][k]         #density, kg/m3
            rho_Sum_R0.append(rho)
            velocity = ((vel_x**2.0)+(vel_y**2.0)+(vel_z**2.0))**0.5
            mf = surface_area*velocity*rho
            Mass_Flow_R0 = Mass_Flow_R0 + mf        #Sum mass flow rate, kg/s

            p_ave = mean(p_Sum_R0)
            T_ave = mean(T_Sum_R0)
rho_ave = mean(rho_Sum_R0)
Mass_Flow_R0_plt.append(Mass_Flow_R0)
P_R0_plt.append(p_ave)
T_R0_plt.append(T_ave)
rho_R0_plt.append(rho_ave)

"Region 1 - Nozzle Outlet"
fileName = sol_dir+'grid/t0000/3D_eilmer.grid.b0001.t0000.gz'
fin = GzipFile(fileName, "rb")
grd = StructuredGrid()
grd.read(f=fin)
fin.close()
print "Read grid: ni=", grd.ni, "nj=", grd.nj, "nk=", grd.nk, "n"

fileName = sol_dir+'flow/'+times[0,i1]+'/3D_eilmer.flow.b0001.'+times[0,i1]+'.gz'
fin = GzipFile(fileName, "rb")
soln = StructuredGridFlow()
soln.read(fin)
fin.close()
ni = soln.ni; nj = soln.nj; nk = soln.nk
print "Read solution: ni=", ni, "nj=", nj, "nk=", nk, "n"

j = nj-1 #Set j to nj-1 to take values from volume exit, north face
a_Sum_R1 = []
p_Sum_R1 = []
T_Sum_R1 = []
rho_Sum_R1 = []
MfV_Sum_R1 = 0
MfVu_Sum_R1 = 0
MfVx_Sum_R1 = 0
MfMach_Sum_R1 = 0
Mass_Flow_R1 = 0
for i in range(ni):
for k in range(nk):
    p0,p1,p2,p3,p4,p5,p6,p7 = grd.get_vertex_list_for_cell(i,j,k)
    # The north cell face has p3, p2, p6, p7 as corners.
surface_area = quad_area(p3, p2, p6, p7)
    vel_x = soln.data["vel.x"][i][j][k]  #velocity vector in x direction, m/s
    vel_y = soln.data["vel.y"][i][j][k]  #velocity vector in y direction, m/s
    vel_z = soln.data["vel.z"][i][j][k]  #velocity vector in z direction, m/s
    rho = soln.data["rho"][i][j][k]  #density, kg/m3
    a = soln.data["a"][i][j][k]  #speed of sound, m/s
    p = soln.data["p"][i][j][k]  #pressure, Pa
    T = soln.data["T[0]"][i][j][k]  #Temperature, deg K
    velocity = ((vel_x**2.0)+(vel_y**2.0)+(vel_z**2.0))**0.5
    mf = surface_area*vel_z*rho
    #a_Sum_R1.append(a)
    Mass_Flow_R1 = Mass_Flow_R1 + mf  #Sum mass flow rate, kg/s
    MfV_Sum_R1 = MfV_Sum_R1 + mf*vel_x  #Sum mf*velocity
    MfVu_Sum_R1 = MfVu_Sum_R1 + mf*vel_y  #Sum mf*vel_y
    MfVx_Sum_R1 = MfVx_Sum_R1 + mf*vel_z  #Sum mf*vel_z
    MfMach_Sum_R1 = MfMach_Sum_R1 + mf*(velocity/a)  #Sum mach
    p_Sum_R1.append(p)  #Sum P
    T_Sum_R1.append(T)  #Sum T
V_R1 = MfV_Sum_R1/Mass_Flow_R1  #Nozzle Exit Absolute Total Velocity, m/s
Vu_R1 = MfVu_Sum_R1/Mass_Flow_R1  #Nozzle Exit Absolute Tangential Velocity, m/s
Vx_R1 = MfVx_Sum_R1/Mass_Flow_R1  #Nozzle Exit Absolute Axial Velocity, m/s
Mach_R1 = Mach_Sum_R1/Mass_Flow_R1
Work_Theory = 0.5*(Mass_Flow_R1)*((V_R1)**2.0) #Theorical Work produced in blade, W
p_ave = mean(p_Sum_R1)
T_ave = mean(T_Sum_R1)
rho_ave = mean(rho_Sum_R1)

Mass_Flow_R1_plt.append(Mass_Flow_R1)
V_R1_plt.append(V_R1)
Vu_R1_plt.append(Vu_R1)
Vx_R1_plt.append(Vx_R1)
Mach_R1_plt.append(Mach_R1)
P_R1_plt.append(p_ave)
T_R1_plt.append(T_ave)
rho_R1_plt.append(rho_ave)
Work_Theory_plt.append(Work_Theory)
Time_plt.append(i1)

print "Time Step ", times[0,i1], "Complete"
print "\nAll Time Steps Complete\n"

#### peak Statistics ####
w_max = max(Work_Theory_plt[0:-1])
w_ind = Work_Theory_plt.index(w_max)
Mass_Flow_peak_R1 = Mass_Flow_R1_plt[w_ind]
V_R1_peak = V_R1_plt[w_ind]
Vu_R1_peak = Vu_R1_plt[w_ind]
Vx_R1_peak = Vx_R1_plt[w_ind]
Mach_R1_peak = Mach_R1_plt[w_ind]

plt.figure(1, figsize=fig_size_set)
plt.plot(Time_plt,Mass_Flow_R0_plt, '--k', mec='k', ms = ms_p, mew=0.5)
plt.plot(Time_plt,Mass_Flow_R1_plt, '-k', mec='k', ms = ms_p, mew=0.5)
plt.legend((r'$m_I$', r'$m_{II}$'), 'lower right')
plt.xlabel('Time Step')
plt.ylabel('Mass Flow Rate (kg/s)')
plt.grid('on')
fig_name = sol_dir + 'Mass Flow.svg'
savefig(fig_name, dpi=dpi_set,transparent=True, format='svg')
fig_name = sol_dir + 'Mass Flow.png'
savefig(fig_name, dpi=dpi_set,transparent=True, format='png')

plt.figure(2, figsize=fig_size_set)
plt.plot(Time_plt,V_R1_plt, '-k', mec='k', ms = ms_p, mew=0.5)
plt.plot(Time_plt,Vu_R1_plt, '--k', mec='k', ms = ms_p, mew=0.5)
plt.plot(Time_plt,Vx_R1_plt, ':k', mec='k', ms = ms_p, mew=0.5)
plt.legend((r'$V_I$', r'$V_{uII}$', r'$V_{xII}$'), 'lower right')
plt.xlabel('Time Step')
plt.ylabel('Fluid Velocities (m/s)')
plt.grid('on')
fig_name = sol_dir + 'Velocity.svg'
savefig(fig_name, dpi=dpi_set,transparent=True, format='svg')
fig_name = sol_dir + 'Velocity.png'
savefig(fig_name, dpi=dpi_set,transparent=True, format='png')

plt.figure(3, figsize=fig_size_set)
plt.plot(Time_plt,Mach_R1_plt, '-k', mec='k', ms = ms_p, mew=0.5)
plt.xlabel('Time Step')
plt.ylabel('Mach Number')
plt.grid('on')
fig_name = sol_dir + 'Mach Number.png'
savefig(fig_name, dpi=dpi_set,transparent=True, format='png')
fig_name = sol_dir + 'Mach.svg'
savefig(fig_name, dpi=dpi_set, transparent=True, format='svg')
fig_name = sol_dir + 'Mach.png'
savefig(fig_name, dpi=dpi_set, transparent=True, format='png')

plt.figure(4, figsize=fig_size_set)
plt.plot(Time_plt, Work_Theory_plt, '-k', mec='k', ms = ms_p, mew=0.5)
plt.ylabel('Work (W)')
plt.grid('on')
fig_name = sol_dir + 'Work.svg'
savefig(fig_name, dpi=dpi_set, transparent=True, format='svg')
fig_name = sol_dir + 'Work.png'
savefig(fig_name, dpi=dpi_set, transparent=True, format='png')

# Nozzle pressure and temperatures
plt.figure(5, figsize=fig_size_set)
plt.plot(Time_plt, array(P_R0_plt)/1.0e6, '-k', mec='k', ms = ms_p, mew=0.5)
plt.plot(Time_plt, array(P_R1_plt)/1.0e6, '-k', mec='k', ms = ms_p, mew=0.5)
plt.legend((r'$P_I$', r'$P_{II}$'), 'lower right')
plt.xlabel('Time Step')
plt.ylabel('Pressure (MPa)')
plt.grid('on')
fig_name = sol_dir + 'Pressure.svg'
savefig(fig_name, dpi=dpi_set, transparent=True, format='svg')
fig_name = sol_dir + 'Pressure.png'
savefig(fig_name, dpi=dpi_set, transparent=True, format='png')

plt.figure(6, figsize=fig_size_set)
plt.plot(Time_plt, array(T_R0_plt)-273.15, '-k', mec='k', ms = ms_p, mew=0.5)
plt.plot(Time_plt, array(T_R1_plt)-273.15, '-k', mec='k', ms = ms_p, mew=0.5)
plt.legend((r'$T_I$', r'$T_{II}$'), 'lower right')
plt.xlabel('Time Step')
plt.ylabel('Temperature (' + r'$^\circ$' + 'C)')
plt.grid('on')
plt.title('Temperature')
fig_name = sol_dir + 'Temperature.svg'
savefig(fig_name, dpi=dpi_set, transparent=True, format='svg')
fig_name = sol_dir + 'Temperature.png'
savefig(fig_name, dpi=dpi_set, transparent=True, format='png')

plt.figure(7, figsize=fig_size_set)
plt.plot(Time_plt, rho_R0_plt, '-k', mec='k', ms = ms_p, mew=0.5)
plt.plot(Time_plt, rho_R1_plt, '-k', mec='k', ms = ms_p, mew=0.5)
plt.legend((r'$\rho_I$', r'$\rho_{II}$'), 'lower right')
plt.xlabel('Time Step')
plt.ylabel('Density' + ' (kg/m' r'$^3$' + ')')
plt.grid('on')
fig_name = sol_dir + 'Density.svg'
savefig(fig_name, dpi=dpi_set, transparent=True, format='svg')
fig_name = sol_dir + 'Density.png'
savefig(fig_name, dpi=dpi_set, transparent=True, format='png')

V_II = V_R1_plt[-1]
print "V_II:  ", V_II
# Store V_II to All Dictionary
for i in range(len(All['V_II'])):
    All['V_II'][i] = V_II
#Write Header to All_Data.csv
fw = open(data_path + data_file + 'All_Data.csv', "w")
for k in range(len(All.keys())):
    fw.write(Header[k] + ",")
fw.write("\n")

#Write Data To All_Data.csv
for i in range(len(All[Header[0]])):
    for j in range(len(Header)):
        fw.write(str(All[Header[j]][i]) + ",")
fw.write("\n")

fw.close()}
Loss_Search.py

#!/usr/bin/python
import os, sys, csv, random, time
from bisect import *
from scipy import *
import Loss_Model as LM
import matplotlib
from matplotlib.pyplot import *
sys.path.append('01_Functions/)
import function_coef_gen as F_coef
import function_geom08_Read_All as F08

### BEGIN INPUTS
-----------------------------------------------------------
### Coefficient Search Parameters
error_accept_set = 0.01              #acceptable error level
delta_error_accept = 0.02           #acceptable delta error
error_check_min_set = 100.0         #Initial error check value
search_count_set = 5                #Initial number search iterations for X_coefficients, minimum of 2 or set to 0 for single run
delta_error_stop = 2               #Number of consecutive delta error_min < error_accept before search stops
count_mode = "Fast"                 #count mode sets whether or not all data points are analysed or only the count_fast points, set to "All" or "Fast"
RPM_Lo_Limit = 1500                #Fast count Lo RPM filter
RPM_Hi_Limit = 3000                #Fast count Hi RPM filter
sound_2 = 175.0                    #Sound Speed at Rotor Exit  (R134a: 175, R245fa: 150)
Bearing_Load = 22.5                #Bearing Radial Load (Weight of Rotor), N
Bearing_Friction_Coef = 0.0005     #Bearing Coefficient of Friction (Steel:  0.0015, Ceramic: 0.005)
Bearing_Core_Diameter = 0.010      #bearing core diameter, m
Moment_Seal = 0.00000              #Seal Moment, N-m   (Steel:  0.0172, Ceramic: 0)
### Set Initial Loss Coefficients
# X_V1 = 1.00                         #Nozzle Exit Velocity Coefficient
# X_clearance_opt = 0.50              #Lo speed affect, 1.00 default, 0.74
# X_discf_opt = 0.70                  #Hi speed affect, 1.00 default, 0.50
# X_sector_opt = 0.50                 #Av speed affect, 1.00 default, 0.69
# X_partial_opt = 1.00                #Hi speed affect, 1.00 default, 0.69
# X_trail_opt = 0.30                  #Lo speed affect, 1.00 default, 0.16
# X_incidence_opt = 0.30              #Lo speed affect, 1.00 default, 0.11
# X_passage_opt = 0.30                #Lo speed affect, 1.00 default, 0.11
# E_discf_opt = 3.10                  #Al speed affect, 3.00 default, 3.59
# E_partial_opt = 3.10                #Al speed affect, 3.00 default, 2.59
# E_sector_opt = 80                   #Av speed affect, 100. default, 100
X_V1 = 1.00                         #Nozzle Exit Velocity Coefficient
X_clearance_opt = 1.00              #Lo speed affect, 1.00 default, 0.74
X_discf_opt = 0.70                  #Hi speed affect, 1.00 default, 0.50
X_sector_opt = 1.75                 #Av speed affect, 1.00 default, 0.69
X_partial_opt = 1.42                #Hi speed affect, 1.00 default, 0.69
X_trail_opt = 0.62                  #Lo speed affect, 1.00 default, 0.16
X_incidence_opt = 0.30              #Lo speed affect, 1.00 default, 0.11
X_passage_opt = 0.35                #Lo speed affect, 1.00 default, 0.11
E_discf_opt = 3.10                  #Al speed affect, 3.00 default, 3.59
E_partial_opt = 3.10                #Al speed affect, 3.00 default, 2.59
E_sector_opt = 1.00                 #Av speed affect, 100. default, 100
###Plot Setup
rc('font',*['family':sans-serif',sans-serif':[Times New Roman']])
r('font', size=10)
fig_size_set = (7.4)
u_mew = 0.45
### Define File Paths - Specified Files
```
data_path_all = ['00_data/03_Gathered/']  # data source directory
data_file_all = ['All_Data_1-2Noz_R245/']  # Import data for use in loss model calculations
```

### Define File Paths - All Files
```
directory = '00_data/
data_path_all = []
data_file_all = []
data_path_list = os.walk(directory).next()[1]
for i0 in range(len(data_path_list)):
    data_file_list = os.walk(directory+data_path_list[i0]).next()[1]
    for i1 in range(len(data_file_list)):
        if data_file_list[i1] != 'Archive' and data_path_list[i0] == '01_Nozzles_R245':
            data_path_all.append(directory + data_path_list[i0] + '/
data_file_all.append(data_file_list[i1] + '/)
```

# Loop For all Data Files
for i_path in range(len(data_path_all)):
    data_path = data_path_all[i_path]
data_file = data_file_all[i_path]
print "data_path:  %s, data_file:  %s" % (data_path, data_file)

### Geom Inputs
```
"Read In Conditions From Config File"
[All, Header] = F08.function_geom08_Read_All(data_path+data_file)
count_stable = len(All["Time_dq"])
```

### Data Inputs To Determine Fast Search Points in count_fast
```
count_fast = []
RPM = []
W_meas = []
W_meas_1noz = []
W_meas_2noz = []
RPM_1noz = []
RPM_2noz = []
```

# Separate Data into 1 and 2 noz arrays
for i in range(count_stable):
    N_noz = All["N_noz"][i]
    RPM.append(All["RPM_dq"][i])
    W_meas.append(All["W_meas_dq"][i])
    if N_noz == 1:
        W_meas_1noz.append(W_meas[-1])
        RPM_1noz.append(RPM[-1])
    if N_noz == 2:
        W_meas_2noz.append(W_meas[-1])
        RPM_2noz.append(RPM[-1])

