REPORT NO: MIT-GFR-038

TOPICAL REPORT

Control System Strategies and Dynamic Response for Supercritical CO_2 Power Conversion Cycles

by:

N. A. Carstens
P. Hejzlar
M. J. Driscoll

September 15, 2006

Center for Advanced Nuclear Energy Systems
MIT Nuclear Engineering Department

Project PI
Professor M. J. Driscoll
MIT Nuclear Engineering
Department
77 Massachusetts Ave.,
Bldg. 24-215
Cambridge, MA 02139
(617) 253-4219
Email: mickeyd@mit.edu

Contract Technical Monitor
Dr. Paul S. Pickard
Manager, Advanced Nuclear
Concepts Department
Sandia National Laboratories
PO Box 5800, MS 1136
Albuquerque, NM 87185-1136
(505) 845-3046
Email: pspicka@sandia.gov

Abstract

GAS-PASS, a dynamic simulation and control code for gas-cooled Brayton Cycle reactor power conversion systems has been significantly modified to deal with the use of supercritical carbon dioxide as the working fluid. A wide range of improvements have been incorporated, including updating to modern Fortran 90 coding, incorporation of NIST real gas properties in a rapid fashion, more detailed modeling of turbomachinery performance, modeling of printed circuit heat exchange, and extensive reformulation and rewriting of key component subroutines. Of particular significance are the methods devised to overcome convergence problems caused by compression near the critical point of CO_2, and the attendant large variations in properties in the main compressor, precooler and low temperature recuperator.

The latest version of the code is exercised on several simple transients, to demonstrate satisfactory performance of both the code and the S-CO_2 PCS which it simulates. Initial results suggest significant potential for several cycle control strategies including temperature control, inventory control, and bypass control. Next-step applications and worthwhile refinements are also targeted.

Acknowledgements

The authors wish to thank Dr. Richard Vilim of Argonne National Laboratory, the original developer of Gas-Pass/H. His close, continuous, and extensive involvement have greatly aided this project.

Jeff Wang originally modeled the S-CO_2 turbomachinery. His initial work has made development of correlations for turbine and compressor performance possible. Dr. Yifang Gong of the MIT Gas Turbine Laboratory of the Department of Aeronautics and Astronautics greatly expanded Mr. Wang's work and has provided frequent data updates. His advice regarding turbomachinery has facilitated understanding and progress.

Dr. Shih-Ping Kao, a research scientist in the MIT CANES group, provided sage advice based on his extensive experience in systems modeling.

Two interns in the Summer 2006 MIT MSRP, Isaac Matthews and Dustin Langewisch, made useful contributions in the component modeling area.

Finally, we acknowledge the Sandia National Laboratory for funding this research under the contract with MIT for "Qualification of the Supercritical CO_2 Power Conversion Cycle for Advanced Reactor Applications" and its successor "Supercritical CO_2 Cycle Control System Simulation and Plant Design Studies."

