REFERENCES

1. http://ect.jmcatalysts.com/applications-ssec-na-gas turbine.htm. Accessed on 22 March 2003.
2. Lefebvre, A. H., *Gas Turbine Combustion*. Second Edition, Taylor & Fancis, Philadelphia, 1998.
3. http://www.rolls-royce.com/energy/overview/powergen/default.jsp. Accessed on 25 May 2003.
4. Saravanamuttoo, H. I. H., Rogers, G. F. C., and Cohen, H., *Gas Turbine Theory*. Fifth edition, Prentice Hall, 2001.
5. Walsh, P. P., and Fletcher. P., *Gas Turbine Performance*. ASME PRESS, 1998.
6. Zomorodian, R., Khaledi, H., and Ghofrani, M. A., New Approach to Optimisation of Cogeneration Systems Using Genetic Algorithm. ASME Paper GT2006-90952, 2006.
7. Rogero, J. M., A Genetic Algorithm Based Optimisation Tool For the Preliminary Design of Gas Turbine Combustors. PhD Thesis, School of Mechanical Engineering, Cranfield University, 2002.
8. Lippert, A. M., Numerical Prediction of Internal Coolant Flow in Gas Turbine Blades. MEng Dissertation, University of Pretoria, 1994.
9. Mongia, H. C., A Synopsis of Gas Turbine Combustor Design Methodology Evolution of last 25 Years. ISABE-2001-1086, 2001.
10. Mongia, H. C., Aero-Thermal Design and Analysis of Gas Turbine Combustion Systems: Current Status and Future Direction. AIAA Paper 98-3982, 1998.
11. Nightingale, P., The Product-Process-Organization Relationship in Complex Development Projects. Research Policy, Elsevier, vol. 29, pp. 913-930, 2000.
12. Anand, M. S., and Priddin, C. H., Combustion CFD - A Key Driver to Reducing Development Cost and Time. Fifteenth International Symposium on Air Breathing Engines. Bangalore, India, ISABE-2001-1087, 2001.
13. Jones, M., Curnock, P., Bradbrook, S. J., and Brich, N., Evolutions in Aircraft Engine Design and Vision for the Future. Fifteenth International Symposium on Air Breathing Engines. Bangalore, India, 2001.

14. Lai, M. K., Reynolds, R. S., and Armstrong, J., CFD-based, Parametric, Design Tool for Gas Turbine Combustors from Compressor Deswirl Exit to Turbine Inlet. ASME Paper GT-2002-30090, 2002.
15. Haase, D., Optimisation in Industry – From Parametrization to Results. Symposium on Design Optimisation, Munchen, Germany, 1999.
16. Carlos, A., and Coello, C., An Updated Survey of Evolutionary Optimisation Techniques: State of the Art and Future Trends. IEEE, 1999.
17. Periaux, J., and Winter, G., *Genetic Algorithms in Engineering and Computer Science*, John Wiley and Sons, 1995.
18. Pardalos, P. M., Recent Developments and Trends in global optimisation, Journal of Computational and Applied Mathematics, vol. 124, pp. 209-228, 2001.
19. Crossley, W. A., Genetic Algorithm Approaches for Multiobjective Design of Rotor Systems. AIAA-96-4025-CP, 1996.
20. Paschereit, C. O., Schuermans, B., and Buche, D., Combustion Process Optimisation Using Evolutionary Algorithm. ASME Paper GT2003-38393, 2003.
21. De Kock, D. J., Optimal Tundish Methodology in a Continuous Casting Process. PhD Thesis, Department of Mechanical and Aeronautical Engineering, University of Pretoria, 2005.
22. Buche, D., Stoll, P., Dornberger, R., and Koumoutsakos, P., Multiobjective Algorithm for the Optimisation of Noisy Combustion Processes. IEEE Transactions on systems. Man, and Cybernetices - Part C: Applications and Reviews, vol. 32, No. 4, 2002.
23. Yurij, G., Evtushenko, G. H., *Numerical Optimization Techniques*. Optimization Software, Inc, Publication Division, 1985.
24. Snyman, J. A., and Hay, A. M., The Dynamic-Q Optimization Method: An alternative to SQP? Computers & Mathematics with Application, vol. 44, No. 14, pp. 1589-1598, 2002.