# Determine Min and Max RPM and Max Work for 1 and 2 noz case
```
if W_meas_1noz:
    W_max_1noz = max(W_meas_1noz)
    RPM_min_1noz = min(RPM_1noz)
    RPM_max_1noz = max(RPM_1noz)
if W_meas_2noz:
    W_max_2noz = max(W_meas_2noz)
    RPM_min_2noz = min(RPM_2noz)
RPM_max_2noz = max(RPM_2noz)
for i in range(count_stable):
    N_noz = All['N_noz'][i]
    RPM.append(All['RPM_dq'][i])
    W_meas.append(All['W_meas_dq'][i])
if N_noz == 1:
    #Append count_fast for lo RPM values
    if RPM[i] < RPM_Lo_Limit:
        count_fast.append(i)
    #Append count_fast for hi RPM values
    if RPM[i] > RPM_Hi_Limit:
        count_fast.append(i)
    #Append count_fast for max RPM values
    if 0.95*W_max_1noz < W_meas[i] < 1.05*W_max_1noz:
        count_fast.append(i)
if N_noz == 2:
    #Append count_fast for lo RPM values
    if RPM[i] < RPM_Lo_Limit:
        count_fast.append(i)
    #Append count_fast for hi RPM values
    if RPM[i] > RPM_Hi_Limit:
        count_fast.append(i)
    #Append count_fast for max RPM values
    if 0.95*W_max_2noz < W_meas[i] < 1.05*W_max_2noz:
        count_fast.append(i)

#Print Header For Loss Coefficients File
print "Write Loss Coefficients File"
fw = open(data_path+data_file+"Loss_Coefficients.csv", "w")
for i in range(len(X_coef_header)):
    fw.write(str(X_coef_header[i]) +",")
fw.write("\n")

### Start Loss Model Iterations ###########################################
iteration_count = 0                     #Initialise iteration_count, number of iterations per coefficient population
delta_error_count = 0                 #Initialise delta_error_count, number of times that delta_error and error_check_min are accepted
search_count = search_count_set       #Reinitialise search_count
error_accept = error_accept_set      #Reinitialise error_accept
delta_error = 1.0                       #Reinitialise delta_error
error_check_i = 1.0                    #Reinitialise error_check_i
error_ave_i = 1.0                      #Reinitialise error_ave_i
error_lo_rpm = 0.0                      #Reinitialise error_lo_rpm
error_hi_rpm = 0.0                      #Reinitialise error_hi_rpm
error_av_rpm = 0.0
error_plot_count = [ ]; error_plot_count.append(0.0)
derror_check_min = [ ]; error_check_min.append(error_check_min_set)
derror_ave_min = [ ]; error_ave_min.append(error_check_min_set)

while delta_error_count <= delta_error_stop:
    #Trigger to stop while loop when only wanting to do one coefficient set.
    if search_count == 0:
        delta_error_count = delta_error_stop + 1
    iteration_count = iteration_count + 1
    data_index = count_fast
    #For last one increase data_index to include all points and set search_count to zero to only run for opt coefficients
if delta_error_count >= delta_error_stop:
    search_count = 0
    error_lo_rpm = 0
    error_hi_rpm = 0
    error_av_rpm = 0
    data_index = []
    for i in range(count_stable):
        data_index.append(i)
if count_mode != "Fast":
    data_index = []
    for i in range(count_stable):
        data_index.append(i)

# Increase error_accept if iteration_count goes past 10 with no improvements
if iteration_count > 3:
    delta_error_accept = delta_error_accept + 0.01
if iteration_count > 3 and delta_error <= delta_error_accept:
    error_accept = error_accept * 1.05
    iteration_count = 0
if search_count > 0:
    # Generate Coefficient Arrays For All Possible Combinations
    X_lo_a = 1.0 - min(0.25, error_check_min[-1])  # Low random coefficient range
    X_lo_b = 1.0 - (1.0 - X_lo_a) * 0.05  # Low random coefficient range
    X_hi_a = 1.0 + min(0.25, error_check_min[-1])  # High random coefficient range
    X_hi_b = 1.0 + (X_hi_a - 1.0) * 0.05  # High random coefficient range
    X_lo_i = random.uniform(X_lo_a, X_lo_b)
    X_hi_i = random.uniform(X_hi_a, X_hi_b)
    count_coef = 1e10
while count_coef > 2000:
    X_clearance_count = int(random.uniform(1, search_count))
    X_discf_count = int(random.uniform(1, search_count))
    X_sector_count = int(random.uniform(1, search_count))
    X_partial_count = int(random.uniform(1, search_count))
    X_trail_count = int(random.uniform(1, search_count))
    X_incidence_count = int(random.uniform(1, search_count))
    X_passage_count = int(random.uniform(1, search_count))
    E_discf_count = int(random.uniform(1, search_count))
    E_partial_count = int(random.uniform(1, search_count))
    # Determine Number of Coefficient Combinations, count_coef
    count_coef = 0
    for i_Xclearance in range(X_clearance_count):
        for i_Xdiscf in range(X_discf_count):
            for i_Xsector in range(X_sector_count):
                for i_Xpartial in range(X_partial_count):
                    for i_Xtrail in range(X_trail_count):
                        for i_Xincidence in range(X_incidence_count):
                            for i_Xpassage in range(X_passage_count):
                                for i_Ediscf in range(E_discf_count):
                                    for i_Epartial in range(E_partial_count):
                                        count_coef = count_coef + 1
                                        print "count_coef: ", count_coef

Generated_Coefficients = F_coef.function_coef_gen(X_lo_i, X_hi_i, search_count, X_clearance_opt,
    X_discf_opt, X_sector_opt, X_partial_opt, X_trail_opt, X_incidence_opt, X_passage_opt, E_discf_opt, E_partial_opt,
X_clearance_count, X_discf_count, X_sector_count, X_partial_count, X_trail_count, X_incidence_count, X_passage_count, E_discf_count, E_partial_count, error_lo_rpm, error_hi_rpm, error_av_rpm)

i_coef = 0
X_clearance = list(Generated_Coefficients[0])
X_discf = list(Generated_Coefficients[1])
X_sector = list(Generated_Coefficients[2])
X_partial = list(Generated_Coefficients[3])
X_trail = list(Generated_Coefficients[4])
X_incidence = list(Generated_Coefficients[5])
X_passage = list(Generated_Coefficients[6])
E_discf = list(Generated_Coefficients[7])
E_partial = list(Generated_Coefficients[8])
count_coef = int(Generated_Coefficients[9])
else:
    i_coef = 0
X_clearance = []; X_clearance.append(X_clearance_opt)
X_discf = []; X_discf.append(X_discf_opt)
X_sector = []; X_sector.append(X_sector_opt)
X_partial = []; X_partial.append(X_partial_opt)
X_trail = []; X_trail.append(X_trail_opt)
X_incidence = []; X_incidence.append(X_incidence_opt)
X_passage = []; X_passage.append(X_passage_opt)
E_discf = []; E_discf.append(E_discf_opt)
E_partial = []; E_partial.append(E_partial_opt)
count_coef = 1

for i_search in range(count_coef):
    print "delta_error_count: %.0f, i_coef: %.0f of %.0f, iteration_count: %.0f, error_accept: %.2f, error_check_min: %.2f, error_check_i: %.2f, error_ave_i: %.2f, delta_error: %.2f" % (delta_error_count, i_coef, count_coef, iteration_count, error_accept, error_check_min[1], error_check_i, error_ave_i, delta_error)

###Declare Variables###########################
omega = []; a_III = []; U = []; Vu_II = []; Vx_II = []; Wu_II = []; Wx_II = []; W_II = []; b_II = []; Vx_III = []; Wx_III = []; Wu_III = []; W_III = []; V_III = []; a_III = []; Kw = []; u_U = []; b_j = []; Reaction = []; W_noloss = []
P_total = []; P_clearance = []; P_discf = []; P_sector = []; P_partial = []; P_trail = []; P_incidence = []; P_bearing = []
P_passage = []
W_meas = []; RPM = []; mf = []; W_calc = []; ETAT_mech = []; ETAT_thrm = []; psi = []; phi = []; Ns = []; W_error = []
W_error_percent = []
W_error_percent_1noz = []; W_error_percent_2noz = []; W_meas_1noz = []; RPM_1noz = []; W_meas_2noz = []; RPM_2noz = []

###Calculate Losses##########################################################################
i0 = 0
for i0 in range(len(data_index)):
    i_data = data_index[i0]
    d_r_m = All['d_r_m'][i_data]
    V_II = All['V_II'][i_data]
    N_noz = All['N_noz'][i_data]
    noz_arc = All['noz.arc'][i_data]
    pitch = All['pitch'][i_data]
    d_r_tp = All['d_r_tp'][i_data]
    h_r_tt = All['h_r_tt'][i_data]
    a_II = All['a_II'][i_data]
    b_II = All['b_II'][i_data]
    b_III = All['b_III'][i_data]
    N_bld = All['N_bld'][i_data]
    w_r_tt = All['w_r_tt'][i_data]
R_ss = All['R_ss'][i_data]
gap = All['gap'][i_data]
tip_clear = All['tip_clear'][i_data]
l = All['l'][i_data]
chord = All['chord'][i_data]
te_r = All['te_r'][i_data]
mf = All['mf_dq'][i_data]
T_I = All['T_I_dq'][i_data]
P_I = All['P_I_dq'][i_data]
H_I = All['H_I_dq'][i_data]
Rho_I = All['Rho_I_dq'][i_data]
P_II = All['P_II_dq'][i_data]
T_III = All['T_III_dq'][i_data]
Rho_III = All['Rho_III_dq'][i_data]
v_III = All['v_III_dq'][i_data]
H_III = All['H_III_dq'][i_data]
Tq = All['Tq_dq'][i_data]
RPM.append(All['RPM_dq'][i_data])
W_meas.append(All['W_meas_dq'][i_data])

####################################################################
### Velocity Triangle, (Glassman, 1994)###############################
### Sign convention (+) in direction of blade
omega = (2.0*pi*RPM[i0])/60.0  # Rotor angular velocity, rad/s
U = 0.5*d_r_m*omega  # m/s
Vu_II = V_II*sin(a_II)  # Based on trig
Vx_II = V_II*cos(a_II)  # Based on trig
Wu_II = Vu_II - U  # Based on trig
Wx_II = Vx_II  # Based on trig
W_II = (Wu_II**2 + Wx_II**2)**0.5  # Based on trig
b_IIi = arcsin(Wu_II/W_II)  # Based on trig
Vu_III = Wu_III + U  # Based on trig
V_III = (Vu_III**2 + Vx_III**2)**0.5  # Based on trig
a_III = arctan(Vu_III/Vx_III)  # Based on trig
u_U = U/(Vu_II-Vu_III) # Blade Jet Speed Ratio
Reaction = (W_III-1)**2 - (W_II-1)**2 / (W_III-1)**2 + V_II**2  # from Figure 8-3 Glassman
W_noloss = mf*U*(Vu_II-Vu_III)  # Work, W
# print 'RPM: %.2f, W_noloss: %.2f, mf: %.2f, Vu_II: %.2f, Vu_III: %.2f, b_IIi: %.2f, U: %.2f, Vx_II: %.2f' % (RPM[i0], W_noloss, mf, Vu_II, Vu_III, b_IIi, U, Vx_II)
W_iso = mf*(H_I-H_III)  # Theoretical work from enthalpy drop, W

####################################################################
### Run Loss Model Function ####################################
Loss_Model_Outputs = LM.Loss_Model(d_r_m, V_II, N_noz, noz_arc, pitch, d_r_tp, h_r_tt, a_II, b_II, b_III, N_bld, w_r_tt, R_ss, gap, tip_clear, l, chord, te_r, mf, sound_2, Bearing_Load, Bearing_Friction_Coef, Bearing_Core_Diameter, Moment_Seal, T_I, P_I, H_I, Rho_I, P_II, T_III, Rho_III, v_III, T_IV, P_III, H_III, Tq, RPM[i0], W_meas[i0])

286
omega[i0], U[i0], Vu_II[i0], Vx_II[i0], Wu_II[i0], W_II[i0], b_II[i0],
Vx_III[i0], Wx_III[i0], Wu_III[i0], Vx_III[i0], Vu_III[i0], Vx_III[i0], a_III[i0],
u_U[i0], b_j[i0], Reaction[i0], W_noloss[i0], W_iso,
X_clearance[i_coef], X_discf[i_coef], X_sector[i_coef], X_partial[i_coef],
X_trail[i_coef], X_incidence[i_coef], X_passage[i_coef], E_discf[i_coef], E_partial[i_coef])

P_clearance.append(Loss_Model_Outputs[0])
P_discf.append(Loss_Model_Outputs[1])
P_sector.append(Loss_Model_Outputs[2])
P_partial.append(Loss_Model_Outputs[3])
P_trail.append(Loss_Model_Outputs[4])
P_incidence.append(Loss_Model_Outputs[5])
P_bearing.append(Loss_Model_Outputs[6])
P_passage.append(Loss_Model_Outputs[7])
P_total.append(Loss_Model_Outputs[8])

W_calc.append(max((W_noloss[i0]-P_total[i0]),0.0))
W_error.append(W_calc[i0]/W_meas[i0])
W_error_percent.append(W_error[i0]/W_meas[i0])
ETAT_mech.append(100.0*(W_calc[i0]/W_noloss[i0]))
ETAT_thrm.append(100.0*(W_calc[i0]/W_iso))

"""--- Turbine Performance Calcs ---"""
psi.append(W_calc[i0]/U[i0]**2.0)  # Stage loading coefficient, pp. 6-9 eqs 6.37
phi.append(Vx_II[i0]/U[i0])  # Flow coefficient, pp. 6-9 eqs 6.38
Ns.append((omega[i0]*(mf/Rho_III)**0.5)/((H_I-H_III)**0.5))  # Specific Speed

if N_noz == 1:
    W_error_percent_1noz.append(W_error_percent[-1])
    W_meas_1noz.append(W_meas[-1])
    RPM_1noz.append(RPM[-1])

if N_noz == 2:
    W_error_percent_2noz.append(W_error_percent[-1])
    W_meas_2noz.append(W_meas[-1])
    RPM_2noz.append(RPM[-1])

# Determine error
i_maxw = W_meas.index(max(W_meas))
W_max_1noz = []; RPM_min_1noz = []; RPM_max_1noz = []; W_error_percent_1_noz_RPM_min = [];
W_error_percent_1_noz_RPM_max = []; W_error_percent_1_noz_W_max = []
RPM_error_plot_1noz = []; W_error_plot_1noz = []
W_max_2noz = []; RPM_min_2noz = []; RPM_max_2noz = []; W_error_percent_2_noz_RPM_min = [];
W_error_percent_2_noz_RPM_max = []; W_error_percent_2_noz_W_max = []
RPM_error_plot_2noz = []; W_error_plot_2noz = []
if W_error_percent_1noz:
    e_coef_1noz = 1.0
    W_max_1noz = max(W_meas_1noz)
    RPM_min_1noz = min(RPM_1noz)
    RPM_max_1noz = max(RPM_1noz)
for i_1noz in range(len(W_error_percent_1noz)):
    # Append W_error_percent_1noz for lo RPM values
    if RPM_1noz[i_1noz] < RPM_Lo_Limit:
        W_error_percent_1_noz_RPM_min.append(W_error_percent_1noz[i_1noz])
        RPM_error_plot_1noz.append(RPM_1noz[i_1noz])
    # Append W_error_percent_1noz for hi RPM values
    if RPM_1noz[i_1noz] > RPM_Hi_Limit:
W_error_percent_1_noz_RPM_max.append(W_error_percent_1noz[i_1noz])
RPM_error_plot_1noz.append(RPM_1noz[i_1noz])
W_error_plot_1noz.append(W_meas_1noz[i_1noz])
#Append W_error_percent_1noz for max RPM values
if 0.95*W_max_1noz < W_meas_1noz[i_1noz] < 1.05*W_max_1noz:
    W_error_percent_1_noz_W_max.append(W_error_percent_1noz[i_1noz])
    RPM_error_plot_1noz.append(RPM_1noz[i_1noz])
    W_error_plot_1noz.append(W_meas_1noz[i_1noz])
else:
    e_coef_1noz = 0.0
    W_error_percent_1_noz_RPM_min = [0.0]
    W_error_percent_1_noz_RPM_max = [0.0]
    W_error_percent_1_noz_W_max = [0.0]
    RPM_error_plot_1noz = [0.0];
    W_error_plot_1noz = [0.0]
if W_error_percent_2noz:
    e_coef_2noz = 1.0
    W_max_2noz = max(W_meas_2noz)
    RPM_min_2noz = min(RPM_2noz)
    RPM_max_2noz = max(RPM_2noz)
for i_2noz in range(len(W_error_percent_2noz)):
    #Append W_error_percent_2noz for lo RPM values
    if RPM_2noz[i_2noz] < RPM_Lo_Limit:
        W_error_percent_2_noz_RPM_min.append(W_error_percent_2noz[i_2noz])
        RPM_error_plot_2noz.append(RPM_2noz[i_2noz])
        W_error_plot_2noz.append(W_meas_2noz[i_2noz])
    #Append W_error_percent_2noz for hi RPM values
    if RPM_2noz[i_2noz] > RPM_Hi_Limit:
        W_error_percent_2_noz_RPM_max.append(W_error_percent_2noz[i_2noz])
        RPM_error_plot_2noz.append(RPM_2noz[i_2noz])
        W_error_plot_2noz.append(W_meas_2noz[i_2noz])
    #Append W_error_percent_2noz for max RPM values
    if 0.95*W_max_2noz < W_meas_2noz[i_2noz] < 1.05*W_max_2noz:
        W_error_percent_2_noz_W_max.append(W_error_percent_2noz[i_2noz])
        RPM_error_plot_2noz.append(RPM_2noz[i_2noz])
        W_error_plot_2noz.append(W_meas_2noz[i_2noz])
else:
    e_coef_2noz = 0.0
    W_error_percent_2_noz_RPM_min = [0.0]
    W_error_percent_2_noz_RPM_max = [0.0]
    W_error_percent_2_noz_W_max = [0.0]
    RPM_error_plot_2noz = [0.0];
    W_error_plot_2noz = [0.0]
#Weighted error value to check progress of coefficient search
error_check_i = ((1.0*e_coef_1noz*average(absolute(W_error_percent_1_noz_RPM_min))
    + 1.0*e_coef_1noz*average(absolute(W_error_percent_1_noz_RPM_max))
    + 2.0*e_coef_1noz*average(absolute(W_error_percent_1_noz_W_max))))
    +(1.0*e_coef_2noz*average(absolute(W_error_percent_2_noz_RPM_min))
    + 1.0*e_coef_2noz*average(absolute(W_error_percent_2_noz_RPM_max))
    + 2.0*e_coef_2noz*average(absolute(W_error_percent_2_noz_W_max)))/((4.0*e_coef_1noz + 4.0*e_coef_2noz)
error_ave_i = average(absolute(W_error_percent))
#Write Best X Coefficients
if error_check_i <= error_check_min[-1] and error_check_i > 0:
    X_clearance_opt = X_clearance[i_coef]
    X_discf_opt = X_discf[i_coef]
    X_sector_opt = X_sector[i_coef]
\[
\begin{align*}
X_{\text{partial\_opt}} &= X_{\text{partial}[i,\text{coef}]} \\
X_{\text{trail\_opt}} &= X_{\text{trail}[i,\text{coef}]} \\
X_{\text{incidence\_opt}} &= X_{\text{incidence}[i,\text{coef}]} \\
X_{\text{passage\_opt}} &= X_{\text{passage}[i,\text{coef}]} \\
E_{\text{discf\_opt}} &= E_{\text{discf}[i,\text{coef}]} \\
E_{\text{partial\_opt}} &= E_{\text{partial}[i,\text{coef}]} \\
\end{align*}
\]

\[
\begin{align*}
\text{error\_lo\_rpm} &= ((e_{\text{coef\_1noz}}*\text{average}(W_{\text{error\_percent\_1\_noz\_RPM\_min}})) + (e_{\text{coef\_2noz}}*\text{average}(W_{\text{error\_percent\_2\_noz\_RPM\_min}})))/(e_{\text{coef\_1noz}} + e_{\text{coef\_2noz}}) \\
\text{error\_hi\_rpm} &= ((e_{\text{coef\_1noz}}*\text{average}(W_{\text{error\_percent\_1\_noz\_RPM\_max}})) + (e_{\text{coef\_2noz}}*\text{average}(W_{\text{error\_percent\_2\_noz\_RPM\_max}})))/(e_{\text{coef\_1noz}} + e_{\text{coef\_2noz}}) \\
\text{error\_av\_rpm} &= ((e_{\text{coef\_1noz}}*\text{average}(W_{\text{error\_percent\_1\_noz\_W\_max}})) + (e_{\text{coef\_2noz}}*\text{average}(W_{\text{error\_percent\_2\_noz\_W\_max}})))/(e_{\text{coef\_1noz}} + e_{\text{coef\_2noz}}) \\
\text{error\_check\_min}.append(\text{error\_check\_i}) \\
\text{error\_ave\_min}.append(\text{error\_ave\_i}) \\
\text{error\_plot\_count}.append(\text{error\_plot\_count}-1 + 1) \\
\text{print} "\text{error\_min\ set\ to: }", \text{error\_check\_min}-1 \\
\text{if error\_check\_i <= error\_check\_min}-1 \text{and error\_check\_i > 0 or delta\_error\_count == delta\_error\_stop:} \\
\#\#\#\text{Plot progress} \\
\text{fig1} = \text{figure}(1, \text{figsize=fig\_size\_set}); \text{clf()} \\
\text{rect} = \text{fig1}.patch; \text{rect.set\_facecolor('white')} \\
\text{plot(\text{error\_plot\_count}, \text{error\_check\_min}, 'o', \text{mec='k', color='k', ms = 5.0, mew=u\_mew})} \\
\text{plot(\text{error\_plot\_count}, \text{error\_ave\_min}, 'o', \text{mec='k', color='w', ms = 5.0, mew=u\_mew})} \\
\text{xlabel('Coefficient Iteration Count'); ylabel('Error, }$Xi$'); \text{grid('on')} \\
\text{xlim([0,\text{error\_plot\_count}\_\text{-1}]}) \\
\text{ylim([0.2])} \\
\text{cx} = 0.10 \\
\text{cy} = 0.10 \\
\text{text(cx, 2.0-1*cy,r'$\Phi_{\text{clearance}}$: %.2f' % (X\_clearance\_opt), \text{fontsize = 10})} \\
\text{text(cx, 2.0-2*cy,r'$\Phi_{\text{discf}}$: %.2f$' % (X\_discf\_opt), \text{fontsize = 10})} \\
\text{text(cx, 2.0-3*cy,r'$\Phi_{\text{sector}}$: %.2f$' % (X\_sector\_opt), \text{fontsize = 10})} \\
\text{text(cx, 2.0-4*cy,r'$\Phi_{\text{partial}}$: %.2f$' % (X\_partial\_opt), \text{fontsize = 10})} \\
\text{text(cx, 2.0-5*cy,r'$\Phi_{\text{trail}}$: %.2f$' % (X\_trail\_opt), \text{fontsize = 10})} \\
\text{text(cx, 2.0-6*cy,r'$\Phi_{\text{incidence}}$: %.2f$' % (X\_incidence\_opt), \text{fontsize = 10})} \\
\text{text(cx, 2.0-7*cy,r'$\Phi_{\text{passage}}$: %.2f$' % (X\_passage\_opt), \text{fontsize = 10})} \\
\text{text(cx, 2.0-8*cy,r'$E_{\text{discf}}$: %.2f$' % (E\_discf\_opt), \text{fontsize = 10})} \\
\text{text(cx, 2.0-9*cy,r'$E_{\text{partial}}$: %.2f$' % (E\_partial\_opt), \text{fontsize = 10})} \\
\text{# text(cx, 2.0-10*cy,r'$E_{\text{sector}}$: %.2f$' % (E\_sector\_opt), \text{fontsize = 10})} \\
\text{text(cx, 2.0-11*cy,r'$\Xi_{\text{check}}$: %.2f$' % (error\_check\_min\_\text{-1}), \text{fontsize = 10})} \\
\text{legend([r'$\Xi_{\text{check}}$', r'$\Xi_{\text{average}}$'], 'upper right$')} \\
\text{fig1.canvas.draw()} \\
\#\#\#\text{Plot Losses versus RPM} \\
\#\text{Plot Hi Speed Factors} \\
\text{fig2} = \text{figure}(2, \text{figsize=fig\_size\_set}); \text{clf()} \\
\text{rect} = \text{fig2}.patch; \text{rect.set\_facecolor('white')} \\
\text{ax} = \text{subplot}(2,1,1,axisbg='w', frameon=True) \\
\text{plot(\text{RPM}, \text{P\_clearance}, alpha=0)} \\
\text{ylabel('Hi Speed Power Loss (W)')} \\
\text{ax.get\_xaxis().tick\_bottom()} \\
\text{ax\text\_axes.get\_xaxis().set\_visible(False)} \\
\text{locator\_params(nbins=5)} \\
\text{grid('on')}
```python
twiny()
lns1 = plot(b_j, P_discf, 'o', mec='k', color='w', ms = 5.0, mew=u_mew, label=r'$P_{windage}$')
lns2 = plot(b_j, P_sector, 'x', mec='k', color='w', ms = 5.0, mew=u_mew, label=r'$P_{sector}$')
lns3 = plot(b_j, P_partial, '|', mec='k', color='w', ms = 5.0, mew=u_mew, label=r'$P_{partial}$')
lns4 = plot(b_j, P_bearing, '4', mec='k', color='w', ms = 5.0, mew=u_mew, label=r'$P_{bearing}$')
locator_params(nbins=5)
xlabel('U/V'); grid('on')
lns = lns1+lns2+lns3+lns4
labs = [l.get_label() for l in lns]
legend(lns, labs, loc='upper left', shadow=True)
leg = gca().get_legend()
ltext  = leg.get_texts() setp(ltext, fontsize=8)