Table of Contents

Control System Strategies and Dynamic Response for Supercritical CO_2 Power Conversion Cycles 1
Abstract 2
Acknowledgements **3**
List of Tables **10**
1 Introduction to the S-CO2 Cycle 11
1.1 Foreword **11**
1.2 Background **11**
1.3 System Model **13**
1.4 Organization of this Report **14**
2 Fluid Properties 16
2.1 Non-linear S-CO_2 Fluid Property Behavior **17**
2.2 Using NIST RefProp with S-CO_2 ..... **20**
2.2.1 Key NIST RefProp Features 21
2.2.2 RefProp CO_2 Equation of State 23
2.2.3 RefProp CO_2 Caveats 24
2.3 Tabular Fluid Properties **25**
2.3.1 Linearly Indexed Property Tables 26
2.3.2 Log Indexed Property Tables 26
2.3.2.1 Background 27
2.3.2.2 Log Function 27
2.3.2.3 Property Indexes 28
2.3.2.4 Speed and Accuracy 31
2.33 GAS-PASS/CO_2 Property Tables 33
2.4 Fluid Property Summary **34**
3 Turbomachinery Performance Mapping 35
3.1 Turbomachinery Performance Maps **35**
3.2 Handling Performance Curves **41**
3.2.1 Interpolating on a Shaft Speed Performance Curve 44
3.2.1.1 Notes on Correlations 44
3.2.1.2 Monotonic Trimming 44
3.2.1.3 Linear Interpolation 45
3.2.1.4 Cubic Spline Interpolation 46
3.2.1.5 Cubic Spline Problems with Rapid Slope Changes 47
3.2.1.6 Shaft Speed Curve Modeling Summary 48
3.2.2 Interpolating Between Shaft Speed Performance Curves 49
3.2.2.1 Interpolating Between Shaft Speed Curve Summary 54
3.3 Turbomachine Fluid Property Variation Effect **54**
3.3.1 Compressor Density Equation 55
3.3.2 Implementing Property Correction 57
4 Printed Circuit Heat Exchangers 58
4.1 Background **59**
4.2 1-D Straight Channel Correlations **60**
4.21 Applicable Literature 60

4.22 Straight Channel Heat Transfer Correlation ... 62
4.23 Straight Channel Pressure Drop ... 63
4.24 Conclusion ... 66
4.3 Computational Application ... 67
4.31 Using a Single Nodalized Axial Channel ... 67
4.32 Solving in Enthalpy Space ... 70
4.33 Nodal Averaging ... 75
4.34 Node Memory ... 75
4.35 Avoiding Surrounding Loops ... 75
4.35 Conclusion ... 76
4.4 Steady State PCHE Results ... 77
4.41 Precooler Heat Exchanger Performance ... 77
4.42 Low Temperature Recuperator Performance ... 82
4.43 High Temperature Recuperator Performance ... 85
4.44 Intermediate Heat Exchanger Performance ... 87
Conclusion ... 87
4.5 Precooler Thermal Inertia ... 88
4.51: Precooler Background ... 88
4.52: Precooler Thermal Inertia Analysis ... 93
4.6: PCHE Implicit Solution Method ... 97
4.7: PCHE Modeling Summary ... 104
5 The GAS-PASS/CO_2 Code ... 105
5.1 Introduction ... 105
5.2 GAS-PASS/CO_2 ... 105
5.2.1 Input/Output ... 106
5.2.2 Source Code ... 107
5.3 GAS-PASS Solution Process ... 108
5.4 Component Conservation Equations ... 110
5.4.1 Equation Models and Assumptions ... 111
5.4.2 Preventing Mass Equation Linear Dependence ... 113
5.4.2.1 Linear Mass Dependence Example – Steady State ... 113
5.4.2.2 Linear Mass Dependence Example – Transient ... 114
5.4.2.3 Avoiding Linear Mass Dependence ... 115
5.4.2.4 Implementing the Solution ... 117
5.5 Gas-Pass Solution Techniques ... 119
5.5.1 Progressive Solution Tolerance ... 119
5.5.2 Time Step Failure ... 120
5.5.3 Jacobian Guessing ... 120
5.5.3.1 Jacobian Slope Estimation ... 120
5.5.3.2 Jacobian Scaling ... 121
5.5.3.3 Numerical Extrapolation ... 122
5.5.4 User Controlled Dynamic Time stepping ... 126
5.5.5 Solution Techniques Conclusion ... 127
5.6 GAS-PASS/CO_2 Summary ... 127
6 S-CO_2 Cycle Dynamic Performance ... 128
6.1 Compressor Property Effects ... 129