25. Snyman, J. A., Stander, N., and Roux, W. J., A Dynamic Penalty Function Method for Solution of Structural Optimization Problems. Applied Mathematical Modelling, vol. 18, pp. 453-460, 1994.
26. Grobman, J. S., Dittrich, R. T., and Graves, C. C., Pressure Drop and Air-Flow Distribution in Gas-Turbine Combustors. ASME Paper No. 56-A-208, 1956.
27. Knight, H. A., and Walker, R. B., The Component Pressure Losses in Combustion Chambers. NGTE, Reports and Memoranda No. 2987, November, 1953.
28. Shang, T., Guenette, G., Epstein, A., and Saxer, A., The Influence of Inlet Temperature Distortion on Rotor Heat Transfer in a Transonic Turbine. AIAA 95-3042, 1995.
29. Barringer, M. D, Thole, K. A., and Polanka, M. D., Effects of Combustor Exit Profiles on High Pressure Turbine Vane Aerodynamics and Heat Transfer. ASME, GT2006-90277, 2006.
30. Chana, K., Hirrion, J., and Jones, T., The Design development and Testing of a Non-Uniform Inlet Temperature Generator for the QinetiQ Transient Turbine Research Facility. ASME Paper 2003-GT-38469, 2003.
31. Povey, T., Chana, K., Jones, T., and Hurrion, J., The Effect of Hot-Streaks on HP Vane Surface and Endwall Heat Transfer: An Experimental and Numerical Study. ASME Paper GT2005-69066, 2005.
32. Krishnamoorty, V., Pai, B., and Sukhatme, S., Influence of Upstream Flow Conditions on the Heat Transfer to Nozzle Guide Vanes. Journal of Turbomachinery, Vol.110, pp.412-416, 1988.
33. Lefebvre, A. H., and Norster, E. R., The Design of Tubular Gas Turbine Combustion Chambers for Optimum Mixing Performance, Proceedings of Institution of Mechanical Engineers, vol. 183, pp. 150-155, 1969.
34. Durbin, M. D., Vangness, M. D., Ballal, D. R., Katta, V. R., Study of Flame Stability in a Step Swirl Combustor. ASME Paper 95-GT-111, 1995.
35. Sankaran, V., and Menon, S., Turbulence-Chemistry Interactions in Spray Combustion. ASME Paper GT-2002-30091, 2002.

36. Malecki, R. E., Colket, M. B., Madabhushi, R. K., McKinney, R. G., Ouyang, H., and Syed, S., Application of an Advanced CFD-Based Analysis System to the PW6000 Combustor to Optimize Exit Temperature Distribution - Part I: Description and Validation of the Analysis Tool. ASME Journal of Engineering for Gas Turbines and Power, 2001.

37. Snyder, T. S., Stewart, J. F., Stoner, M. D., and McKinney, R. G., Application of an Advanced CFD-Based Analysis System to the PW6000 Combustor to Optimize Exit Temperature Distribution - Part II: Comparison of Predictions to Full Annular Rig Test data. ASME Journal of Engineering for Gas Turbines and Power, 2001.

38. McGuik, J. J., and Spencer, A., Coupled and Uncoupled CFD Prediction of the Characteristics of Jets from Combustor Air Admission Ports. ASME Paper 2000-GT-0125, 2000.

39. Samareh, A. J., Status and Future of Geometry Modelling and Grid Generation for Design and Optimization. Journal of Aircraft, vol. 36, No. 1. 1999.

40. Lai, M. K., Reynolds, R. S., and Armstrong, J., CFD-Based, Parametric, Design Tool for Gas Turbine Combustors from Compressor Deswirl Exit to Turbine Inlet. ASME Turbo Expo 2002, Amsterdam, The Netherlands.

41. FLUENT News, “The Berl Combustor Revisited”, Vp1. xii, ISSUE1, 2003.

42. http://www.tu-darmstadt.de/fb/mb/ekt/tecflam/ebi.html. Accessed on 10 August 2003.

43. Widmann, J. F., Charagundla, S. R., and Presser, C., Benchmark Database for Input and Validation of Multiphase Combustion Models. Chemical and Physical Processes in Combustion. Combustion Institute, Proceedings, 1999, Raleigh, NC 1999.

44. Launder, B. E., and Spalding, D. B., The Numerical Computation of Turbulent Flows. Computer Methods in Applied Mechanics and Engineering (3), 269-289, 1974.

45. Trinh, C. M., Turbulence Modelling of Confined Swirling Flows. PhD Thesis, Riso National Laboratory, Roskilde, Denmark, 1993.

46. Magnussen, B. F., and Hjerthager, B. H., On Mathematical Modeling of Turbulent Combustion with Special Emphasis on Soot Formation and Combustion. The Combustion Institute, Sixteenth Symposium on Combustion, Pittsburgh, Pennsylvania, 1976.
47. Holdeman, J. D., Liscinsky, D. S., Oechsle, V. L., Samuelsen, G. S., and Smith, C. E., Mixing of Multiple Jets with a Confined Subsonic Cross Flow: Part 1-Cylindrical Duct, ASME, Journal of Engineering for Gas Turbine and Power, vol. 119, pp. 852-862, 1997.
48. Gulati, T., Tolpadi, A., and Vanduesen, G., Effect of Dillution Air on Scalar Flow Field at the Combustor Exit, AIAA, 94-0221, 1994.
49. Tangarila, V., Tolpadi, A., Danis, A., and Mongia, H., Parametric Modeling Approach to Gas Turbine Combustor Design, ASME Paper, IGTI Conference, 2000-GT-0129, 2000.
50. Catalano, L. A., Dadone, A., and Manodoro, D., Saponaro, A., Design Optimization and Validation of Improved Duct-Burners for Combined-Cycle Plants. International Joint Power Generation Conference, New Orleans, 2001-PWR-19026, LOUISIANA, 2001.
51. Catalano, L. A., Dadone, A., and Manodoro, D., Design of High Performance Duct-Burners for Combined-Cycle Plants Using Progressive Optimization. ASME Turbo Expo 2002, GT-20002-30087, Amsterdam, The Netherlands, 2002.
52. Becz, S., and Cohen, J. M., Charanterization Of Mixing For A Jet In Cross-flow Using Proper Orthogonal Decomposition. AIAA paper 2005-0307, 2005.
53. Despierre, A., Stuttaford P. J., and Rubini, P. A., Preliminary gas turbine combustor design using a genetic algorithm. ASME paper 97-GT-72, 1997.
54. Fuligno, L., Micheli, D. and Poloni, C., An Integrated Design Approach for Micro Gas Turbine Combustors: Preliminary 0-D and Simplified CFD Based Optimisation. ASME Paper, GT2006-90542, 2006.
55. Back, T., and Schwefel, H., An Overview of Evolutionary Algorithms for Parameter Optimisation. Evolutionary Computations vol. 1, No. 1, pp. 1-23, 1993.