#Plot Lo Speed Factors
ax = subplot(2,1,2, axisbg='w', frameon=True)
lns4 = plot(RPM, P_clearance, '.', mec='k', color='k', ms = 5.0, mew=u_mew, label=r'$P_{clearance}$')
lns5 = plot(RPM, P_trail, '|', mec='k', color='k', ms = 5.0, mew=u_mew, label=r'$P_{trail}$')
lns6 = plot(RPM, P_incidence, 'x', mec='k', color='k', ms = 5.0, mew=u_mew, label=r'$P_{incidence}$')
lns7 = plot(RPM, P_passage, 'd', mec='k', color='w', ms = 5.0, mew=u_mew, label=r'$P_{passage}$')
locator_params(nbins=5)
xlabel('RPM'); ylabel('Lo Speed Power Loss (W)')
lns = lns4+lns5+lns6+lns7
labs = [l.get_label() for l in lns]
legend(lns, labs, loc='upper right', shadow=True)
leg = gca().get_legend() ltext  = leg.get_texts() setp(ltext, fontsize=8) grid('on')
fig2.canvas.draw()

###Plot Works versus RPM
fig3 = figure(3, figsize=fig_size_set); clf()
rect = fig3.patch; rect.set_facecolor('white')
plot(b_j, W_calc, alpha=0);
# plot(b_j, W_noloss, 'x');
xlabel('U/V'); ylabel('Work (W)'); grid('on')
twiny()
plot(RPM, W_calc, 'o', mec='k', color='k', ms = 5.0, mew=u_mew)
plot(RPM, W_meas, 'o', mec='k', color='w', ms = 5.0, mew=u_mew)
plot(RPM_error_plot_1noz, W_error_plot_1noz, '|', mec='k', color='k', ms = 5.0, mew=u_mew)
plot(RPM_error_plot_2noz, W_error_plot_2noz, '|', mec='k', color='k', ms = 5.0, mew=u_mew)
locator_params(nbins=5)
xlabel('RPM'); ylabel('Work (W)'); grid('on')
legend((r'$W_{calc}$', r'$W_{meas}$'), 'upper left')
leg = gca().get_legend() ltext  = leg.get_texts() setp(ltext, fontsize=10); fig3.canvas.draw()