6.1.1 Increasing Heat Rejection 129
6.2 Decreasing Reactor Power Transient.. 131
6.3 Increasing Reactor Power Transient 137
6.4 Increasing Heat Rejection Transient 142
6.5 Transient Summary 145
7 Summary, Conclusions, & Future Work.. 146
7.1 Summary... 146
7.2 Past Goals 147
7.3 Future Work.. 149
Appendix A: GAS-PASS/CO_2 Equations.. 151
A.1 Steady State Equations.. 151
7.3.1.1 A.1.1 Intermediate Heat Exchanger.. 152
7.3.1.2 A.1.2 Turbine.. 153
7.3.1.3 A.1.3 High Temperature Recuperator.. 154
7.3.1.4 A.1.4 Mixing Junction 155
7.3.1.5 A.1.5 Low Temperature Recuperator 156
7.3.1.6 A.1.6 Valve 157
7.3.1.7 A.1.7 Recompressing Compressor... 157
7.3.1.8 A.1.8 Splitting Junction 158
7.3.1.9 A.1.9 Precooler 159
7.3.1.10 A.1.10 Main Compressor... 160
7.3.1.11 A.1.11 Inventory 160
7.3.1.12 A.1.12 Shaft 161
A.2 Time-Difference Equations.. 161
7.3.1.13 A.2.1 Intermediate Heat Exchanger.. 161
7.3.1.14 A.2.2 Turbine.. 161
7.3.1.15 A.2.3 High Temperature Recuperator.. 162
7.3.1.16 A.2.4 Low Temperature Recuperator 162
7.3.1.17 A.2.5 Mixing Junction 162
7.3.1.18 A.2.6 Main Compressor.. 162
7.3.1.19 A.2.7 Recompressing Compressor... 163
7.3.1.20 A.2.8 Splitting Junction 163
7.3.1.21 A.2.9 Precooler 163
7.3.1.22 A.2.10 Inventory 163
7.3.1.23 A.2.11 Shaft 163
Appendix B: Roots of Nonlinear Systems.. 164
B.1 Mathematical Methods.. 164
7.3.1.24 B.1.1 Good Guess (Linear Leaps).. 164
7.3.1.25 B.1.2 Global Convergence (Curve Crawlers) 165
7.3.1.26 B.1.3 Tensor Methods... 165
B.2 Computational Implementations 165
7.3.1.27 B.2.1 The IMSL Solution Method 165
7.3.1.28 B.2.2 SUNDIALS 166
7.3.1.29 B.2.3 TRILINOS.. 166
Appendix C: References 167

List of Figures

Figure 1-1: Reference Version, S-CO_2 Recompression Cycle 12
Figure 1-2: 300 MWe S-CO2 Layout 14
Figure 2-1: S-CO_2 Recompression cycle Diagram 17
Figure 2-2: S-CO_2 Cycle Density 18
Figure 2-3: S-CO_2Cycle Isobaric Specific Heat 19
Figure 2-4: S-CO_2 Cycle Temperature vs. Entropy Diagram 19
Figure 2-5: Main Compressor Enthalpy vs. Entropy Region 20
Figure 2-6: CO_2 BWR versus FEQ Equations of State.. 23
Figure 2-7: Pressure Data Density 29
Figure 2-8: Enthalpy Data Density 30
Figure 2-9: Entropy Data Density 30
Figure 2-10: Relative Tabular Error 32
Figure 2-11: Absolute Log-Indexed Tabular Error 32
Figure 3-1: Radial Main Compressor Efficiency Map 36
Figure 3-2: Radial Recompressing Compressor Efficiency Map 36
Figure 3-3: Radial Main Compressor Pressure Ratio Map 37
Figure 3-4: Radial Recompressing Compressor Pressure Ratio Map 37
Figure 3-5: Axial Turbine Efficiency Map 38
Figure 3-6: Axial Turbine Pressure Ratio Map 38
Figure 3-7: Axial Main Compressor Efficiency Map 40
Figure 3-8: Axial Recompressing Compressor Efficiency Map 40
Figure 3-9: Axial Main Compressor Pressure Ratio Map 41
Figure 3-10: Axial Recompressing Compressor Pressure Ratio Map 41
Figure 3-11: Original Gas-Pass Axial Turbine Efficiency Off-Design Performance Map 42
Figure 3-12: Original Gas-Pass Helium Axial Compressor Shaft Speed Lines 43
Figure 3-13: Axial Main Compressor Efficiency Curves 43
Figure 3-14: S-CO_2 Turbine 120% Shaft Speed Efficiency versus Mass Flow Rate 45
Figure 3-15: Linear and Cubic Spline Interpolation for the S-CO2 Axial 46
Figure 3-16: Cubic Spline Slope Oscillation 48
Figure 3-17: Shaft Speed Interpolation Example 49
Figure 3-18: Connecting Points Example 50
Figure 3-19: Maxima Line 51
Figure 3-20: Parallel Maxima Lines 52
Figure 3-21: Turbine Curve Efficiency Difficulty 53
Figure 3-22: Turbine Curve Pressure Ratio Complexity 53
Figure 4-1: S-CO_2 Recompression cycle 59
Figure 4-2: Semi-circular Tube Thermal Entrance Length 62
Figure 4-3: PCHE Channel Layout 67
Figure 4-4: PCHE Single Channel 68
Figure 4-5: PCHE Single Channel Nodalized Axially 68
Figure 4-6: PCHE Computation Method 69
Figure 4-7: Precooler Temperature Profile 71
Figure 4-8: Precooler Enthalpy Profile 71