56. Jakob, W., Gorges-Schleuter, M., and Blume, C., Application of Genetic Algorithms to Task Planning and Learning. Parallel Problem Solving from Nature PPSN2, pp. 291-310, 1992.

57. Rudoff, R. C., Design, Evaluation, and Characterization of a Gas Turbine Model Laboratory Combustor with Discrete Wall Injections. MSc. Dissertation, University of California, Irvine, 1986.

58. Lefebvre, A. H., *Atomization and Spray.* Hemisphere Publishing Corporation, Washington, DC, 1987.

59. Lefebvre, A. H., The Role of Fuel Preparation in Low-Emissions Combustion. Jounrnal of Engineering and Power, vol.117, pp. 617-652, 1996.

60. Micklow, G. J., Roychoudhury, S., Nguyen, H. L., and Cline, M. C., Emissions Reduction by Varying the Swirler Airflow Split in Advanced Gas Turbine Combustors. IGTI Paper 92-GT-110, 1992.

61. Su, K., and Zhou, C. Q., Numerical Study of Spray Parametric Effects on Gas Turbine Combustion Performance. ASME Paper GT2001-44365, 2001.

62. Claypole, T. C., Pollutant Formation in Swirling Jets. PhD Thesis, Department of Mechanical Engineering and Energy Studies, University College, Cardiff, 1980.

63. Gupta, A. K., Lewis, M. J., and Qi, S., Effect of Swirl on Combustion Characteristics in Premixed Flames. ASME Paper 97-GT-276, 1997.

64. Hafez, A. H., El-Mahallawy, F. M., and Shehata, A. M., Effect of Burner and Operating Conditions on Mixing and Heat Transfer Along a Combustion Chamber. ASME COGEN-TURBO, IGTI, vol. 9, 1994.

65. Squires, K. D., and Eaton, J. K., Particle response and turbulence modification in isotropic turbulence. Physics of Fluids, vol. 2, No. 7, pp. 1191-1203, 1990.

66. Datta, A., and Som, S. K., Combustion and emission characteristics in a gas turbine combustor at different pressure and swirl. Applied Thermal Engineering, vol. 19, No. 199, pp. 949-967, 1999.

67. Richards, C. D., and Samuelsen, S., The Role of Primary Jets in the Dome Region Aerodynamics of Model Can Combustor. IGTI, No. 90-GT-142, 1990.

68. Vilayanur, S. R., Davis, N. T., and Samuelsen, S., Effect of Inlet Conditions on Lean Premixed Gas Turbine Combustor Performance. International Gas Turbine & Aeroengine Congress & Exhibition, Stockholm, Sweden, 1998.

69. Sivaramakrishna, G., Muthuveerappan, N., Shankar, V., and Sampathkumaran, T. K., CFD Modeling of Aero Gas Turbine Combustor. ASME Paper, 2001-GT-0063, 2001.

70. Hinze, J. O., *Turbulence*. McGraw-Hill, Inc, 1975.

71. Poinsot, T., and Veynante, D., *Theoretical and Numerical Combustion*. R T Edwards, Inc, Philadelphia, 2001.

72. Kuo, K. K., *Principles of Combustion*. John Wiley & Sons, Inc, 1986.

73. Abbott, M. B., and Basco, D. R., *Computational Fluid Dynamics, An Introduction for Engineers*. Longman Scientific & Technical, UK, 1989.

74. "FLUENT 6 User's Guide", Lebanon, NH, USA, FLUENT Inc, 2003.

75. Jones, W. P., and Launder, B. E., "The Prediction of Laminarisation with a Two-Equation Turbulence Model", International Journal Heat Mass Transfer, vol. 3, p. 239, 1974.

76. Launder, B. E., and Spalding, D. B., Lectures in Mathematical Models of Turbulence. Academic Press, New York, 1974.

77. Magnussen, B. F., and Hjertager, B. H., On Mathematical Modeling of Turbulent Combustion with Special Emphasis on Soot Formation and Combustion. Sixteenth Symposium (International) on Combustion, The Combustion Institute, pp.719-729, 1976.

78. Buckmaster, J. D., and Ludford, G. S. S., *Lectures on Mathematical Combustion*. The Universities Press Ltd, Belfast, 1983.

79. Chen, H. C., and Patel, V. C., Near-Wall Turbulence Models for Complex Flows Including Separation. AIAA Journal, vol. 26, No. 6, pp. 641-648, 1988.

80. Williams, F. A., *Combustion Theory*. Second Edition, Benjamin/Cunning Publishing Company, Inc, 1934.

81. Samarskii, A., and Pokhilko, V., On Problem of Toxic Species Formation in natural Gas Combustion. Experimentation, Modelling and Computation in

Flow, Turbulence and Combustion, John Wiley & Sons, Inc, vol. 2, pp. 113-128, 1997.