###Efficiencies versus RPM
fig4 = figure(4, figsize=fig_size_set); clf()
rect = fig4.patch; rect.set_facecolor('white')
plot(b_j, ETAT_mech, alpha=0)
xlabel('U/V'); ylabel('Efficiency (%)', r'$\eta$'); grid('on')
twiny()
plot(RPM, ETAT_mech, 'o', mec='k', color='k', ms = 5.0, mew=u_mew)
plot(RPM, ETAT_thrm, 'o', mec='k', color='w', ms = 5.0, mew=u_mew)
locator_params(nbins=5)
xlabel('RPM'); ylabel('Efficiency (%)', r'$\eta$')
```

if abs(error_check_i) <= error_accept:
    X_coef_write = [data_path, data_file, N_noz, mf, i_coef, error_check_i, W_error_percent[i_maxw],
                    X_clearance[i_coef], X_discf[i_coef], X_sector[i_coef],
                    X_partial[i_coef], X_trail[i_coef], X_incidence[i_coef], X_passage[i_coef],
                    E_discf[i_coef], E_partial[i_coef], E_trail[i_coef]]

    for i in range(len(X_coef_write)):
        fw.write(str(X_coef_write[i]) + ",")
        fw.write("\n")
        print "write coefficients for error: ", abs(error_check_i)
        i_coef = i_coef + 1
        i_search = i_search + 1

    #evaluate delta_error_count
    if len(error_check_min) < 2:
        delta_error = 1.00
    else:
        delta_error = abs(error_check_min[-1]-error_check_min[-2])/error_check_min[-1]

    if (delta_error < delta_error_accept and error_check_min[-1]<error_accept):
        delta_error_count = delta_error_count + 1
        iteration_count = 0

fig1.savefig(data_path + data_file + "Progress.png", dpi=dpi_set, format='png')
# fig1.savefig(data_path + data_file + "Progress.svg", dpi=dpi_set, format='svg')
fig2.savefig(data_path + data_file + "Losses.png", dpi=dpi_set, format='png')
# fig2.savefig(data_path + data_file + "Losses.svg", dpi=dpi_set, format='svg')
fig3.savefig(data_path + data_file + "Work.png", dpi=dpi_set, format='png')
# fig3.savefig(data_path + data_file + "Work.svg", dpi=dpi_set, format='svg')
fig4.savefig(data_path + data_file + "Efficiencies.png", dpi=dpi_set, format='png')
# fig4.savefig(data_path + data_file + "Efficiencies.svg", dpi=dpi_set, format='svg')
close('all')

fw.close()
print "$n---Simulation Results---$

##print "$n---VT Info-----------$
print i_maxw = %.3f" % (i_maxw)
print mf = %.3f kg/s" % (mf)
print RPM = %.3f" % (RPM[i_maxw])
print a_II = %.3f deg" % (a_II)
print a_III = %.3f deg" % (a_III[i_maxw])
print b_II = %.3f deg" % (b_II)
print b_III = %.3f deg" % (b_III)
print U = %.3f m/s" % (U[i_maxw])
print V_II = %.3f m/s" % (V_II)
print Vu_II = %.3f m/s" % (Vu_II[i_maxw])
print Vu_III = %.3f m/s" % (Vu_III[i_maxw])
print Vx_II = %.3f m/s" % (Vx_II[i_maxw])
print Vx_III = %.3f m/s" % (Vx_III[i_maxw])
print W_II = %.3f m/s" % (W_II[i_maxw])
print Wu_II = %.3f m/s" % (Wu_II[i_maxw])
print Wx_II = %.3f m/s" % (Wx_II[i_maxw])
print "W_III = %.3f m/s" % (W_III[i_maxw])
print "Wu_III = %.3f m/s" % (Wu_III[i_maxw])
print "Wx_III = %.3f m/s" % (Wx_III[i_maxw])
print "Reaction = %.3f" % (Reaction[i_maxw])
print "u_U = %.3f" % (u_U[i_maxw])
print "b_j = %.3f" % (b_j[i_maxw])
print "---Loss Info--------
print "P_clearance = %.3f, (%.2f)" % (P_clearance[i_maxw], P_clearance[i_maxw]/W_noloss[i_maxw])
print "P_discf = %.3f, (%.2f)" % (P_discf[i_maxw], P_discf[i_maxw]/W_noloss[i_maxw])
print "P_sector = %.3f, (%.2f)" % (P_sector[i_maxw], P_sector[i_maxw]/W_noloss[i_maxw])
print "P_partial = %.3f, (%.2f)" % (P_partial[i_maxw], P_partial[i_maxw]/W_noloss[i_maxw])
print "P_trail = %.3f, (%.2f)" % (P_trail[i_maxw], P_trail[i_maxw]/W_noloss[i_maxw])
print "P_incidence = %.3f, (%.2f)" % (P_incidence[i_maxw], P_incidence[i_maxw]/W_noloss[i_maxw])
print "P_bearing = %.3f, (%.2f)" % (P_bearing[i_maxw], P_bearing[i_maxw]/W_noloss[i_maxw])
print "P_passage = %.3f, (%.2f)" % (P_passage[i_maxw], P_passage[i_maxw]/W_noloss[i_maxw])
print "---Work Info--------
print "W_iso = %.5f W" % (W_iso)
print "W_noloss = %.5f W" % (W_noloss[i_maxw])
print "W_calc = %.5f W" % (W_calc[i_maxw])
print "W_meas = %.5f W" % (W_meas[i_maxw])
print "W_error_percent = %.2f(%%)" % (100.0*W_error_percent[i_maxw])
print "ETAT_mech = %.5f (%%)" % (ETAT_mech[i_maxw])
print "ETAT_thrm = %.5f (%%)" % (ETAT_thrm[i_maxw])
print "---Perf Info--------
print "Stage Loading Coefficient = %.2f" % (psi[i_maxw])
print "Flow Coefficient = %.2f" % (phi[i_maxw])
print "Specific Speed = %.2f" % (Ns[i_maxw])
print "---Coeff Info--------
print "X_clearance = %.2f" % (X_clearance_opt)
print "X_discf = %.2f" % (X_discf_opt)
print "X_sector = %.2f" % (X_sector_opt)
print "X_partial = %.2f" % (X_partial_opt)
print "X_trail = %.2f" % (X_trail_opt)
print "X_incidence = %.2f" % (X_incidence_opt)
print "X_passage = %.2f" % (X_passage_opt)
print "E_discf = %.2f" % (E_discf_opt)
print "E_partial_opt = %.2f" % (E_partial_opt)
# print "E_sector_opt = %.2f" % (E_sector_opt)

###Plot Values For Optimum Coefficient Values

###Plot Reaction, Velocities, Psi, and Phi versus RPM
fig5 = figure(5, fig_size_set)
subplots_adjust(wspace=0.4, hspace=0.4)
rect = fig5.patch; rect.set_facecolor('white')
subplot(2,2,1, axisbg='w', frameon=True)
plot(RPM, Reaction, 'o', mec='k', color='k', ms = 3.0, mew=0.25)
xlabel('Reaction')
ylabel('Reaction vs RPM')
title('Reaction vs RPM')

###Plot Velocities
subplot(2,2,2, axisbg='w', frameon=True)
plot(RPM, Vu_II, 'o', mec='k', color='k', ms = 5.0, mew=u_mew)
plot(RPM, Vu_II, 'o', mec='k', color='w', ms = 5.0, mew=u_mew)
plot(RPM, Vu_III, 'd', mec='grey', color='grey', ms = 5.0, mew=u_mew)
plot(RPM, Vu_III, 'd', mec='grey', color='w', ms = 5.0, mew=u_mew)
xlabel('RPM')
ylabel('Velocity (m/s)')
legend(r'SV_[u,II]', r'SV_(u,II)', r'SV_[u,III]', r'SV_(u,III)', 'upper left')
leg = gca().get_legend(); ltext = leg.get_texts()
setp(ltext, fontsize='small')
grid('on')
#Plot Stage Loading Coefficient, psi
subplot(2,2,3,axisbg='w', frameon=True)
plot(RPM, psi, 'o', mec='k', color='k', ms = 5.0, mew=u_mew)
xlabel('RPM'); ylabel('Loading Coefficient')
grid('on')
#Plot Flow Coefficient, phi
subplot(2,2,4,axisbg='w', frameon=True)
plot(RPM, phi, 'o', mec='k', color='k', ms = 5.0, mew=u_mew)
xlabel('RPM'); ylabel('Flow Coefficient')
grid('on')
fig5.savefig(data_path + data_file + "Flow.svg", dpi=dpi_set,format='svg')
fig5.savefig(data_path + data_file + "Flow.png", dpi=dpi_set,format='png')
close()

###Velocity Triangle For Selected RPM####
fig6 = figure(6,figsize=fig_size_set)
fig6.suptitle(data_file, fontsize=10)
rect = fig6.patch; rect.set_facecolor('white')
####
---
Begin Inputs
----
####
L = 10          #Graphical length for plotting nozzle
w_rotor = 20    #Graphical width for plotting rotor
####
---
End Inputs
----
####
E
End Inputs
----
####

xoff = 0.05*Vu_II[i_maxw]
yoff = 0.10*(Vx_II[i_maxw]+Vx_III[i_maxw]+w_rotor)
ptA = mat([0,0])
ptB = mat([L*sin(a_II),L*cos(a_II)])

pt1 = mat([ptB[0,0],ptB[0,1]+2*yoff])
pt2 = mat([pt1[0,0]+U[i_maxw],pt1[0,1]])
pt3 = mat([pt1[0,0]+Vu_II[i_maxw],pt1[0,1]])
pt4 = mat([pt3[0,0]+Vx_II[i_maxw],pt3[0,1]+Vu_II[i_maxw]])

ptC = mat([pt4[0,0],pt4[0,1]+2*yoff])
ptD = mat([ptC[0,0],ptC[1,1]+w_rotor])

pt5 = mat([ptD[0,0],ptD[1,1]+2*yoff])
pt6 = mat([pt5[0,0]-U[i_maxw],pt5[0,1]])
pt7 = mat([pt6[0,0]+Vu_III[i_maxw],pt6[0,1]])
pt8 = mat([pt7[0,0]+Vx_III[i_maxw],pt7[0,1]+Vu_III[i_maxw]])

t_x = min([ptA[0,0],ptB[0,0],ptC[0,0],ptD[0,0],pt1[0,0],pt2[0,0],pt3[0,0],pt4[0,0],pt5[0,0],pt6[0,0],pt7[0,0],pt8[0,0]])
t_y = min([ptA[0,1],ptB[0,1],ptC[0,1],ptD[0,1],pt1[0,1],pt2[0,1],pt3[0,1],pt4[0,1],pt5[0,1],pt6[0,1],pt7[0,1],pt8[0,1]])

#Plot Inlet Nozzle - Stator
plot(array([ptA[0,0],ptB[0,0]]),array([ptA[0,1], ptB[0,1]]),'-k')
plot(array([ptA[0,0]+8*xoff, ptB[0,0]+8*xoff]),array([ptA[0,1], ptB[0,1]]),'-k')

#Plot U1
plot(array([pt1[0,0], pt2[0,0]]),array([pt1[0,1], pt2[0,1]]),'-k')
text(mean([pt1[0,0], pt2[0,0]]),mean([pt1[0,1], pt2[0,1]]),yoff,'U')

#Plot Vu_II
plot(array([pt3[0,0], pt2[0,0]]),array([pt3[0,1], pt2[0,1]]),'-k')
text(mean([pt3[0,0], pt2[0,0]]),pt2[0,1],yoff,'Vu_II')

#Plot Vx_II
plot(array([pt3[0,0], pt4[0,0]]),array([pt3[0,1], pt4[0,1]]),'-k')
text(pt3[0,0]+xoff,mean([pt3[0,1], pt4[0,1]]),Vx_II)
# Plot V_II
plot(array([pt1[0,0], pt4[0,0]]),array([pt1[0,1], pt4[0,1]]),'-k')
text(mean([pt1[0,0], pt4[0,0]])-xoff,mean([pt1[0,1], pt4[0,1]]),'V_II')

# Plot W_II
plot(array([pt2[0,0], pt4[0,0]]),array([pt2[0,1], pt4[0,1]]),':k')
text(mean([pt2[0,0], pt4[0,0]])-xoff,mean([pt2[0,1], pt4[0,1]]),'W_II')

# Plot U2
plot(array([pt6[0,0], pt5[0,0]]),array([pt6[0,1], pt5[0,1]]),'>k')
text(mean([pt5[0,0], pt6[0,0]])-yoff,'U2')

# Plot Vu_III
plot(array([pt6[0,0], pt7[0,0]]),array([pt6[0,1], pt7[0,1]]),'--k')
text(mean([pt6[0,0], pt7[0,0]])-yoff,'Vu_III')

# Plot Vx_III
plot(array([pt7[0,0], pt8[0,0]]),array([pt7[0,1], pt8[0,1]]),'--k')
text(mean([pt7[0,0], pt8[0,0]])-2*xoff,mean([pt7[0,1], pt8[0,1]]),'Vx_III')

# Plot V_III
plot(array([pt6[0,0], pt8[0,0]]),array([pt6[0,1], pt8[0,1]]),'-k')
text(mean([pt6[0,0], pt8[0,0]])+xoff,mean([pt6[0,1], pt8[0,1]]),'V_III')

# Plot W_III
plot(array([pt5[0,0], pt8[0,0]]),array([pt5[0,1], pt8[0,1]]),':k')
text(mean([pt5[0,0], pt8[0,0]])+xoff,mean([pt5[0,1], pt8[0,1]]),'W_III')

# Add cycle information to plot
 cx = -90.0
 cy = 10.0

text(cx, 1*cy,'$SW: %.0f$ % (W_noIoss[i_maxw]), fontsize = 8)
text(cx, 2*cy,'$\alpha_{II}: %.2f^\circ$ % (a_II), fontsize = 8)
text(cx, 3*cy,'$\beta_{III}: %.2f^\circ$ % (b_III), fontsize = 8)
text(cx, 4*cy,'$SD_m: %.2f ms^{-1}$ % (d_r_m), fontsize = 8)
text(cx, 5*cy,'$SU: %.0f m/s$ % (U[i_maxw]), fontsize = 8)
text(cx, 6*cy,'$U/V: %.2f$ % (U[i_maxw]), fontsize = 8)
text(cx, 7*cy,'$SV_{\mathcal{U}_1}: %.2f$ % (Vu_II[i_maxw]), fontsize = 8)
text(cx, 8*cy,'$SV_{\mathcal{U}_1}: %.2f$ % (Vu_III[i_maxw]), fontsize = 8)
text(cx, 9*cy,'$SV_{\mathcal{U}_1}: %.2f$ % (Vu_III[i_maxw]), fontsize = 8)
text(cx, 10*cy,'$SW_{\mathcal{U}_1}: %.2f$ % (Wu_II[i_maxw]), fontsize = 8)
text(cx, 11*cy,'$SW_{\mathcal{U}_1}: %.2f$ % (Wu_III[i_maxw]), fontsize = 8)
text(cx, 12*cy,'$SW_{\mathcal{U}_1}: %.2f$ % (Wu_III[i_maxw]), fontsize = 8)
text(cx, 13*cy,'$SW_{\mathcal{U}_1}: %.2f$ % (Wu_III[i_maxw]), fontsize = 8)
text(cx, 14*cy,'$SRPM: %.2f$ % (RPM[i_maxw]), fontsize = 8)
text(cx, 15*cy,'$\beta_{\mathcal{U}_1}: %.2f$ % (b_II[i_maxw]*90.0/pi), fontsize = 8)

axis("equal")
grid('on')
xlabel('Tangential Velocity, m/s')
ylabel('Axial Velocity, m/s')
title('Velocity Triangle')

# fig6.savefig(data_path + data_file + "Velocity Triangle.svg", dpi=dpi_set, format='svg')
fig6.savefig(data_path + data_file + "Velocity Triangle.png", dpi=dpi_set, format='png')
close()
from numpy import *

def function_coef_gen(X_lo, X_hi, search_count, X_clearance, X_discf, X_sector, X_partial, X_trail, X_incidence, X_passage, E_discf, E_partial, X_clearance_count, X_discf_count, X_sector_count, X_partial_count, X_trail_count, X_incidence_count, X_passage_count, E_discf_count, E_partial_count, error_lo_rpm, error_hi_rpm, error_av_rpm):
    error_lo_rpm = 0.15*error_lo_rpm
    error_hi_rpm = 0.15*error_hi_rpm
    error_av_rpm = 0.15*error_av_rpm

    ###Clearance###
    X_clearance_range = []  #Declare X Range
    # X_clearance_range.append(X_clearance)
    X_clearance = X_clearance*(1.0+error_lo_rpm)  #Adjust X for error
    X_clearance_range_i = linspace(max(0.30,X_lo*X_clearance),min(1.25,X_hi*X_clearance), max(1,X_clearance_count))  #Create X_range
    for i in range(len(X_clearance_range_i)):
        X_clearance_range.append(X_clearance_range_i[i])

    ###Dis###
    X_discf_range = []  #Declare X Range
    # X_discf_range.append(X_discf)
    X_discf = X_discf*(1.0+error_hi_rpm)
    X_discf_range_i = linspace(max(0.30,X_lo*X_discf),min(1.25,X_hi*X_discf), max(1,X_discf_count))  #Create X_range
    for i in range(len(X_discf_range_i)):
        X_discf_range.append(X_discf_range_i[i])

    ###Sector###
    X_sector_range = []  #Declare X Range
    # X_sector_range.append(X_sector)
    # X_sector = X_sector*(1.0+average(error_av_rpm,error_hi_rpm))
    X_sector = X_sector*(1.0+error_hi_rpm)
    X_sector_range_i = linspace(max(0.30,X_lo*X_sector),min(1.75,X_hi*X_sector), max(1,X_sector_count))  #Create X_range
    for i in range(len(X_sector_range_i)):
        X_sector_range.append(X_sector_range_i[i])

    ###Partial###
    X_partial_range = []  #Declare X Range
    # X_partial_range.append(X_partial)
    X_partial = X_partial*(1.0+error_av_rpm)  #Adjust X for error
    X_partial_range_i = linspace(max(0.30,X_lo*X_partial),min(1.25,X_hi*X_partial), max(1,X_partial_count))  #Create X_range
    for i in range(len(X_partial_range_i)):
        X_partial_range.append(X_partial_range_i[i])

    ###Trail###
    X_trail_range = []  #Declare X Range
    # X_trail_range.append(X_trail)
    X_trail = X_trail*(1.0+error_lo_rpm)
    X_trail_range_i = linspace(max(0.30,X_lo*X_trail),min(1.25,X_hi*X_trail), max(1,X_trail_count))  #Create X_range
    for i in range(len(X_trail_range_i)):
        X_trail_range.append(X_trail_range_i[i])
### Incidence ###

X_incidence_range = []  # Declare X Range
X_incidence = X_incidence*(1.0+error_lo_rpm)
X_incidence_range_i = linspace(max(0.30,X_lo*X_incidence),min(1.25,X_hi*X_incidence), max(1,X_incidence_count))  # Create X_range
for i in range(len(X_incidence_range_i)):
    X_incidence_range.append(X_incidence_range_i[i])

### Passage ###

X_passage_range = []  # Declare X Range
X_passage = X_passage*(1.0+error_lo_rpm)
X_passage_range_i = linspace(max(0.25,X_lo*X_passage),min(1.25,X_hi*X_passage), max(1,X_passage_count))  # Create X_range
for i in range(len(X_passage_range_i)):
    X_passage_range.append(X_passage_range_i[i])

### E Discf ###

E_discf_range = []  # Declare X Range
E_discf_range_i = linspace(2.90,3.10, max(1,E_discf_count))  # Create X_range
for i in range(len(E_discf_range_i)):
    E_discf_range.append(E_discf_range_i[i])

### E Partial ###

E_partial_range = []  # Declare X Range
E_partial_range_i = linspace(2.90,3.10, max(1,E_partial_count))  # Create X_range
for i in range(len(E_partial_range_i)):
    E_partial_range.append(E_partial_range_i[i])

### E Sector ###

E_sector_range = []  # Declare X Range
E_sector = E_sector*(1.0+error_av_rpm)
E_sector_range_i = linspace(X_lo*E_sector,X_hi*E_sector, max(1,E_sector_count))  # Create X_range
for i in range(len(E_sector_range_i)):
    E_sector_range.append(E_sector_range_i[i])

print "X_hi: ", X_hi
print "X_lo: ", X_lo
print "error_lo_rpm: ", error_lo_rpm
print "error_hi_rpm: ", error_hi_rpm
print "X_clearance: ", X_clearance_range
print "X_discf: ", X_discf_range
print "X_sector: ", X_sector_range
print "X_partial: ", X_partial_range