Figure 4-9: Precooler Pressure Profile 72
Figure 4-10: Main Compressor Operating Space 73
Figure 4-11: Gas-Pass Temperature Jacobian 74
Figure 4-12: Gas-Pass Enthalpy Jacobian 74
Figure 4-13: Steady State CO_2 Cp 77
Figure 4-14: Steady State Water HTC 78
Figure 4-15: Steady State CO_2 HTC 78
Figure 4-16: Steady State Overall HTC 79
Figure 4-17: CO_2 Pin=8MPa Cp 79
Figure 4-18: CO_2 Pin=7.5MPa Cp 80
Figure 4-19: CO_2 Cp Profiles at 10% Mass Flow Rate 80
Figure 4-20: CO_2 Cp Profiles at 50% Mass Flow Rate 81
Figure 4-21: CO_2 Cp Profiles at 110% Mass Flow Rate 81
Figure 4-22: LTR Steady State HTCs 83
Figure 4-23: LTR Steady State Cps 83
Figure 4-24: LTR 10% Mass Flow Rate HTRs 84
Figure 4-25: LTR 50% Mass Flow Rate HTRs 84
Figure 4-26: LTR 110% Mass Flow Rate HTRs 85
Figure 4-27: HTR Steady State HTCs 86
Figure 4-28: HTR Steady State Cps 86
Figure 4-29: Precooler Temperatures 89
Figure 4-30: Precooler Density 90
Figure 4-31: Precooler Isobaric Specific Heat 90
Figure 4-32: Precooler Reynolds #'s 91
Figure 4-33: Precooler HTCs 91
Figure 4-34: Precooler Overall HTC 92
Figure 4-35: Precooler Heat Transfer 92
Figure 4-36: Precooler Biot # 93
Figure 4-37: Titanium Temperature Distribution 94
Figure 4-38: Titanium Temperature Difference 94
Figure 4-39: Precooler Total Heat Capacity 95
Figure 4-40: PCHE Zone Model 97
Figure 4-41: PCHE Zone Computational Model 100
Figure 5-1: Jacobian 109
Figure 5-2: GAS-PASS/CO2 Flowchart 110
Figure 5-3: Network Plant Diagram for the S-CO_2 Recompression cycle 111
Figure 5-4: Component Mass Equations 113
Figure 5-5: Jacobian Scaling Example 121
Figure 5-6: Axial Main Compressor Efficiency 123
Figure 5-7: Axial Main Compressor Efficiency with Linear Extrapolation 124
Figure 5-8: Axial Main Compressor Efficiency with Added Extensions 124
Figure 6-1: 150% Mpre - Compressor Pressures 130
Figure 6-2: 150% Mpre - Compressor Pressures without Property Variation 130
Figure 6-3: -Qrx Immediate Transient - Heat Transfer 131
Figure 6-4: -Qrx Immediate Transient - Temperatures 132
Figure 6-5: -Qrx Immediate Transient – Mass Flow Rates\ 133