82. Crowe, C. T., Sharma, M. P., and Stock, D. E., The Particle-Source-In-Cell (PSI-CELL) Model for Gas-Droplet Flows. Journal of Fluids Engineering, vol. 99, pp. 325-332, 1997.
83. Tap, F. A., Dean, A. J., and Van Buijttenen, J. P., Experimental and Numerical Spray Characterization of a Gas Turbine Fuel Atomizer in Cross Flow. ASME paper GT-2002-30100, 2002.
84. Hetsroni, G., Particles-turbulence interaction. International Journal Multiphase Flow, vol. 15. pp. 735-748, 1992.
85. Gosman, A. D., and Ioannides, E., Aspect of computer simulations of liquid-fueled combustors. Journal of Energy, vol. 7, pp. 482-490, 1983.
86. Faith, G. M., Evaporation and Combustion of Sprays. Progress in Energy and Combustion Science, vol. 9, pp. 1-76, 1983.
87. Baumal, A. E., McPhee, J. and Calamai, P. H., Application of genetic algorithms of an active vehicle suspension design, Computer Methods in Applied Mechanics and Engineering," vol. 163, pp. 87-94, 1998.
88. Eberhard, P., Schiehlen, W., and Bestle, D., "Some advantages of stochastic methods in multi-criteria optimization of multibody systems," Archive of Applied Mechanics, vol. 69, pp. 543-554, 1998.
89. Els, P. S., and Uys, P. E., "Investigation of the applicability of the Dynamic-Q Optimisation Algorithm to Vehicle Suspension Design," Mathematical and Computer Modelling, vol. 37, Nos. 9-10, pp. 1029-1046, 2003.
90. Visser, J. A., and De Kock, D. J., "Optimization of Heat Sink Mass Using the Dynamic-Q Numerical Optimization Method," Communications in Numerical Methods in Engineering, vol. 18, No. 10, pp. 721-727, 2002.
91. Snyman, J. A., A New Dynamic Method for Unconstrained Minimization. Applied Mathematical Modelling, vol. 6, pp. 449-462, 1982.
92. Snyman, J. A., The LFOPC Leap-Frog Algorithm for Constrained Optimization, Computers and Mathematics with Applications, vol. 40, pp. 1085-1096, 2000.

93. Hay, A. M., Optimal Dimensional Systhesis of Planar Parallel Manipulators with Respect to Workspaces. PhD Thesis, Department of Mechanical and Aeronautical Engineering, University of Pretoria, South Africa, 2003.

94. Snyman, J. A., and Stander. N., A New Successive Approximation Method for Optimum Structural Design. AIAA Journal, vol. 32, pp. 1310-1315, 1994.

95. Snyman, J. A., Private Communication. Department of Mechanical and Aeronautical Engineering, University of Pretoria, South Africa, 1998.

96. De Kock, D. J., Industrial Applications of Computational Flow Optimisation. MSc Dissertation, Department of Mechanical and Aeronautical Engineering, University of Pretoria, 1999.

97. Morris, R. M., Snyman, J. A., and Meyer, J. P., "Jets in Crossflow Mixing Analysis Using Computational Fluid Dynamics and Mathematical Optimization," Journal of Propulsion and Power, vol. 23, No. 3, pp. 618-628, 2007.

98. De Kock, D. J., Optimal Tundish Methodology in a Continuous Casting Process. PhD Thesis, Department of Mechanical and Aeronautical Engineering. University of Pretoria, 2005.

99. QNET CFD "Best Practice Advice for Combustion and Heat Transfer". vol. 1 No. 4, 2002.

100. Libby, P. A., and Williams, F. A., *Turbulent Reacting Flows*. Academic Press, 1994.

101. Morris, R. M., An Experimental and Numerical Investigation of gas Turbine Research Combustor. MSc. Dissertation, University of Pretoria, Pretoria, 2000.

102. Dukowicz, J. K. A., Particle Fluid Numerical Model for Liquid Sprays. Journal of Computational Physics, vol. 35, pp. 229-253, 1980.

103. O'Rouke, P. J., Collective Drop Effects in Vaporizing Liquid Sprays. Ph.D. Thesis, Princeton University, and Los Alamos National Laboratory Report, LA-9069-T, 1981.