print "X_trail: ", X_trail_range
print "X_incidence: ", X_incidence_range
print "X_passage: ", X_passage_range
print "E_discf: ", E_discf_range
print "E_partial: ", E_partial_range
print "E_sector: ", E_sector_range
print 
X_clearance_all = []; X_discf_all = []; X_sector_all = []; X_partial_all = []; X_trail_all = []; X_incidence_all = []
X_passage_all = []; E_discf_all = []; E_partial_all = [];
count_coef = 0

# Create arrays that allow for searching all possible combinations
for i_Xclearance in range(len(X_clearance_range)):
    for i_Xdiscf in range(len(X_discf_range)):
        for i_Xsector in range(len(X_sector_range)):
            for i_Xpartial in range(len(X_partial_range)):
                for i_Xtrail in range(len(X_trail_range)):
                    for i_Xincidence in range(len(X_incidence_range)):
                        for i_Xpassage in range(len(X_passage_range)):
                            for i_Ediscf in range(len(E_discf_range)):
                                for i_Epartial in range(len(E_partial_range)):
                                    # for i_Esector in range(len(E_sector_range)):
                                        X_clearance_all.append(X_clearance_range[i_Xclearance])
                                        X_discf_all.append(X_discf_range[i_Xdiscf])
                                        X_sector_all.append(X_sector_range[i_Xsector])
                                        X_partial_all.append(X_partial_range[i_Xpartial])
                                        X_trail_all.append(X_trail_range[i_Xtrail])
                                        X_incidence_all.append(X_incidence_range[i_Xincidence])
                                        X_passage_all.append(X_passage_range[i_Xpassage])
                                        E_discf_all.append(E_discf_range[i_Ediscf])
                                        E_partial_all.append(E_partial_range[i_Epartial])
                                        # E_sector_all.append(E_sector_range[i_Esector])
                                        count_coef = count_coef + 1
                                        return [X_clearance_all, X_discf_all, X_sector_all, X_partial_all, X_trail_all, X_incidence_all, X_passage_all, E_discf_all, E_partial_all, count_coef]
Loss_Model.py

#!/usr/bin/python
#Indent = 4 spaces
import os, sys, csv, random, time
from bisect import *
from scipy import *
from math import pi
from matplotlib.pyplot import *
sys.path.append('01_Functions/')
import function_passage as F_pas
# '''
# References For Loss Calculations
# '''

def Loss_Model(d_r_m, V_II, N_noz, noz_arc, pitch, d_r_tp, h_r_tt, a_II, b_II, b_III, N_bld, w_r_tt, R_ss, gap, tip_clear, l, chord, te_r, mf, sound_2, Bearing_Load, Bearing_Friction_Coef, Bearing_Core_Diameter, Moment_Seal, T_I, P_I, H_I, Rho_I, P_II, T_I, P_II, H_II, Tq, RPM, W_meas, omega, U, Vu_II, Vx_II, Wu_II, Vx_II, W_II, b_IIi, Vx_III, Wx_III, W_III, Vu_III, V_III, a_III, u_U, b_j, Reaction, W_noloss, W_iso, X_clearance, X_discf, X_sector, X_partial, X_trail, X_incidence, X_passage, E_discf, E_partial):

    # Tip Clearance Losses (Glassman, 1994 Figure 8-3) # Tip clearance as a function of blade height
    tip_ratio = tip_clear / h_r_tt
    m_clear = mat([[1.000, 1.000, 1.000, 0.000], [-0.023, -0.002, 0.983, 0.01], [-0.046, -0.005, 0.966, 0.02], [-0.069, -0.008, 0.949, 0.03], [-0.092, -0.011, 0.932, 0.04], [-0.115, -0.014, 0.916, 0.05], [-0.138, -0.017, 0.889, 0.06], [-0.161, -0.020, 0.882, 0.07], [-0.184, -0.023, 0.865, 0.08], [-0.207, -0.026, 0.849, 0.09], [-0.230, -0.029, 0.832, 0.10], [-0.235, -0.044, 0.748, 0.15], [-0.260, -0.058, 0.644, 0.20], [-0.285, -0.073, 0.580, 0.25], [-0.300, -0.088, 0.496, 0.30], [-0.305, -0.102, 0.412, 0.35], [-0.320, -0.117, 0.328, 0.40]])
    tip_ratio_index_hi = bisect_right(m_clear[:,3], tip_ratio)
    tip_ratio_index_lo = tip_ratio_index_hi - 1
    tip_ratio_lo = m_clear[tip_ratio_index_lo,3]
    tip_ratio_hi = m_clear[tip_ratio_index_hi,3]
    A_lo = m_clear[tip_ratio_index_lo,0]
    B_lo = m_clear[tip_ratio_index_lo,1]

    # Bilinear Interpolation Procedure to Calculate K_clearance
    tip_ratio_index_hi = bisect_right(m_clear[:,3], tip_ratio)
    tip_ratio_index_lo = tip_ratio_index_hi - 1
    tip_ratio_lo = m_clear[tip_ratio_index_lo,3]
    tip_ratio_hi = m_clear[tip_ratio_index_hi,3]
    A_lo = m_clear[tip_ratio_index_lo,0]
    B_lo = m_clear[tip_ratio_index_lo,1]
C_lo = m_clear[tip_ratio_index_lo,2]
K_clearance_lo = A_lo*Reaction**2.0 + B_lo*Reaction + C_lo
A_hi = m_clear[tip_ratio_index_hi,0]
B_hi = m_clear[tip_ratio_index_hi,1]
C_hi = m_clear[tip_ratio_index_hi,2]
K_clearance_hi = A_hi*Reaction**2.0 + B_hi*Reaction + C_hi
K_clearance = \frac{(tip\_ratio\_hi - tip\_ratio)/(tip\_ratio\_hi - tip\_ratio\_lo) \cdot K\_clearance\_hi + ((tip\_ratio - tip\_ratio\_lo)/(tip\_ratio\_hi - tip\_ratio\_lo)) \cdot K\_clearance\_lo}{(1.0-K\_clearance) \cdot W\_no\_loss} #W
mf\_leak = mf*(P\_clearance/W\_iso)

#Disc Friction Losses (Roelke 1994 and Augnier 2006)################################
#based on no-through flow, Roelke: eqs 8-9, 8-10, 8-12, 8-14, 8-16
#based on no-through flow, Augnier: eqs 4-109 to 4-114
if omega > 0.0:
#Windage Disc Friction Losses
s = gap
a = d_r_tp/2.0
s_a = s/a
Re\_disc = (omega*(a**2.0)*Rho_III)/v_III #eq 8-10
Cmo\_I = (2.0*pi) / (s_a*Re\_disc) #eq 8-9, Flow Regime I: Laminar, Small Clearance
Cmo\_II = (3.70*(s_a**0.10)) / Re\_disc**0.5 #eq 8-12, Flow Regime II: Laminar, Large Clearance
Cmo\_III = 0.080 / ((s_a*(1.0/6.0)) * (Re\_disc**0.25)) #eq 8-14, Flow Regime III: Turbulent, Small Clearance
Cmo\_IV = (0.1020 * (s_a**0.10)) / (Re\_disc**0.2) #eq 8-16, Flow Regime IV: Turbulent, Large Clearance
Cmo = max([Cmo\_I,Cmo\_II,Cmo\_III,Cmo\_IV])
P\_windage = X\_discf*(Cmo*Rho_III*(omega**E_di\_scf)*(a**5.0)/2.0) #eq 8-7, Roelke, R.J.W
P\_windage\_coupling = (Cmo*Rho_III*(omega**3.0)*(0.05**5.0)/2.0) #Magnetic Coupling
P\_discf = P\_windage + P\_gap + P\_windage\_coupling
else:
P\_discf = 0.0

#Partial Admission Sector Losses (Glassman, 1994)################################
#Ks = (1.0 - (pitch/3*active\_arc))*(E\_sector**active\_fraction) #Rotor velocity coefficient for sector loss, eq 8-24 (Roelke modified version b)
#Ks = (1.0 - (E\_sector**active\_fraction*pitch/(active\_arc))) #Rotor velocity coefficient for sector loss, eq 8-24 (Roelke modified version a)
#Ks = (1.0 - (pitch/3.0*active\_arc)) #Rotor velocity coefficient for sector loss, eq 8-24 (Roelke)
#Ks = (1.0 - (N\_noz*pitch/3.0*active\_arc)) #Rotor velocity coefficient for sector loss, eq 8-24 (Roelke/Varma)
Kw = abs(W\_III)/abs(W\_II) #Rotor relative-velocity ratio for, W2/W1
K\_sector = U\_W\_II**sin(b\_II)*2.0*(K\_sector + (1.0+Kw*Ks)) #Roelke, eq 8-28

# K\_sector = E\_sector * N\_noz * chord * U * W\_II * 0.95 / (d\_r\_m * active\_fraction)
# P_sector = X_sector*(K_sector*(mf-mf_leak))
P_sector = X_sector*(W_noloss-K_sector*(mf-mf_leak))

# print ""# print "P_sector: ", P_sector

#Partial Admission Pumping Losses
P_partial = X_partial * 3.63 * Rho_III * (U**E_partial) * (h_r_tt**1.5) * d_r_m*(1.0-active_fraction)   #eq 8-23, Roelke

#Profile Losses of trailing edge (Japikse and Baines figure 6.62 and eq 6.52)####
x = te_r/w_r_tt
K_trail_Imp = 0.275*x**2.0 + 0.080*x
K_trail_Axi = 0.478*x**2.0 + 0.158*x
K_trail = K_trail_Imp + ((b_IIi/b_III)**2) * (K_trail_Axi - K_trail_Imp)
P_trail = X_trail*0.5*(mf-mf_leak)*K_trail*(W_III**2.0)

#Incidence Loss (Glassman, 1994)########################
incidence_opt = 6.0
incidence_opt = 6.0*(pi/180)                                                #optimum incidence from Roelke, page 245 chpt 8
incidence_opt = 6.0*(pi/180) #optimum incidence from Roelke, page 245 chpt 8
K_incidence = ((W_II**2.0)/2.0) * (1.0 - cos(incidence-incidence_opt)**n)   #eq 8-34 in J/kg
P_incidence = X_incidence*((mf-mf_leak)*K_incidence) #W

#Profile Losses and Secondary Losses
#Aungier, chpt 4
f_dc = 0.0334  #Correlation for tip clearance to chord
B_1 = 0.339037 #Blade inlet metal angle in tangential reference
b1s = pi/2.0 - b_IIi #Gas relative inlet angle in tangential reference
b2s = pi/2.0 - (b_III) #Gas relative outlet angle in tangential reference
b_mean = pi/2.0 - arctan((1.0/(tan(b1s + b2s)))/2.0) #mean flow angle, Aungier 4-79
F_ar = 2.0 * ((1/tan(b1s)) - (1/tan(b2s))) #Aspect Ratio Correction, Aungier 4-81
C_L = 2.0 * ((1/tan(b1s)) - (1/tan(b2s))) #Lift coefficient, Aungier 4-77
Z = (C_L**2.0) * (sin(b2s)**2.0) / (sin(b_mean)**3.0) #Ainley loading factor, 4-78
Re_c = Rho_III * W_III * chord / v_III #Blade chord Reynolds number, Aungier 4-72
Kre = (log10(500000.0)/log10(Re_c))**2.58 #Reynolds correction for turbulent, Aungier 4-74
Kp = 1.0 #Compressibility correction, Aungier 4-63

K_secondary_pre = f_dc * F_ar * Z * sin(b2s)/sin(B_1) #Prelim Secondary Loss

Ks = 1.0 - (1.0-Kp)*(chord/h_r_tt)**2.0 / (1.0+(l/h_r_tt)**2.0) #Modified compressibility factor, Aungier 4-83 (Note that this should be modified if using blades who's axial chord project is different from the chord

K_secondary = Kre * Ks * sqrt((K_secondary_pre**2.0)/(1.0+7.5*K_secondary_pre**2.0)) #Secondary loss coefficient, Aungier 4-82
P_{secondary} = 0.5*(W_{III}^{2.0}) * K_{secondary} * (mf-mf_{leak}) \quad \#\text{Secondary power loss}

#Profile Passage Losses
active_blds = (noz_arc*N_{noz}) / pitch
K_{profile} = F_{pas}.function_passage(w_r_{tt}, h_r_{tt}, W_{II}, W_{III}, R_{ss}, l, Rho_{III}, v_{III}, active_blds)
P_{profile} = K_{profile}*(mf-mf_{leak})
P_{passage} = X_{passage}*(P_{profile}+P_{secondary})

#Deep Groove Ball Bearing Losses
#From SKF website. Online Calculator for bearing 61802-2RS1
Moment_bearing = (0.5*Bearing_Friction_Coef*Bearing_Load*Bearing_Core_Diameter)+Moment_Seal
P_{bearing} = Moment_bearing*omega*0.1

#Sum of Losses in W
P_{total} = P_{clearance}+P_{discf}+P_{sector}+P_{partial}+P_{trail}+P_{incidence}+P_{bearing}+P_{passage}

return P_{clearance}, P_{discf}, P_{sector}, P_{partial}, P_{trail}, P_{incidence}, P_{bearing}, P_{passage}, P_{total}
import os, sys
import time
from numpy import *

def function_passage(w_r_tt, h_r_tt, W_II, W_III, R_ss, l, Rho_III, v_III, active_blds):
    
    """
    FUNCTION TO CALCULATE THE PASSAGE LOSS BASED ON THE CORRELATION PROPOSED BY
    MUSGRAVE, 1980 AND RODGERS 1987."
    ###Inputs
    Cf_Turbulent_1 = 1e-3
    Cf_Turbulent_2 = 1e-1
    tol1 = 0.0001
    tol2 = 0.0001
    k_friction = 1e-7
    #Based on "Surface Roughness Prediction of 6061-T6 Aluminium Alloy Machining Using
    Hossei,K.A.,
    
    #SKIN FRICTION MODEL:
    Lh = 1
    Dh = ((2.0 * w_r_tt*h_r_tt) / (w_r_tt + h_r_tt))*active_blds
    Re_passage = Rho_III*W_II*Dh/v_III
    
    if(Re_passage < 2100.0):
        #LAMINAR REGIME (FANNING FRICTION FACTOR):
        Cf = 16.0/Re_passage
    elif(Re_passage >= 2100.0 and Re_passage <= 4000.0):
        #TRANSITION REGIME: WEIGHTING FUNCTION BETWEEN LAMINAR AND TURBULENT REGIME.
        #TURBULENT REGIME: Colebrook-White equation estimates the (dimensionless) Darcy-Weisbach
        #friction factor f for fluid flows in filled pipes. For transition regime of flow,
        #in which the friction factor varies with both R and e/D, the equation universally adopted
        #is due to Colebrook and White (1937).
        #Cf IN TURBULENT REGIME IS CALCULATED MAKING USE OF THE BISECTION METHOD FOR
        Colebrook-White EQUATION:
        #FANNING FRICTION FACTOR:
        Cf_Laminar = 16.0/Re_passage
        #LIMITS OF THE SOLUTION:
        Cf_Turbulent = BISECTION_METHOD(Cf_Turbulent_1, Cf_Turbulent_2, tol1, tol2, k_friction, Dh,
        Re_passage, COLEBROOK_WHITE_FUNCTION_FANNING)
        #WEIGHTING FUNCTION:
        laminar_turbulent_percentage = (Re_passage - 2100.0)/(4000.0 - 2100.0)
        Cf = (1.0 - laminar_turbulent_percentage) * Cf_Laminar + laminar_turbulent_percentage * Cf_Turbulent
    else:
        #FULL TURBULENT REGIME:
        Cf = BISECTION_METHOD(Cf_Turbulent_1, Cf_Turbulent_2, tol1, tol2, k_friction, Dh, Re_passage,
        COLEBROOK_WHITE_FUNCTION_FANNING)
    
    #CALCULATING THE SKIN FRICTION COEFFICIENT FOR CURVED PIPES AS PER:
    # Cf = Cf * (1.0 + 0.075 * pow(Re_passage,(1.0/4.0)) * sqrt(Dh/(R_ss)))
    Cfc_munson = 0.2*Cf  
    #Based on Munson table 8.2 for loss coefficients for pipe components, flanged 180 bend
    
    #CALCULATING THE MEAN RELATIVE VELOCITY THROUGH THE ROTOR PASSAGE AS IN:
    #Coppage et al., "Study of Supersonic Radial Compressor for Refrigeration and Pressurization Systems”, WADC
    mean_relative_velocity = (abs(W_II) + abs(W_III))/2.0

302
#THE ENTHALPY DROP DUE TO FRICTION CAN BE CALCULATED THROUGH THE FOLLOWING
EQUATION:

\[
\text{friction\_loss} = C_f \cdot \mu_s \cdot L_h/D_h \cdot \left(\frac{\text{mean\_relative\_velocity}}{2.0}\right)^2
\]

#Munson, page 508 for \( h_L = f(l/D_h)*V^2/2g \)

\[
\text{passage\_loss} = \text{friction\_loss}
\]

return passage\_loss

def COLEBROOK_WHITE_FUNCTION_FANNING(Cf_Colebrook, k_friction, Dh, Re_passage):

"""FUNCTION TO SET UP THE COLEBROOK-WHITE FUNCTION IN THE PROGRAM."""

lhs = 1.0/sqrt(Cf_Colebrook)

rhs = -4.0 * math.log10((k_friction/Dh)/3.7+(1.256/(Re_passage * sqrt(Cf_Colebrook))))

result = lhs - rhs

return result

def BISSECTION_METHOD(xa_bissection, xb_bissection, tol1, tol2, k_friction, Dh, Re_passage, function_to_evaluate):

"""FUNCTION TO CALCULATE THE ZEROS OF A GIVEN FUNCTION THROUGH THE BISECTION
METHOD."""

if(function_to_evaluate(xa_bissection, k_friction, Dh, Re_passage) * function_to_evaluate(xb_bissection, k_friction, Dh, Re_passage) <= 0.0):
    #IF WE HAVE A SOLUTION: PROCEED IMPROVING THE GUESS.
    while 1:
        x_zero = 0.5 * (xa_bissection + xb_bissection)
        if(function_to_evaluate(x_zero, k_friction, Dh, Re_passage) * function_to_evaluate(xa_bissection, k_friction, Dh, Re_passage) < 0.0):
            xb_bissection = x_zero
        else:
            xa_bissection = x_zero

        if(math.fabs(xb_bissection - xa_bissection) < tol1 or
           math.fabs(function_to_evaluate(x_zero, k_friction, Dh, Re_passage)) < tol2):
            break

else:
    print "INITIAL GUESS FOR BISECTION METHOD FOR EQUATION DID NOT BRACKET A SOLUTION"
    x_zero = 0.0

return x_zero
Appendix C. Small Impulse Turbine Experimental Results

R134a, Single Nozzle, 0.035kg/s, 144°C

![Graph showing mass flow rate over time](image_url)
Average $P_{III}$: 0.65

Average $P_{I}$: 1.53
R134a, Single Nozzle, 0.038kg/s, 140°C
R134a, Single Nozzle, 0.043kg/s, 140°C