Figure 6-6: -Qrx Immediate Transient - HTR Mass Flow Rates 134
Figure 6-7: -Qrx Immediate Transient - Turbine Mass Flow Rates 135
Figure 6-8: -Qrx Long-Term - Efficiency... 135
Figure 6-9: -Qrx Long Term – Temperatures 136
Figure 6-10: -Qrx Long Term – Pressures 136
Figure 6-11: +Qrx Immediate Transient - Heat Transfer.. 138
Figure 6-12: +Qrx Immediate Transient – HTR Mass Flow Rates 138
Figure 6-13: +Qrx Immediate Transient – Work 139
Figure 6-14: +Qrx Immediate Transient - Temperatures.. 139
Figure 6-15: +Qrx Long-Term – Heat Transfer.. 141
Figure 6-16: +Qrx Long-Term - Work 141
Figure 6-17: +Mpre Short Term - Mass Flow Rates 142
Figure 6-18: +Mpre Long-Term - Pressures.. 143
Figure 6-19: +Mpre Long-Term – Temperatures 144
Figure 6-20: +Mpre Long-Term - Work.. 144
Figure 7-1: Axial Main Compressor 100% Shaft Speed Efficiency............................. 147

List of Tables

Table 1-1: Topics for S-CO2 Indirect Cycle Control Evaluations 12
Table 2-1: NIST RefProp 7.0 Available Fluids 22
Table 2-2: Known RefProp CO_2 Problems 24
Table 2-3: Gas-Pass Run Times by Data Source 26
Table 2-4: GAS-PASS/CO_2 Property Tables 33
Table 3-1: Density Ratios at Main Compressor Operating Conditions 56
Table 4-1: Hot Fluid Data Points 69
Table 4-2: Cold Fluid Data Points 70
Table 4-3: Precooler Geometry 88
Table 4-4: Precooler Volumes 88
Table 4-5: Precooler Inlet and Outlet Properties 88
Table 5-1: Gas-Pass Input and Output Files 107
Table 5-2: Gas-Pass Source Files 108
Table 7-1: Topics for Future S-CO_2 Brayton Indirect Cycle Control Evaluations 149

1 Introduction to the S-CO2 Cycle

1.1 Foreword

This topical report is a deliverable under our Sandia contract "Supercritical CO_2 Cycle Control System Simulation and Plant Design Studies" and its predecessor "Qualification of the Supercritical CO_2 Power Conversion Cycle for Advanced Reactor Applications."

It covers work done on modification of the GAS-PASS reactor system dynamics code since last reported in:

> M.J. Driscoll, P. Hejzlar, N. Carstens and Y. Wang, "Simulation of Supercritical CO_2 Brayton Cycle Plants," MIT-GFR-017, Sept. 2004

followed by

> N.A. Carstens, P. Hejzlar and M.J. Driscoll, "Description of Supercritical CO_2 Systems Control Model," MIT-GFR-027, Sept. 2005.