104. Rossin, P., and Rammler, B., The Laws Governing the Fineness of Powdered Coal. Inst. Of Fuel, pp. 29-36, 1933.

105. Rizk, N. K., and Lefebvre, A. H., Drop-Size Distribution Characteristics of Spill-Return Atomizers. Journal of Propulsion, vol. 1, No. 1, 1984.
106. Chin, J. S., Li, W. M., and Zhang, Y., Experimental Study of the Effect of Dense Spray on drop Size Measurement by Light Scattering Technology. Transactions of ASME IGTI, Paper No. 90-GT-1, 1990.
107. Guoqiang, L., and Gutmark, E. J., Geometry Effects on the Flow Field and the Spectral Characteristics of a Tripple Annular Swirler. ASME Paper, Procceedings of ASME TURBO EXPO, GT-2003-38799, 2003.
108. Mathews, J. H., and Fink, K. D., *Numerical Methods Using MATLAB*. Prentice Hall, London, 1999.
109. Kuffmann, A., and Henry-Labordere, G., Integer and Mixed Programming: Theory and Applications. Mathematics in Science and Engineering, vol. 137, Academic Press, 1977.
110. Salkin, H. M., Integer Programming. Addison-Wisley publishing Company, 1975.
111. Steuer, R. E., *MultiCruteria Optimisation: Theory, Computation, and Application.* John Wiley & Sons, 1986.
112. Snyman, J. A., De Kock, D. J., Craig, K. J., and Venter, P. J., Toolkit for Design Optimization (TDO): An educational aid to mathematical modelling and optimization. Quaestiones Mathematicae, Suppl. 1, pp. 227-236, 2001.
113. Vanoverberche, K. P., Van Den Bulck, E. V., and Tummers, M. J., "Confined Annular Swirling Jet Combustion," Combustion Science and Technology, vol. 175, pp. 545-578, 2003.