R134a, Single Nozzle, 0.036kg/s, 140°C
R245fa, Single Nozzle, 0.044 kg/s, 140°C
R245fa, Single Nozzle, 0.044kg/s, 140°C
R245fa, Single Nozzle, 0.040kg/s, 140°C
R245fa, Single Nozzle, 0.041 kg/s, 140°C
R245fa, Single Nozzle, 0.038kg/s, 140°C
R245fa, Single Nozzle, 0.038kg/s, 140°C
R245fa, Single Nozzle, 0.035kg/s, 140°C

Average mf: 0.035

Average P_L: 1.45
Average P_H: 0.41
Appendix D. ORC Test Rig Operating Procedures

The laboratory test facility employed is a condensing Rankine cycle consisting of a pump, evaporator, turbine, regenerator and a condenser. The basic layout of the experimental setup is shown in Figure 32.

Figure 94. Turbine Test Loop

The loop is designed for the following specifications listed in

<table>
<thead>
<tr>
<th>Value</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working Fluids</td>
<td>HCFC Refrigerants (R134a, R245fa)</td>
</tr>
<tr>
<td>Working Fluid Mass Flow Rate</td>
<td>0 – 0.10 kg/s</td>
</tr>
<tr>
<td>Maximum Evaporator Pressure</td>
<td>2,000 kPa</td>
</tr>
<tr>
<td>Hot Oil Maximum Inlet Temperature</td>
<td>180 °C</td>
</tr>
<tr>
<td>Maximum Heat Addition</td>
<td>20 kW</td>
</tr>
<tr>
<td>Cooling Water Inlet Temperature</td>
<td>20-25 °C</td>
</tr>
<tr>
<td>Maximum Expander RPM</td>
<td>10,000 RPM</td>
</tr>
<tr>
<td>Maximum Expander Torque</td>
<td>22 N-m</td>
</tr>
<tr>
<td>Maximum Turbine Outlet Pressure</td>
<td>1,000 kPa</td>
</tr>
</tbody>
</table>

Variable Test Matrix
The turbine is to be tested at multiple test sets consisting of unique combinations of varied mass flow rate (mf), turbine inlet temperature ($T_s$), number of nozzles, and the rotational speed (RPM). For each test set the RPM should be varied from high to low speed achieving 10 different rotational speeds and this should be done two times in each recorded data set. Each speed should be held for 2 min. An example plot of RPM versus time for a test set is illustrated in Figure 95.

![Example plot of RPM versus Time](image)

Figure 95. Example plot of RPM versus Time

The first test set will be conducted using a two nozzle configuration in the turbine. The second set of data will be collected testing the turbine in a single nozzle configuration.

**Required Personal Protective Equipment:**
- Safety glasses at all time
- Safety gloves when handling refrigerants
- Ear protection when running turbine

**Turbine Preparation**
1. Have the turbine disassembled.
2. Clean parts and check parts for wear. Record any wear and determine if wear is acceptable or if parts need to be replaced.
3. Replace bearings in the housing. Use SKF P/N 61802-2RS1.
4. Prepare the nozzle block. First wipe the mating surfaces of the top and bottom nozzle blocks with Loctite 7649 Primer. Allow primer to dry for 5 minutes and do not wipe surfaces after the primer has dried.
5. Apply a very thin even layer of Loctite 510 Flange Sealant on to the primed surfaces of the top and bottom nozzle block. Take care not to get sealant into the flow path of the nozzle.

6. Fit top and bottom nozzle blocks together and fasten using M5 hex head bolts and measure torque with torque wrench to 2.6 N-m. When fitting together hold the block up to the light and look make sure that no light can be seen through the mating surface. If light can be seen, then the blocks are not fitted properly.

7. On the discharge end of the nozzle block prepare the nozzle block o-ring. Use Molykotesilicon o-ring lubricant to coat the o-ring groove.

8. Fit o-ring to nozzle block o-ring groove. Use SIL P/N 151s.

9. Fit nozzle block into the housing. Fasten block into housing using 8 x M5 bolts and measure torque with torque wrench to 2.6 N-m.

10. Prepare the nozzle plate o-ring groove for o-ring using Molykote silicon o-ring lubricant to coat the o-ring groove.

11. Fit o-ring to nozzle plate o-ring groove. Use SIL P/N 159s.

12. Fit the nozzle plate to the housing. Fasten plate to housing using 10 x M5 bolts and measure torque with torque wrench to 2.6 N-m.

13. Prepare Housing Ring (QGECE18_04) o-ring grooves for o-rings using Molykote silicon o-ring lubricant to coat the o-ring groove.

14. Fit o-rings to the Housing Ring (QGECE18_04). Using SIL P/N 173s.

15. Align the Housing Nozzle Side (QGECE18_02) to Housing Ring (QGECE18_04).

16. Install the shaft/rotor assembly (QGECE18_01 and QGECE18_15 press fitted together). Place the bearing seat of the shaft into the bearing in the Housing Nozzle Side (QGECE18_02). Take extra care not to damage any of the blades.

17. Fit the Housing Exhaust Side (QGECE18_03). Slide the housing into place so that bearing seat on the shaft inserts into the bearing. Make sure the Housing is oriented so the exhaust port is at the bottom.

18. Fasten the Housing together with the 11 x M12 all thread bolts and measure torque with torque wrench to 29.7 N-m.

**Pressure Test Loop**

1. Before the loop is to be operated, the loop should be pressure tested using nitrogen.
   
   a. Note: When using nitrogen to pressure test the loop ensure that the maximum outlet pressure from the nitrogen bottle is set to 2,000kPa. It is imperative that the regulator on the nitrogen bottle be set to no more than 2,000kPa to avoid...
overpressure of the loop as nitrogen bottles are supplied with an initial pressure of 20,000kPa.

2. Prior to pressure testing, screw in pressure relief fully to ensure that it doesn’t release during pressure test.

3. Ensure the loop is evacuated (no gas in loop). To be sure it is evacuated open the bleed valves at the pump inlet and outlet and bleed off any pressure inside the loop.

4. Perform low pressure test on the entire loop.
   a. Open valves at the turbine inlet and outlet.
   b. Connect the N2 source with ¼” charging hoses at the fill ports at the pump inlet and outlet (points A and B on Figure 32). Ensure that the allowable working pressure for the charging hoses is at least two times greater than the maximum pressure of the pressure test.
   c. Pressurize the loop in the following sequence.
      i. 300kPa for 10 min. Use leak detector to try and find leaks.
      ii. 700kPa for 10 min. Use leak detector to try and find leaks.
      iii. 1,000kPa for 10 min. Use leak detector to try and find leaks.
      iv. Maintain 1,000kPa and observe pressure after 1 hr. The pressure should hold stable. If pressure has dropped by more than 100kPa, then identify and repair leaks.

5. Perform high pressure test on loop between the pump outlet and the valve at the turbine inlet. Turbine has to be isolated for high pressure test because the magnetic coupling is only rated to 1,600kPa.
   a. Close the valves at the turbine inlet and outlet.
   b. Bleed off pressure in the turbine using the bleed valve at the turbine exit.
   c. Pressurize the loop in the following sequence.
      i. 1,500kPa for 10 min. Use leak detector to try and find leaks.
      ii. 2,000kPa for 10 min. Use leak detector to try and find leaks.
      iii. Maintain 2,000kPa and observe pressure after 1 hr. The pressure should hold stable. If pressure has dropped by more than 100kPa, then identify and repair leaks.

6. Set pressure relief valve (PRV) to 2,000kPa.
a. With pressure in the loop at 2,000kPa, begin to unscrew the PRV until the valve begins to allow gas to release. Stop immediately when the valve begins to release gas.

b. Bleed off pressure in the loop using the bleed valves at the pump inlet and outlet until the PRV closes.

c. Verify the PRV will release pressure at the required pressure of 2,000kPa. Apply pressure slowly to the loop with nitrogen until the PRV releases. Take note of the pressure. If the pressure is between 1,900kPa and 2,000kPa then it is deemed acceptable. If it is not continue the process of setting the PRV until an acceptable PRV release pressure is attained.

**Pull Vacuum On Loop**

1. Ensure the loop is depressurised (no gas in loop). To be sure it is depressurised open the bleed valves at the pump inlet and outlet and bleed off pressure inside the loop.

2. Install ¼" charging hoses at the fill ports at the pump inlet and outlet. Use a pressure gauge manifold to manifold the two charging lines into a single line.

3. Connect digital vacuum meter inline between the pressure gauge manifold and the vacuum pump using T-fitting.

4. Check to make sure vacuum pump has oil. If the vacuum pump requires oil fill using Javac P/N VC2063 vacuum pump oil for use with Vacuum pump CC-81.

5. Turn on the vacuum pump and monitor the vacuum on the digital vacuum meter. The vacuum reading should begin to show a vacuum after several minutes. If it does not then there is a leak somewhere in the loop.

6. Pull vacuum on the loop for 12 hours. A good vacuum is achieved when the reading is ~400-500microns of mercury.

7. The vacuum pump can be left to run over night. Be sure that the rooms refresh fan is turned on and that no hoses or electrical cables are touching the vacuum pump as the pump side operates between 50-70°C.

**Fill Loop With Refrigerant**

To fill loop with refrigerant use the Javac XTR-Pro refrigerant recovery machine as shown in Figure 96. When handling refrigerant it is important to wear safety glasses and gloves. Always read the MSDS sheets of the refrigerants being used before attempting to handle them.
1. To fill the loop use the Javac XTR-Pro refrigerant recovery machine. To the fill loop connect gas bottle, hoses, and recovery machine as shown in Figure 97.

2. With the valves at the fill ports in the loop closed, open the gas bottle valve. Allow gas to fill lines and recovery machine displacing any air.

3. Slightly open the charging line connections at the fill ports (red and blue lines at points A and B in Figure 97). Watch see liquid refrigerant seeping out. When this occurs all air has been displaced from the lines and recovery machine. Tighten the connections.
4. Zero the scale to achieve a tare point reading of the gas bottles weight.
5. Open the valves at the fill points on the loop.
6. Turn on the recovery machine.
7. While filling the loop watch the pressure on the pressure manifold. The pressure should be the saturated vapor pressure of the refrigerant at the current room temperature.
8. Observe the mass readings on the scale. When the mass reading reaches the target mass close inlet valve on the recovery machine.
9. Then close the valve on the gas bottle.
10. Now open the inlet valve on the recovery machine.
11. After 10 seconds close the inlet valve again.
12. Then turn set the valves to the positions shown in Figure 98. Be sure to slowly turn the inlet valve to purge taking care not to over-pressure the recovery machine.

![Figure 98. Recovery machine valve settings for purge](image)

13. When the LP gauge on the recovery machine reads a vacuum then the machine is completely purged.
14. Turn the inlet valve to closed.
15. Turn off the recovery machine.
16. Close the valves at the fill ports on the loop.
17. Disconnect charging hoses. Take care when removing hoses as some refrigerant will still be in the lines.
18. Place end caps onto fill ports.
19. Put away refrigerant recovery machine and store gas bottle.

**Withdraw Refrigerant From Loop**
To withdraw refrigerant from loop use the Javac XTR-Pro refrigerant recovery machine as shown in Figure 96. When handling refrigerant it is important to wear safety glasses and gloves. Always read the MSDS sheets of the refrigerants being used before attempting to handle them.

1. To withdraw refrigerant from the loop use the Javac XTR-Pro refrigerant recovery machine. To withdraw the refrigerant, connect gas bottle, hoses, and recovery machine as shown in Figure 99. Be sure that the bottle
   a. Is a pump down bottle and no check valve is installed in the bottles ports
   b. Has sufficient capacity to receive the refrigerant in the loop
   c. Is marked with its rated pressure. Take note of the pressure rating and make sure not to exceed this pressure when filling the bottle. Most pump down bottles have a pressure rating of 5,200kPa. But always check and verify.

   ![Figure 99. Refrigerant withdrawal setup.](image)

2. With the valve at the gas bottle inlet port closed, open the valves at the fill ports. Allow gas to fill lines and recovery machine displacing any air.
3. Slightly open the charging line connections at the gas bottle inlet port.
4. Watch to see liquid refrigerant seeping out. When this occurs all air has been displaced from the lines and recovery machine. Tighten the connections.
5. Zero the scale to achieve a tare point reading of the gas bottles weight.
6. Open the gas bottle inlet valve.
7. Turn on the recovery machine.
8. While filling the gas bottle, watch the pressure on the pressure manifold. The pressure should be the saturated vapor pressure of the refrigerant at the current room temperature. Make sure that the pressure does not exceed the rated pressure of the gas bottle.
9. Observe the mass readings on the scale. When the reading stabilizes this indicates that all the refrigerant has been withdrawn from the loop.
10. Then close the valves at the fill ports on the loop.
11. After 10 seconds, set the valves to the positions shown in Figure 100. Be sure to slowly turn the inlet valve to purge taking care not to over-pressure the recovery machine.

![Figure 100. Recovery machine valve settings for purge](image)

12. When the LP gauge on the recovery machine reads a vacuum then the machine is completely purged.
13. Turn the inlet valve to closed.
14. Turn off the recovery machine.
15. Close the gas bottle inlet valve.
16. Disconnect charging hoses. Take care when removing hoses as some refrigerant will still be in the lines.
17. Place end caps onto fill ports.
18. Put away refrigerant recovery machine and store gas bottle.

**Operate Loop**
1. Turn on cooling water pump in the following sequence
   a. Ensure that the short circuit valve connecting the inlet and outlet pipes on the water lines is open.
b. Ensure that the water inlet and outlet valves are open.
c. Turn on water.
d. Slowly close the short circuit valve.

2. Turn on oil pump.

3. Turn on oiler and set temperature.
   a. Switch main power supply on the side of the panel to “on”.
   b. Press the “green” button on the front of the panel.
   c. Once the LED display shows the temperature, press “set”.
   d. Adjust temperature using the up and down buttons.
   e. Press “set” again to save setting.

4. Allow the oiler to reach operating temperature set point.