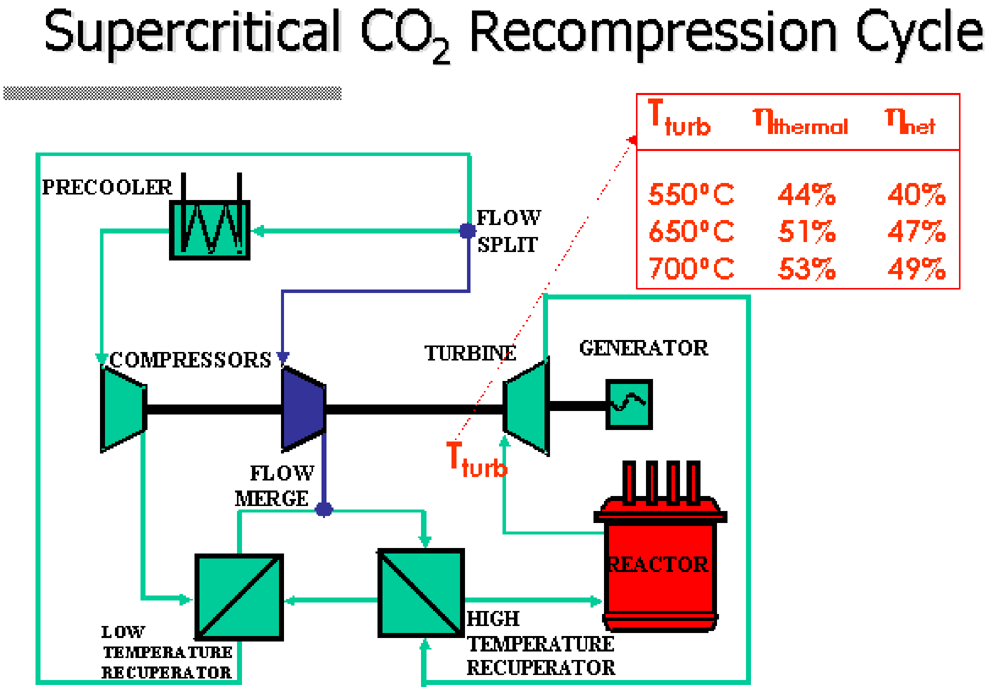
The body of the report focuses on two major topics: recent changes made to the aforementioned earlier versions, and presentation of the results of several transient runs made in the course of commissioning the code. In general, the presentation is descriptive and interpretative: mathematical details are documented in *Appendix A:* GAS-PASS/CO2 Equations.

Because the subject program is a work in progress, the following convention has been adopted for identification of the version of the code extant at any given point in time. The supercritical carbon dioxide (hereafter referenced S-CO_2) capable revision of the original GAS-PASS/H is titled GAS-PASS/CO_2. It is hover noted that the code is being developed as a general code capable of modeling various Brayton cycles with different fluids. The cycle specific layout is introduced through input data. Thus, the version GAS-PASS/CO_2 should be understood in this context as a version where S-CO_2 is developed and tested. When subsequent significant changes are introduced then this title will also be followed by a code version number. The version described in this report will be designated 1.0.

1.2 Background

Sufficient work has now been carried out worldwide to confirm the attractive steady state performance of the S-CO_2 recompression cycle, as shown in *Figure 1-1*. The principal unmet need is for more attention to transient behavior and control requirements.

Development of the GAS-PASS/CO_2 code, in a joint MIT/ANL effort, was initiated to address this objective.



Supercritical CO_2 Recompression Cycle

T_{turb}	$\eta_{thermal}$	η_{net}
550°C	44%	40%
650°C	51%	47%
700°C	53%	49%

Figure 1-1: Reference Version, S-CO_2 Recompression Cycle

Table 1-1 (from MIT-GFR-017) lists the initial menu of subtasks identified at the start of this process. Since then the following additions and modifications have been identified:

(1) comparisons of radial vs. axial turbomachinery to the reference all-axial case, specifically
   - radial main and recompression compressors
   - radial main and axial recompression
   - and, at lower priority, radial turbine

(2) effect of temperature and flow changes in precooler inlet water (hence compressor inlet CO_2), and use of recirculation to modulate same

(3) determination of minimum self-sustainable power level to evaluate use of the power conversion system for shutdown heat removal; likewise use of a pony (or startup) motor to extend the low end range.