APPENDIX A

GAMBIT JOURNAL FILE FOR GEOMETRY MODELLING AND GRID GENERATION

Since geometric modelling and grid generation are the most time consuming and labour intensive processes in computational fluid dynamics based design systems, the GAMBIT journaling toolkit has been intensively used to replay model building for different computational fluid dynamics sessions. This appendix outlines the GAMBIT journal file procedures that are necessary to replay the geometry modelling, mesh generation and boundary condition for consecutive computational fluid dynamics simulations. This procedure is performed for every design suggested by the mathematical optimiser.

```
/ Journal File for GAMBIT 2.2.30, Database 2.2.14, ntx86 BH04110220
/ Identifier "default_id1772"
/ File opened for write Thu Jan 27 10:04:58 2005.

vertex create coordinates 0 0 0
vertex create coordinates 0 27.75 0
vertex create coordinates 0 34.8 -5
vertex create coordinates 0 37.35 -16
vertex create coordinates 0 37.35 -33
vertex create coordinates 0 38.15 -33
vertex create coordinates 0 38.15 -28.15
vertex create coordinates 0 40.4 -28.5
vertex create coordinates 0 40.4 -99
vertex create coordinates 0 41.2 -99
vertex create coordinates 0 41.2 -94.5
vertex create coordinates 0 43.1 -94.5
vertex create coordinates 0 43.1 -164
vertex create coordinates 0 0 -164

edge create straight "vertex.1" "vertex.2" "vertex.3" "vertex.4" "vertex.5"\
"vertex.6" "vertex.7" "vertex.8" "vertex.9" "vertex.10" "vertex.11" "vertex.12"\
"vertex.13" "vertex.14"

face create revolve "edge.1" "edge.2" "edge.3" "edge.4" "edge.5" "edge.6"\
 "edge.7" "edge.8" "edge.9" "edge.10" "edge.11" "edge.12" "edge.13" dangle \
180 vector 0 0 1 origin 0 0 0

vertex create coordinates 0 0 0
vertex create coordinates -0.75 7.864318 0
vertex create coordinates -4.015767 6.803206 0
vertex create coordinates -7.551550 11.669794 0
vertex create coordinates -0.75 13.879751 0

edge create center2points "vertex.29" "vertex.31" "vertex.30" minarc arc
```

```
edge create center2points "vertex.29" "vertex.32" "vertex.33" minarc arc
edge create straight "vertex.30" "vertex.33"
edge create straight "vertex.31" "vertex.32"

face create wireframe "edge.39" "edge.40" "edge.41" "edge.42"

face cmove "face.14" multiple 4 dangle 36 vector 0 0 1 origin 0 0 0

face split "face.1" connected face "face.14"
face split "face.1" connected face "face.15"
face split "face.1" connected face "face.16"
face split "face.1" connected face "face.17"
face split "face.1" connected face "face.18"

edge create straight "vertex.9" "vertex.23"
edge create straight "vertex.18" "vertex.5"

vertex create coordinates 0 0 -164

edge create straight "vertex.29" "vertex.70"
edge split "edge.79" tolerance 1e-06 edge "edge.81" keeptool connected
edge split "edge.13" tolerance 1e-06 edge "edge.81" keeptool connected
edge split "edge.80" tolerance 1e-06 edge "edge.81" keeptool connected
edge delete "edge.81" lowertopology

vertex create coordinates 0 0 0

edge create straight "vertex.80" "vertex.79" "vertex.73" "vertex.76"
face create wireframe "edge.38" "edge.30" "edge.33" "edge.35" "edge.83" \
  "edge.90" "edge.85" real
face create wireframe "edge.9" "edge.10" "edge.11" "edge.79" "edge.90" \
  "edge.13" "edge.12" real
face create wireframe "edge.21" "edge.24" "edge.27" "edge.29" "edge.83" \
  "edge.89" "edge.80" real
face create wireframe "edge.5" "edge.6" "edge.7" "edge.87" "edge.89" \
  "edge.79" "edge.8" real
edge split "edge.1" vertex "vertex.80" connected
face create wireframe "edge.16" "edge.18" "edge.20" "edge.80" "edge.88" \
  "edge.1" real
face create wireframe "edge.88" "edge.87" "edge.4" "edge.3" "edge.2" \
  "edge.91" real

face create wireframe "edge.83" "edge.28" "edge.79" real
face create wireframe "edge.80" "edge.19" "edge.87" real

$x5=-48.5

edge create revolve "vertex.53" "vertex.57" "vertex.61" "vertex.64" \
  "vertex.67" height 10 angle $x5 vector 0 0 1 origin 0 0 0
volume create rotate "face.14" onedge "edge.92" twist $x5
```

```
volume create rotate "face.15" onedge "edge.93" twist $x5
volume create rotate "face.16" onedge "edge.94" twist $x5
volume create rotate "face.17" onedge "edge.95" twist $x5
volume create rotate "face.18" onedge "edge.96" twist $x5

edge delete "edge.92" "edge.93" "edge.94" "edge.95" "edge.96" lowertopology


undo begingroup
edge picklink "edge.75" "edge.78" "edge.72" "edge.73" "edge.68" "edge.69" \
  "edge.65" "edge.64" "edge.62" "edge.59"
edge mesh "edge.59" "edge.62" "edge.64" "edge.65" "edge.69" "edge.68" \
  "edge.73" "edge.72" "edge.78" "edge.75" successive ratio1 1 intervals 7
undo endgroup
undo begingroup
edge picklink "edge.76" "edge.77" "edge.71" "edge.74" "edge.70" "edge.67" \
  "edge.66" "edge.63" "edge.61" "edge.60"
edge mesh "edge.60" "edge.61" "edge.63" "edge.66" "edge.67" "edge.70" \
  "edge.74" "edge.71" "edge.77" "edge.76" successive ratio1 1 intervals 8
undo endgroup
undo begingroup
edge picklink "edge.133" "edge.136" "edge.125" "edge.128" "edge.117" \
  "edge.120" "edge.109" "edge.112" "edge.101" "edge.104"
edge mesh "edge.104" "edge.101" "edge.112" "edge.109" "edge.120" "edge.117" \
  "edge.128" "edge.125" "edge.136" "edge.133" successive ratio1 1 intervals 7
undo endgroup
undo begingroup
edge picklink "edge.134" "edge.135" "edge.126" "edge.127" "edge.118" \
  "edge.119" "edge.110" "edge.111" "edge.102" "edge.103"
edge mesh "edge.103" "edge.102" "edge.111" "edge.110" "edge.119" "edge.118" \
  "edge.127" "edge.126" "edge.135" "edge.134" successive ratio1 1 intervals 8
undo endgroup
undo begingroup
edge picklink "edge.130" "edge.129" "edge.132" "edge.131" "edge.124" \
  "edge.122" "edge.123" "edge.116" "edge.121" "edge.115" "edge.114" \
  "edge.108" "edge.107" "edge.113" "edge.105" "edge.106" "edge.98" "edge.97" \
  "edge.100" "edge.99"
edge mesh "edge.99" "edge.100" "edge.97" "edge.98" "edge.106" "edge.105" \
  "edge.113" "edge.107" "edge.108" "edge.114" "edge.115" "edge.121" \