5. Turn on working fluid pump at low mass flow rate (VSD set to 10).
   a. To operate VSD ensure power is being supplied to the VSD. There are two circuit breakers, one inside the lab and one next the VSD that need to be turned on.
   b. Press “Quick Menu”
   c. Navigate to channel 3 using the “+” or “-” buttons.
   d. On channel 3 press the “Change Data” button.
   e. Select the frequency setting required for the motor speed using “+” or “-“ buttons.

6. Apply torque to turbine and maintain rotation speed ~2000RPM.

7. Allow loop to run until turbine exit temperature stabilizes. Use the predicted value from the cycle TS diagram as a guide to what the exit temperature should be.

8. Adjust mass flow rate to required mass flow and allow turbine to run at maximum speed.

9. Once turbine exit temperatures stabilize data gathering can begin.

10. Start SuperDaq recording.
    a. Double click the “Start SuperDaq” icon on the Desktop
    b. Double click the “SuperDaq GUI” icon on the Desktop
    c. Click on “Connect” button in the upper right corner of the SuperDaq GUI window.
    d. A pop up window will appear. Click “Connect”.
    e. Press the “►” icon to start viewing data.
    f. Press the “●” icon to start recording data.

11. To gather data
    a. Adjust pump speed to achieve required mass flow rate.
    b. Set supply voltage to hysteresis brake to achieve desired RPM.
c. Allow turbine to run at speed for 2 min. Record RPM, Torque, Tgeo, P5, P6, T6, Power
   d. Continue adjusting RPM until all readings are recorded.

12. Shut off working fluid pump.
13. Shut off oil heater.
14. Shut off oil pump.
15. Shut off cooling water.
16. Shut off power supply to hysteresis brake.
17. Shut off SuperDaq
   a. When finished recording press the “■” icon to stop recording.
   b. Close the SuperDaq GUI window.
   c. Double click the “Stop SuperDaq Server” icon on the Desktop.
Appendix E.  Original Impulse Turbine FEA Results
Simulation Report_QGECE18_01

Rotor Shaft

2. Model Information

| Document | E:\02_Lab\01_CAD\QGECE18_## Turbine_03\QGECE18_01.par | - | Miniature | Axial |

3. Study Properties

<table>
<thead>
<tr>
<th>Study Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study name</td>
<td>Static Study 1</td>
</tr>
<tr>
<td>Study Type</td>
<td>Linear Static</td>
</tr>
<tr>
<td>Mesh Type</td>
<td>Tetrahedral</td>
</tr>
<tr>
<td>Iterative Solver</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran Geometry Check</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran command line</td>
<td></td>
</tr>
<tr>
<td>NX Nastran study options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran generated options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran default options</td>
<td></td>
</tr>
<tr>
<td>Surface results only option</td>
<td>On</td>
</tr>
</tbody>
</table>

4. Study Geometry

4.1 Solids

<table>
<thead>
<tr>
<th>Solid Name</th>
<th>Material</th>
<th>Mass</th>
<th>Volume</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>QGECE18_01.par</td>
<td>Stainless Steel, 316</td>
<td>0.298 kg</td>
<td>37159.383 mm³</td>
<td>2923.128 mN</td>
</tr>
</tbody>
</table>

5. Material Properties

5.1 Stainless Steel, 316

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>8027.000 kg/m³</td>
</tr>
</tbody>
</table>
6. Loads

<table>
<thead>
<tr>
<th>Load Name</th>
<th>Load Type</th>
<th>Load Value</th>
<th>Load Direction</th>
<th>Load Direction Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Torque 1</td>
<td>Torque</td>
<td>5.000 N-m</td>
<td>Location = ( 0.00, 0.00, 0.00 ), Axis = ( 1.00, 0.00, 0.00 )</td>
<td></td>
</tr>
<tr>
<td>Torque 2</td>
<td>Torque</td>
<td>5.000 N-m</td>
<td>Location = ( 0.00, 0.00, 0.00 ), Axis = ( -1.00, 0.00, 0.00 )</td>
<td></td>
</tr>
<tr>
<td>Centrifugal 1</td>
<td>Centrifugal</td>
<td>Angular Velocity = 60000.000 deg/s, Angular Acceleration = 0.000e+000 deg/s^2</td>
<td>Location = ( 0.00, 0.00, 0.00 ), Axis = Angular Velocity = ( -1.00, 0.00, 0.00 ), Angular Acceleration = ( -1.00, 0.00, 0.00 )</td>
<td></td>
</tr>
<tr>
<td>Temperature 1</td>
<td>Temperature</td>
<td>20.000 C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. Constraints

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Constraint Type</th>
<th>Degrees of Freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cylindrical 1</td>
<td>Cylindrical</td>
<td>FREE DOF: 1</td>
</tr>
</tbody>
</table>

8. Mesh Information

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Tetrahedral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of bodies meshed</td>
<td>1</td>
</tr>
<tr>
<td>Total number of elements</td>
<td>14,684</td>
</tr>
<tr>
<td>Total number of nodes</td>
<td>22,895</td>
</tr>
<tr>
<td>Subjective mesh size (1-10)</td>
<td>4</td>
</tr>
</tbody>
</table>

9. Results
### 9.1 Displacement Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>9.097e-005 mm</td>
<td>-31.700 mm</td>
<td>3.693 mm</td>
<td>-6.397 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>9.846e-003 mm</td>
<td>-89.200 mm</td>
<td>0.000 mm</td>
<td>5.000 mm</td>
</tr>
</tbody>
</table>

### 9.2 Stress Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>1.505e-001 MegaPa</td>
<td>-89.500 mm</td>
<td>-0.085 mm</td>
<td>-0.250 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>7.053e+001 MegaPa</td>
<td>-39.646 mm</td>
<td>5.146 mm</td>
<td>-0.000 mm</td>
</tr>
</tbody>
</table>
9.3 Factor of Safety Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000</td>
<td>-39.646 mm</td>
<td>5.146 mm</td>
<td>-0.000 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>2.000</td>
<td>-89.500 mm</td>
<td>-0.085 mm</td>
<td>-0.250 mm</td>
</tr>
</tbody>
</table>
Simulation Report_QGECE18_02

Inlet Housing

2. Model Information

<table>
<thead>
<tr>
<th>Study Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study name</td>
<td>Static Study 1</td>
</tr>
<tr>
<td>Study Type</td>
<td>Linear Static</td>
</tr>
<tr>
<td>Mesh Type</td>
<td>Tetrahedral</td>
</tr>
<tr>
<td>Iterative Solver</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran Geometry Check</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran command line</td>
<td></td>
</tr>
<tr>
<td>NX Nastran study options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran generated options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran default options</td>
<td></td>
</tr>
<tr>
<td>Surface results only option</td>
<td>On</td>
</tr>
</tbody>
</table>

3. Study Properties

<table>
<thead>
<tr>
<th>Solid Name</th>
<th>Material</th>
<th>Mass</th>
<th>Volume</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>QGECE18_02.par</td>
<td>Aluminum, 6061-T6</td>
<td>7.433 kg</td>
<td>2740790.858 mm^3</td>
<td>72843.643 mN</td>
</tr>
</tbody>
</table>

4. Study Geometry

4.1 Solids

5. Material Properties

5.1 Aluminum, 6061-T6

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>2712.000 kg/m^3</td>
</tr>
<tr>
<td>Coef. of Thermal Exp.</td>
<td>0.0000 /C</td>
</tr>
<tr>
<td>Thermal Conductivity</td>
<td>0.180 kW/m-C</td>
</tr>
</tbody>
</table>
Specific Heat | 920.000 J/kg-C
Modulus of Elasticity | 68947.570 MegaPa
Poisson's Ratio | 0.330
Yield Stress | 275.790 MegaPa
Ultimate Stress | 310.264 MegaPa
Elongation % | 0.000

6. Loads

<table>
<thead>
<tr>
<th>Load Name</th>
<th>Load Type</th>
<th>Load Value</th>
<th>Load Direction</th>
<th>Load Direction Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure 1</td>
<td>Pressure</td>
<td>800.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
</tbody>
</table>

7. Constraints

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Constraint Type</th>
<th>Degrees of Freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed 1</td>
<td>Fixed</td>
<td>FREE DOF: None</td>
</tr>
</tbody>
</table>

8. Mesh Information

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Tetrahedral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of bodies meshed</td>
<td>1</td>
</tr>
<tr>
<td>Total number of elements</td>
<td>62,761</td>
</tr>
<tr>
<td>Total number of nodes</td>
<td>94,433</td>
</tr>
<tr>
<td>Subjective mesh size (1-10)</td>
<td>3</td>
</tr>
</tbody>
</table>

9. Results

9.1 Displacement Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000e+000 mm</td>
<td>-130.000 mm</td>
<td>30.000 mm</td>
<td>145.000 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>3.229e-002 mm</td>
<td>1.384 mm</td>
<td>0.000 mm</td>
<td>11.021 mm</td>
</tr>
</tbody>
</table>
Total Translation

E:\02_Lab\01_CAD\QGECE18_01_CAD\QGECE18_02_Simulation\Static Study 1_1\Displacement_Total Translation.jpg

9.2 Stress Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>9.185e-002 MegaPa</td>
<td>150.000 mm</td>
<td>30.000 mm</td>
<td>-101.053 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>6.361e+001 MegaPa</td>
<td>-65.926 mm</td>
<td>0.000 mm</td>
<td>102.113 mm</td>
</tr>
</tbody>
</table>
9.3 Factor of Safety Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000</td>
<td>-65.926 mm</td>
<td>0.000 mm</td>
<td>102.113 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>2.000</td>
<td>150.000 mm</td>
<td>30.000 mm</td>
<td>-101.053 mm</td>
</tr>
</tbody>
</table>
Factor of Safety: E:\02_Lab\01_CAD\QGECE18 - Miniature Axial Turbine_03\QGECE18_02_Simulation\Static Study 1_1\Stress_Factor of Safety.jpg
Outlet Housing

2. Model Information

<table>
<thead>
<tr>
<th>Document</th>
<th>E:\02_Lab\01_CAD\QGECE18_##</th>
<th>-</th>
<th>Miniature</th>
<th>Axial</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QGECE18_03.par</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Study Properties

<table>
<thead>
<tr>
<th>Study Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study name</td>
<td>Static Study 1</td>
</tr>
<tr>
<td>Study Type</td>
<td>Linear Static</td>
</tr>
<tr>
<td>Mesh Type</td>
<td>Tetrahedral</td>
</tr>
<tr>
<td>Iterative Solver</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran Geometry Check</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran command line</td>
<td></td>
</tr>
<tr>
<td>NX Nastran study options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran generated options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran default options</td>
<td></td>
</tr>
<tr>
<td>Surface results only option</td>
<td>On</td>
</tr>
</tbody>
</table>

4. Study Geometry

4.1 Solids

<table>
<thead>
<tr>
<th>Solid Name</th>
<th>Material</th>
<th>Mass</th>
<th>Volume</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>QGECE18_03.par</td>
<td>Aluminum, 6061-T6</td>
<td>0.000 kg</td>
<td>0.000 mm^3</td>
<td>0.000 mN</td>
</tr>
</tbody>
</table>

5. Material Properties

5.1 Aluminum, 6061-T6

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>2712.000 kg/m^3</td>
</tr>
<tr>
<td>Coef. of Thermal Exp.</td>
<td>0.0000 /C</td>
</tr>
<tr>
<td>Thermal Conductivity</td>
<td>0.180 kW/m-C</td>
</tr>
</tbody>
</table>
### Specific Heat

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific Heat</td>
<td>920,000 J/kg-C</td>
</tr>
</tbody>
</table>

### Modulus of Elasticity

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modulus of Elasticity</td>
<td>68947.570 MegaPa</td>
</tr>
</tbody>
</table>

### Poisson's Ratio

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson's Ratio</td>
<td>0.330</td>
</tr>
</tbody>
</table>

### Yield Stress

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yield Stress</td>
<td>275.790 MegaPa</td>
</tr>
</tbody>
</table>

### Ultimate Stress

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ultimate Stress</td>
<td>310.264 MegaPa</td>
</tr>
</tbody>
</table>

### Elongation %

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elongation %</td>
<td>0.000</td>
</tr>
</tbody>
</table>

### Loads

<table>
<thead>
<tr>
<th>Load Name</th>
<th>Load Type</th>
<th>Load Value</th>
<th>Load Direction</th>
<th>Load Direction Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure 1</td>
<td>Pressure</td>
<td>800.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
</tbody>
</table>

### Constraints

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Constraint Type</th>
<th>Degrees of Freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed 1</td>
<td>Fixed</td>
<td>FREE DOF: None</td>
</tr>
</tbody>
</table>

### Mesh Information

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Tetrahedral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of bodies meshed</td>
<td>1</td>
</tr>
<tr>
<td>Total number of elements</td>
<td>52,189</td>
</tr>
<tr>
<td>Total number of nodes</td>
<td>78,387</td>
</tr>
<tr>
<td>Subjective mesh size (1-10)</td>
<td>3</td>
</tr>
</tbody>
</table>

### Results

#### 9.1 Displacement Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000e+000 mm</td>
<td>-40.000 mm</td>
<td>0.000 mm</td>
<td>-150.000 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>1.987e-002 mm</td>
<td>3.536 mm</td>
<td>30.000 mm</td>
<td>8.536 mm</td>
</tr>
</tbody>
</table>
9.2 Stress Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>1.090e-002 MegaPa</td>
<td>-150.000 mm</td>
<td>38.000 mm</td>
<td>150.000 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>4.156e+001 MegaPa</td>
<td>-66.773 mm</td>
<td>0.000 mm</td>
<td>101.526 mm</td>
</tr>
</tbody>
</table>
Von Mises E:A02_Lab\01_CAD\QGECE18 - # - Miniature Axial Turbine_03\QGECE18_03_Simulation\Static Study 1\Stress Von Mises.jpg
Simulation Report_QGECE18_04

Housing Ring

2. Model Information

| Document | VBoxsr\Desktop\02_Lab\01_CAD\QGECE18_## - Miniature Axial Turbine_03\QGECE18_04.par |

3. Study Properties

<table>
<thead>
<tr>
<th>Study Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study name</td>
<td>Static Study 1</td>
</tr>
<tr>
<td>Study Type</td>
<td>Linear Static</td>
</tr>
<tr>
<td>Mesh Type</td>
<td>Tetrahedral</td>
</tr>
<tr>
<td>Iterative Solver</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran Geometry Check</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran command line</td>
<td></td>
</tr>
<tr>
<td>NX Nastran study options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran generated options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran default options</td>
<td></td>
</tr>
<tr>
<td>Surface results only option</td>
<td>On</td>
</tr>
</tbody>
</table>

4. Study Geometry

4.1 Solids

<table>
<thead>
<tr>
<th>Solid Name</th>
<th>Material</th>
<th>Mass</th>
<th>Volume</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>QGECE18_04.par</td>
<td>Aluminum, 6061-T6</td>
<td>10.609 kg</td>
<td>3911901.058 mm³</td>