Table 1-1: Topics for S-CO2 Indirect Cycle Control Evaluations

1. Control strategies: Bypass/Inventory/Temperature; including location of bypass valves, adjustment of main/recompression compressor flow split

2. Quasi-steady-state operation between 50% and 100% power, and power transition between states: 10% steps and 5%/min. ramps
3. Normal Cycle startup and shutdown
4. Rapid Transients
    (a) reactor scram
    (b) primary system loss of flow, loss of coolant
    (c) loss of load
    (d) loss of heat sink
    (e) power cycle lowdown
    (f) turbomachinery failure

1.3 System Model

The subject program was written in a highly modular fashion so as to make it potentially applicable to a wide variety of Brayton cycles. However this first version was confined in its commissioning runs to a very specific application, as follows:

(a) The code simulates a 600 MWth, ~300 MWe recompression type, S-CO_2 PCS. Figure 1-2 shows the specific modular component layout developed. However the code is fully capable if simulation over the full range of interest, 20 to 1200 MWe, as defined in the companion reports, Ref ([1],[2],[3],[4],[5]).

(b) CO_2 is the working fluid of choice. But note that the code is capable of importing properties of over 50 pure fluids from the NIST RefProp data base, and can also concoct mixtures thereof.

(c) The power conversion system (PCS) is coupled to the nuclear reactor primary coolant system via an intermediate heat exchanger. In this instance the primary coolant is sodium in view of the recent decision to pursue this option for GNEP. Extension to other primary coolants is straightforward.

(d) After initial evaluation using all axial turbomachines the system modeling effort has evolved to using radial compressors, which are described and developed in Ref([6]).

It is anticipated that these restrictions will be maintained for the foreseeable future, until one complete set of transient and control studies has been completed.

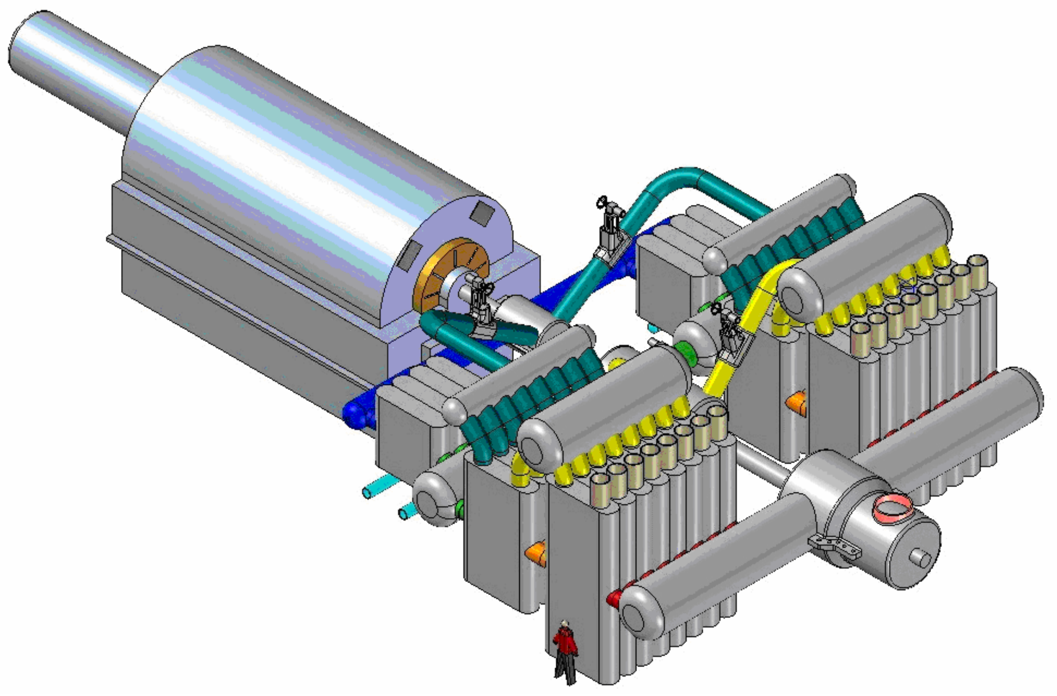


Figure 1-2: 300 MWe S-CO2 Layout

1.4 Organization of this Report

This report is organized in seven chapters and an appendices:

Chapter 1 presented a brief S-CO_2 cycle introduction and this report's organization.