  "edge.116" "edge.123" "edge.122" "edge.124" "edge.131" "edge.132" \
  "edge.129" "edge.130" successive ratio1 1 intervals 10
undo endgroup
undo begingroup
volume mesh "volume.5" "volume.4" "volume.3" "volume.2" "volume.1" map \
  intervals 10

volume create stitch "face.31" "face.28" "face.29" "face.4" "face.3" "face.2" \
  "face.1" "face.14" "face.15" "face.16" "face.17" "face.18" real
volume create stitch "face.30" "face.26" "face.27" "face.8" "face.5" "face.6" \
  "face.7" "face.31" real
```

```
volume create stitch "face.9" "face.10" "face.11" "face.12" "face.13" \
  "face.24" "face.25" "face.30" real

$x11=6
$x21=2.5
$d1=$x11-1
$z1=$x11

volume create height 60 radius1 $x21 radius3 $x21 offset 0 0 30 zaxis frustum
volume move "volume.9" dangle 90 vector -1 0 0 origin 0 0 0
volume move "volume.9" offset 0 0 -68.8
volume move "volume.9" dangle (-180/($z1*2)) vector 0 0 -1 origin 0 0 0

face split "face.8" connected face "face.58"
volume delete "volume.9" lowertopology

do para "$z1" init 1 incr (1)
face cmove "face.60" multiple $d1 dangle (180/$z1) vector 0 0 1 origin 0 0 0
enddo

face split "face.8" connected face "face.61"
face split "face.8" connected face "face.62"
face split "face.8" connected face "face.63"
face split "face.8" connected face "face.64"
face split "face.8" connected face "face.65"

$x1=4
$x2=3.3
$d=$x1-1
$z=$x1

volume create height 60 radius1 $x2 radius3 $x2 offset 0 0 30 zaxis frustum
volume move "volume.9" dangle 90 vector -1 0 0 origin 0 0 0
volume move "volume.9" offset 0 0 -44.1
volume move "volume.9" dangle (-180/($z*2)) vector 0 0 -1 origin 0 0 0

face split "face.8" connected face "face.72"
volume delete "volume.9" lowertopology

do para "$z" init 1 incr (1)
face cmove "face.74" multiple $d dangle (180/$z) vector 0 0 1 origin 0 0 0
enddo

face split "face.8" connected face "face.75"
face split "face.8" connected face "face.76"

$x3=5
$x4=6
$dd=$x3-1
$zd=$x3
```

```
volume create height 60 radius1 $x4 radius3 $x4 offset 0 0 30 zaxis frustum
volume move "volume.9" dangle 90 vector -1 0 0 origin 0 0 0
volume move "volume.9" offset 0 0 -123.5
volume move "volume.9" dangle (-180/($zd*2)) vector 0 0 -1 origin 0 0 0

face split "face.12" connected face "face.80"
volume delete "volume.9" lowertopology

do para "$z" init 1 incr (1)
face cmove "face.82" multiple $dd dangle (180/$zd) vector 0 0 1 origin 0 0 0
enddo

face split "face.12" connected face "face.83"
face split "face.12" connected face "face.84"
face split "face.12" connected face "face.85"
face split "face.12" connected face "face.86"

face link "face.24" "face.25" edges "edge.90" "edge.13" vertices "vertex.76" \
  "vertex.76" reverse periodic
face link "face.26" "face.27" edges "edge.89" "edge.79" vertices "vertex.73" \
  "vertex.73" reverse periodic
face link "face.28" "face.29" edges "edge.88" "edge.87" vertices "vertex.79" \
  "vertex.79" reverse periodic

undo begingroup
edge picklink "edge.4"
edge mesh "edge.4" successive ratio1 1 intervals 20
undo endgroup
undo begingroup
edge picklink "edge.3"
edge mesh "edge.3" successive ratio1 1 intervals 10
undo endgroup
undo begingroup
edge picklink "edge.2"
edge mesh "edge.2" successive ratio1 1 intervals 8
undo endgroup
undo begingroup
edge picklink "edge.91"
edge mesh "edge.91" firstlength ratio1 1.387 intervals 35
undo endgroup
undo begingroup
edge picklink "edge.17" "edge.15" "edge.14"
edge mesh "edge.14" "edge.15" "edge.17" successive ratio1 1 intervals 70
undo endgroup
undo begingroup
edge picklink "edge.25" "edge.26" "edge.19" "edge.22"
edge mesh "edge.22" "edge.19" "edge.26" "edge.25" successive ratio1 1 \
  intervals 70
undo endgroup
```

```
undo begingroup
edge picklink "edge.27"
edge mesh "edge.27" successive ratio1 1 intervals 2
undo endgroup
undo begingroup
edge picklink "edge.21"
edge mesh "edge.21" successive ratio1 1 intervals 1
undo endgroup
undo begingroup
edge picklink "edge.88"
edge mesh "edge.88" lastlength ratio1 1.189 intervals 45
undo endgroup
undo begingroup
edge picklink "edge.80"
edge mesh "edge.80" lastlength ratio1 1.684 intervals 25
undo endgroup
undo begingroup
edge picklink "edge.24"
edge mesh "edge.24" successive ratio1 1 intervals 5
undo endgroup
face mesh "face.2" map size 1
face mesh "face.3" map size 1
undo begingroup
face mesh "face.2" map size 1
face mesh "face.3" map size 1
face mesh "face.4" map size 1
face mesh "face.7" triangle size 1
face mesh "face.6" pave size 1
face mesh "face.31" triangle size 1
face mesh "face.1" triangle size 1
face mesh "face.28" triangle size 1
volume mesh "volume.6" tetrahedral intervals 10

edge picklink "edge.8"
edge mesh "edge.8" successive ratio1 1.008 ratio2 1.008 intervals 50
undo endgroup
undo begingroup
edge picklink "edge.30"
edge mesh "edge.30" successive ratio1 1 intervals 1
undo endgroup
undo begingroup
edge picklink "edge.33"
edge mesh "edge.33" successive ratio1 1 intervals 5
undo endgroup
undo begingroup
edge picklink "edge.35"
edge mesh "edge.35" successive ratio1 1 intervals 2
undo endgroup
undo begingroup
edge picklink "edge.89"
```

```
edge mesh "edge.89" successive ratio1 1.012 intervals 35
undo endgroup
undo begingroup
edge picklink "edge.83"
edge mesh "edge.83" firstlength ratio1 2.08 intervals 30
undo endgroup
undo begingroup
edge picklink "edge.34" "edge.36"
edge mesh "edge.36" "edge.34" successive ratio1 1.01 ratio2 1.01 intervals 70
undo endgroup
undo begingroup
edge picklink "edge.28" "edge.31"
edge mesh "edge.31" "edge.28" successive ratio1 1 intervals 70
undo endgroup
undo begingroup
edge picklink "edge.38"
edge mesh "edge.38" firstlength ratio1 2.278 intervals 40
undo endgroup
undo begingroup
edge picklink "edge.90"
edge mesh "edge.90" lastlength ratio1 2.08 intervals 35
undo endgroup
undo begingroup
edge picklink "edge.85"
edge mesh "edge.85" successive ratio1 1 intervals 30
undo endgroup
undo begingroup
edge picklink "edge.37"
edge mesh "edge.37" successive ratio1 1 intervals 60
undo endgroup
```