<td>103968.942 mN</td>
</tr>
</tbody>
</table>

5. Material Properties

5.1 Aluminum, 6061-T6

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>2712.000 kg/m³</td>
</tr>
<tr>
<td>Coef. of Thermal Exp.</td>
<td>0.0000 /C</td>
</tr>
<tr>
<td>Thermal Conductivity</td>
<td>0.180 kW/m-C</td>
</tr>
</tbody>
</table>
Specific Heat | 920.000 J/kg-C
Modulus of Elasticity | 68947.570 MegaPa
Poisson's Ratio | 0.330
Yield Stress | 275.790 MegaPa
Ultimate Stress | 310.264 MegaPa
Elongation % | 0.000

6. Loads

<table>
<thead>
<tr>
<th>Load Name</th>
<th>Load Type</th>
<th>Load Value</th>
<th>Load Direction</th>
<th>Load Direction Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure 1</td>
<td>Pressure</td>
<td>800.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
<tr>
<td>Temperature 1</td>
<td>Temperature</td>
<td>30.000 C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. Constraints

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Constraint Type</th>
<th>Degrees of Freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed 1</td>
<td>Fixed</td>
<td>FREE DOF: None</td>
</tr>
</tbody>
</table>

8. Mesh Information

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Tetrahedral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of bodies meshed</td>
<td>1</td>
</tr>
<tr>
<td>Total number of elements</td>
<td>13,456</td>
</tr>
<tr>
<td>Total number of nodes</td>
<td>22,202</td>
</tr>
<tr>
<td>Subjective mesh size (1-10)</td>
<td>3</td>
</tr>
</tbody>
</table>

9. Results

9.1 Displacement Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000e+000 mm</td>
<td>-30.000 mm</td>
<td>70.000 mm</td>
<td>-150.000 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>7.555e-002 mm</td>
<td>150.000 mm</td>
<td>70.000 mm</td>
<td>150.000 mm</td>
</tr>
</tbody>
</table>
9.2 Stress Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>2.982e-002 MegaPa</td>
<td>-141.011 mm</td>
<td>0.000 mm</td>
<td>132.949 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>4.933e+001 MegaPa</td>
<td>-150.000 mm</td>
<td>0.000 mm</td>
<td>-150.000 mm</td>
</tr>
</tbody>
</table>
9.3 Factor of Safety Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000</td>
<td>-150.000 mm</td>
<td>0.000 mm</td>
<td>-150.000 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>2.000</td>
<td>-141.011 mm</td>
<td>0.000 mm</td>
<td>132.949 mm</td>
</tr>
</tbody>
</table>
Factor of Safety: 

E:\02_Lab\01_CAD\QGECE18_03\Miniature Axial Turbine_03\QGECE18_04_Simulation\Static Study 1_2\Stress_Factor of Safety.jpg
Nozzle Top

2. Model Information

| Document | E:\02_Lab\01_CAD\QGECE18_## - Miniature Axial Turbine_03\QGECE18_11.par |

3. Study Properties

<table>
<thead>
<tr>
<th>Study Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study name</td>
<td>Static Study 1</td>
</tr>
<tr>
<td>Study Type</td>
<td>Linear Static</td>
</tr>
<tr>
<td>Mesh Type</td>
<td>Tetrahedral</td>
</tr>
<tr>
<td>Iterative Solver</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran Geometry Check</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran command line</td>
<td></td>
</tr>
<tr>
<td>NX Nastran study options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran generated options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran default options</td>
<td></td>
</tr>
<tr>
<td>Surface results only option</td>
<td>On</td>
</tr>
</tbody>
</table>

4. Study Geometry

4.1 Solids

<table>
<thead>
<tr>
<th>Solid Name</th>
<th>Material</th>
<th>Mass</th>
<th>Volume</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>QGECE18_11.par</td>
<td>Aluminum, 6061-T6</td>
<td>0.000 kg</td>
<td>0.000 mm^3</td>
<td>0.000 mN</td>
</tr>
</tbody>
</table>

5. Material Properties

5.1 Aluminum, 6061-T6

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>2712.000 kg/m^3</td>
</tr>
<tr>
<td>Coef. of Thermal Exp.</td>
<td>0.0000 /C</td>
</tr>
<tr>
<td>Thermal Conductivity</td>
<td>0.180 kW/m-C</td>
</tr>
</tbody>
</table>
Specific Heat | 920,000 J/kg°C
---|---
Modulus of Elasticity | 68947.570 MegaPa
Poisson's Ratio | 0.330
Yield Stress | 275.790 MegaPa
Ultimate Stress | 310.264 MegaPa
Elongation % | 0.000

### 6. Loads

<table>
<thead>
<tr>
<th>Load Name</th>
<th>Load Type</th>
<th>Load Value</th>
<th>Load Direction</th>
<th>Load Direction Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure 1</td>
<td>Pressure</td>
<td>2500.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
<tr>
<td>Pressure 2</td>
<td>Pressure</td>
<td>2500.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
<tr>
<td>Pressure 3</td>
<td>Pressure</td>
<td>2500.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
<tr>
<td>Pressure 4</td>
<td>Pressure</td>
<td>2500.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
<tr>
<td>Pressure 5</td>
<td>Pressure</td>
<td>2500.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
<tr>
<td>Pressure 6</td>
<td>Pressure</td>
<td>2500.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
<tr>
<td>Pressure 7</td>
<td>Pressure</td>
<td>2500.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
<tr>
<td>Pressure 8</td>
<td>Pressure</td>
<td>2500.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
<tr>
<td>Pressure 9</td>
<td>Pressure</td>
<td>2500.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
</tbody>
</table>

### 7. Constraints

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Constraint Type</th>
<th>Degrees of Freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed 1</td>
<td>Fixed</td>
<td>FREE DOF: None</td>
</tr>
</tbody>
</table>

### 8. Mesh Information

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Tetrahedral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of bodies meshed</td>
<td>1</td>
</tr>
<tr>
<td>Total number of elements</td>
<td>49,127</td>
</tr>
<tr>
<td>Total number of nodes</td>
<td>74,263</td>
</tr>
<tr>
<td>Subjective mesh size (1-10)</td>
<td>3</td>
</tr>
</tbody>
</table>

### 9. Results

#### 9.1 Displacement Results

**Result component: Total Translation**

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000e+000 mm</td>
<td>99.093 mm</td>
<td>13.441 mm</td>
<td>7.128 mm</td>
</tr>
</tbody>
</table>
### 9.2 Stress Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>2.052e-008 MegaPa</td>
<td>98.801 mm</td>
<td>37.000 mm</td>
<td>-5.000 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>6.429e+000 MegaPa</td>
<td>87.419 mm</td>
<td>-34.075 mm</td>
<td>-22.262 mm</td>
</tr>
</tbody>
</table>
9.3 Factor of Safety Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000</td>
<td>87.419 mm</td>
<td>-34.075 mm</td>
<td>-22.262 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>2.000</td>
<td>98.801 mm</td>
<td>37.000 mm</td>
<td>-5.000 mm</td>
</tr>
</tbody>
</table>
Rotor

2. Model Information

| Document | E:\02_Lab\01_CAD\QGECE18_## - Miniature Axial Turbine_03\QGECE18_15.par |

3. Study Properties

<table>
<thead>
<tr>
<th>Study Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study name</td>
<td>Static Study 1</td>
</tr>
<tr>
<td>Study Type</td>
<td>Linear Static</td>
</tr>
<tr>
<td>Mesh Type</td>
<td>Tetrahedral</td>
</tr>
<tr>
<td>Iterative Solver</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran Geometry Check</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran command line</td>
<td></td>
</tr>
<tr>
<td>NX Nastran study options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran generated options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran default options</td>
<td></td>
</tr>
<tr>
<td>Surface results only option</td>
<td>On</td>
</tr>
</tbody>
</table>

4. Study Geometry

4.1 Solids

<table>
<thead>
<tr>
<th>Solid Name</th>
<th>Material</th>
<th>Mass</th>
<th>Volume</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>QGECE18_15.par</td>
<td>Aluminum, 6061-T6</td>
<td>2.095 kg</td>
<td>772363.854 mm³</td>
<td>20527.578 mN</td>
</tr>
</tbody>
</table>

5. Material Properties

5.1 Aluminum, 6061-T6

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>2712.000 kg/m³</td>
</tr>
<tr>
<td>Coef. of Thermal Exp.</td>
<td>0.0000 /C</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Thermal Conductivity</td>
<td>0.180 kW/m-C</td>
</tr>
<tr>
<td>Specific Heat</td>
<td>920.000 J/kg-C</td>
</tr>
<tr>
<td>Modulus of Elasticity</td>
<td>68947.570 MegaPa</td>
</tr>
<tr>
<td>Poisson's Ratio</td>
<td>0.330</td>
</tr>
<tr>
<td>Yield Stress</td>
<td>275.790 MegaPa</td>
</tr>
<tr>
<td>Ultimate Stress</td>
<td>310.264 MegaPa</td>
</tr>
<tr>
<td>Elongation %</td>
<td>0.000</td>
</tr>
</tbody>
</table>

6. Loads

<table>
<thead>
<tr>
<th>Load Name</th>
<th>Load Type</th>
<th>Load Value</th>
<th>Load Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centrifugal 1</td>
<td>Centrifugal</td>
<td>Angular Velocity = 60000.000 deg/s, Angular</td>
<td>Location = ( 0.00, 0.00, 0.03 ), Axis = Angular Velocity = ( 0.00, 0.00, 1.00 ),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Acceleration = 0.000e+000 deg/s^2</td>
<td>Angular Acceleration = ( 0.00, 0.00, 1.00 )</td>
</tr>
<tr>
<td>Temperature 1</td>
<td>Temperature</td>
<td>70.000 C</td>
<td></td>
</tr>
</tbody>
</table>

7. Constraints

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Constraint Type</th>
<th>Degrees of Freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cylindrical 1</td>
<td>Cylindrical</td>
<td>FREE DOF: 1</td>
</tr>
</tbody>
</table>

8. Mesh Information

<table>
<thead>
<tr>
<th>Mesh type</th>
<th>Tetrahedral</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of bodies meshed</td>
<td>1</td>
</tr>
<tr>
<td>Total number of elements</td>
<td>470,548</td>
</tr>
<tr>
<td>Total number of nodes</td>
<td>695,225</td>
</tr>
<tr>
<td>Subjective mesh size (1-10)</td>
<td>6</td>
</tr>
</tbody>
</table>
9. Results

9.1 Displacement Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000 mm</td>
<td>-9.425 mm</td>
<td>7.336 mm</td>
<td>50.728 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>0.136 mm</td>
<td>-3.591 mm</td>
<td>99.935 mm</td>
<td>51.228 mm</td>
</tr>
</tbody>
</table>

![Displacement_Total_Translation.jpg](E:\02_Lab\01_CAD\QGECE18_\ # - Miniature Axial Turbine_03\QGECE18_15_Simulation\Static Study 1_4\Displacement_Total_Translation.jpg)

9.2 Stress Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>1.187e+000 MegaPa</td>
<td>-27.487 mm</td>
<td>-96.148 mm</td>
<td>2.208 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>1.665e+002 MegaPa</td>
<td>10.504 mm</td>
<td>5.685 mm</td>
<td>50.728 mm</td>
</tr>
</tbody>
</table>
9.3 Factor of Safety Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000</td>
<td>10.504 mm</td>
<td>5.685 mm</td>
<td>50.728 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>2.000</td>
<td>-27.487 mm</td>
<td>-96.148 mm</td>
<td>2.208 mm</td>
</tr>
</tbody>
</table>
Simulation Report_QGECE18_19

Nozzle Cover

2. Model Information

| Document | \vboxsrv\Desktop\02_Lab\01_CAD\QGECE18_## - Miniature Axial Turbine_03\QGECE18_19.par |

3. Study Properties

<table>
<thead>
<tr>
<th>Study Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study name</td>
<td>Static Study 1</td>
</tr>
<tr>
<td>Study Type</td>
<td>Linear Static</td>
</tr>
<tr>
<td>Mesh Type</td>
<td>Tetrahedral</td>
</tr>
<tr>
<td>Iterative Solver</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran Geometry Check</td>
<td>On</td>
</tr>
<tr>
<td>NX Nastran command line</td>
<td></td>
</tr>
<tr>
<td>NX Nastran study options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran generated options</td>
<td></td>
</tr>
<tr>
<td>NX Nastran default options</td>
<td></td>
</tr>
<tr>
<td>Surface results only option</td>
<td>On</td>
</tr>
</tbody>
</table>

4. Study Geometry

4.1 Solids

<table>
<thead>
<tr>
<th>Solid Name</th>
<th>Material</th>
<th>Mass</th>
<th>Volume</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>QGECE18_19.par</td>
<td>Aluminum, 6061-T6</td>
<td>0.634 kg</td>
<td>233681.953 mm^3</td>
<td>6210.705 mN</td>
</tr>
</tbody>
</table>

5. Material Properties

5.1 Aluminum, 6061-T6

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density</td>
<td>2712.000 kg/m^3</td>
</tr>
<tr>
<td>Coef. of Thermal Exp.</td>
<td>0.0000 /C</td>
</tr>
<tr>
<td>Thermal Conductivity</td>
<td>0.180 kW/m-C</td>
</tr>
<tr>
<td>Property</td>
<td>Value</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Specific Heat</td>
<td>920.000 J/kg-C</td>
</tr>
<tr>
<td>Modulus of Elasticity</td>
<td>68947.570 MegaPa</td>
</tr>
<tr>
<td>Poisson's Ratio</td>
<td>0.330</td>
</tr>
<tr>
<td>Yield Stress</td>
<td>275.790 MegaPa</td>
</tr>
<tr>
<td>Ultimate Stress</td>
<td>310.264 MegaPa</td>
</tr>
<tr>
<td>Elongation %</td>
<td>0.000</td>
</tr>
</tbody>
</table>

### 6. Loads

<table>
<thead>
<tr>
<th>Load Name</th>
<th>Load Type</th>
<th>Load Value</th>
<th>Load Direction</th>
<th>Load Direction Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure 1</td>
<td>Pressure</td>
<td>2500.000 kPa</td>
<td>Compressive</td>
<td>Normal to face</td>
</tr>
</tbody>
</table>

### 7. Constraints

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Constraint Type</th>
<th>Degrees of Freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed 1</td>
<td>Fixed</td>
<td>FREE DOF: None</td>
</tr>
</tbody>
</table>

### 8. Mesh Information

<table>
<thead>
<tr>
<th>Mesh type</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of bodies meshed</td>
<td>1</td>
</tr>
<tr>
<td>Total number of elements</td>
<td>43,098</td>
</tr>
<tr>
<td>Total number of nodes</td>
<td>66,157</td>
</tr>
<tr>
<td>Subjective mesh size (1-10)</td>
<td>3</td>
</tr>
</tbody>
</table>

### 9. Results

#### 9.1 Displacement Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000e+000 mm</td>
<td>-0.000 mm</td>
<td>-125.851 mm</td>
<td>-49.768 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>7.621e-002 mm</td>
<td>-42.000 mm</td>
<td>-110.618 mm</td>
<td>7.075 mm</td>
</tr>
</tbody>
</table>
9.2 Stress Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>6.795e-002 MegaPa</td>
<td>-37.714 mm</td>
<td>-70.618 mm</td>
<td>47.075 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>2.039e+002 MegaPa</td>
<td>-0.000 mm</td>
<td>-151.118 mm</td>
<td>13.500 mm</td>
</tr>
</tbody>
</table>
9.3 Factor of Safety Results

<table>
<thead>
<tr>
<th>Extent</th>
<th>Value</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.000</td>
<td>-0.000 mm</td>
<td>-151.118 mm</td>
<td>13.500 mm</td>
</tr>
<tr>
<td>Maximum</td>
<td>2.000</td>
<td>-37.714 mm</td>
<td>-70.618 mm</td>
<td>47.075 mm</td>
</tr>
</tbody>
</table>