APPENDIX B

C++ PROGRAM FOR NUMERICAL INTEGRATION

This Appendix shows a numerical integration technique that is used to derive the objective function for mathematical optimisation. The Trapezoidal Rule was implemented into a computer code using a C++ compiler for numerical integration of the curve generated by the combustor exit temperature profile.

```
#include <stdio.h>          // Functions for performing input and output
#include <math.h>           // Mathematical functions (such as abs())

double *Tvalues;            // Vector of T values read from file
double *xvalues;            // Vector of x values read from file

int count = 0;              // Number of readings to read from input file
char inputfname[80];        // Name of the input file
char outputfname[80];       // Name of the output file
double interval = 0.0;      // Interval over which to integrate
double Ti = 0.0;            // Ideal temperature
double Xn = 0.0;            // Domain of ideal temperature


/*
 * This function calculates the integral of the
 * curve defined by the x and T values read from the file.
 * The integral is calculated using the traditional trapezoidal
 * method.
 */
double integral(int n, double h) {
        double result = 0.0;  // The variable in which the integral is saved
        int i;  // The current index in the Tvalues vector

        result = result + Tvalues[0];           // f(a)
        result = result + Tvalues[n - 1];       // f(b)

        // sum_of(2 * f(a + kh))
        for (i = 1; i < n; i++)
                result = result + 2.0 * Tvalues[i];

        return result * (h / 2.0);
}

/*
 * This function reads x and T values from a CFD-format file.  The header
 * and footer information should be removed from the file before running
 * the program.
 */
```

```
void getValuesFromFile(void) {
	FILE *f = 0;
	double T = 0.0;
	double x = 0.0;
	int index = 1;
	int i = 0;
	char s[256];

	f = fopen(inputfname, "r");
	if (!f) {
		printf("Couldn't open file %s!", inputfname);
		exit(0);
	}

	// Skip first 4 lines of input file

	for (i = 0; i < 3; i++)
		fscanf(f, "%[^\n]\n", s);

	// Read count - 1 lines (x, y values) from input file.
	// We use count - 1 because count was incremented in
	// getSettingsFromFile to accommodate extra f(a) value
	for (i = 0; i < (count - 1); i++) {
		fscanf(f, "%le\t%le", &x, &T);
		Tvalues[index] = T;
		xvalues[index] = x;
		index++;
	}
	fclose(f);

	// Insert extra f(a) value, where f(a) = f(h)
	Tvalues[0] = Tvalues[1];
	xvalues[0] = 0.0;
}

/*
 * This function reads six configuration settings from a file
 * called settings.txt.  These six configurations settings are
 * the name of the input file, the name of the file to which
 * output should be directed, a count of how many readings
 * should/can be read from the input file, the size of the
 * interval over which to integrate (h), the ideal temperature
 * (Ti), and the domain over which this ideal temperature should
 * occur (Xn).  The settings.txt file must list these settings
 * in the given order, preferably separated by new lines.
 */
void getSettingsFromFile(void) {
	FILE *f = fopen("settings.txt", "r");
	if (!f) {
		printf("Couldn't open file settings.txt!");
```

```
                exit(0);
        }

        fscanf(f, "%s", inputfname);
        fscanf(f, "%s", outputfname);
        fscanf(f, "%d", &count);
        count++;

        fscanf(f, "%le", &interval);

        fscanf(f, "%le", &Ti);
        fscanf(f, "%le", &Xn);
        fclose(f);
}

/*
 * This methods sorts (in numerical order) the two n-dimensional double
 * arrays that are passed to it as parameters.  The a-array is sorted
 * in numerical order, while the b-array is adjusted to remain in sync
 * with the a-array.  In other words, assume that before running this
 * method, a value x was at index i in array a, and some value y was
 * also at index i, but in array b.  After running the method, assume
 * value x was moved to index j in array a.  Therefore, value y will
 * now be at index j in array b.
 */
void selectionSort(double *a, double *b, int n) {
        double temp;
        int chosen;
        int leftmost;
        int j;

        for (leftmost = 0; leftmost < n - 1; leftmost++) {
                chosen = leftmost;
                for (j = leftmost + 1; j < n; j++)
                        if (a[j] < a[chosen])
                                chosen = j;
                temp = a[chosen];
                a[chosen] = a[leftmost];
                a[leftmost] = temp;

                temp = b[chosen];
                b[chosen] = b[leftmost];
                b[leftmost] = temp;
        }
}

void main(void) {
        double integ = 0.0;
        double idealArea = 0.0;
        FILE *f = 0;
```

```
    // Read the configuration settings
    getSettingsFromFile();

    // Allocate memory space for the arrays containing the T
    // and x values
    Tvalues = (double *) malloc(sizeof(double) * count);
    xvalues = (double *) malloc(sizeof(double) * count);

    // Read the T and x values from the input file
    getValuesFromFile();

    // Sort the T and x values, according to x
    selectionSort(xvalues, Tvalues, count);

    // Find the area under the curve defined by the T values
    // read from the input file
    integ = integral(count, interval);

    // Calculate the ideal area
    idealArea = Ti * Xn;

    // Open the output file and write the absolute value of the
    // difference between the ideal area and the curve area to
    // it
    f = fopen(outputfname, "w");
    fprintf(f, "%f", (fabs(idealArea - integ)));
    fclose(f);

    // Deallocate the memory we allocated above for the T and x
    // values
    free(Tvalues);
    free(xvalues);
}
```