Chapter 2 presents the fluid property methods and models used to allow rapid and accurate calculation of real fluid properties. It will focus upon using the NIST RefProp code with S-CO_2 and generating linear and log-indexed tabular fluid data for use within GAS-PASS/CO_2.

Chapter 3 presents the methods and models used to simulate turbomachinery performance. It will focus upon accurately modeling turbomachinery performance curve data generated with detailed turbomachinery design codes with S-CO_2.

Chapter 4 presents the methods and models used to calculate heat transfer and pressure drop in the printed circuit heat exchangers used in this cycle. The emphasis of the chapter is on the development of a computational model appropriate for use in a transient performance code.

Chapter 5 presents the overall GAS-PASS/CO_2 code. It will provide a description of the code, the integration of the various methods and models addressed in *Chapters 2-4*, and a description of the techniques used to allow a Newtonian solver to be used with a highly non-linear system.

Chapter 6 analyses transient simulation results currently available. This chapter will demonstrate the unique behavior arising from non-linear fluid properties coupled to turbomachinery performance curves, and offer analysis of three quasi-step-change transients showing the major characteristics of system behavior.

Chapter 7 presents a report summary and addresses near term future work.

Appendix A: GAS-PASS/CO2 Equations compiles the differential/difference equations which constitute the mathematical underpinnings of the GAS-PASS/CO_2 code.

Appendix B: Roots of Nonlinear Systems briefly summarizes available mathematical techniques and computational codes available to solve non-linear systems of equations.

Appendix C: References documents the references listed in this report.

2 Fluid Properties

This chapter provides methods to rapidly and accurately introduce real fluid properties into the GAS-PASS code. Previously the GAS-PASS code simulated only ideal fluids like helium, thus allowing the use of the Ideal Gas Law for most fluid properties. The property requirements for the S-CO_2 cycle are considerably more complex.

The key challenges of properties in this cycle are:

- S-CO_2 shows highly non-linear behavior near its critical point which, by design, is a key area of operation in this cycle. These non-linear changes are fast enough that solving the equation of state may become challenging both in machine precision and in root finding as described in *Section 2.2*.
- The difficulty of using a Newtonian solver with a non-linear system requires solving the system of equations using enthalpy instead of temperature as addressed in detail in *Chapter 5*. Fluid properties must be based upon enthalpy and pressure (instead of temperature and pressure) but there is a also a need to convert back and forth with temperature in the heat exchangers.
- The complexity of heat exchange with S-CO_2 requires detailed heat transfer and pressure drop calculations, preventing the use of simple but common relations like log-mean temperature (addressed in detail in *Chapter 4*). These detailed calculations require thermodynamic transport properties like thermal conductivity and viscosity. Even if a simple relation like the Ideal Gas Law could be developed for S-CO_2 one would have to find a way to provide transport properties.
- Simulating the turbomachinery performance curves requires converting to and from entropy. The turbomachinery efficiencies are expressed in isentropic efficiency (as is typical) so the property code must be able to calculate entropy using enthalpy and pressure then to calculate back to enthalpy using entropy and pressure.
- Besides S-CO_2 it is necessary to accurately simulate liquid water for heat rejection and liquid sodium for heat addition. Since these fluids are used for heat exchange with S-CO_2 one must also be able to calculate their transport properties.

In summary, the fluid properties calculations must be robust, very accurate, able to convert to and from a wide variety of properties, and able to use a variety of fluids. While this is a challenging list already, the most difficult requirement for fluid property simulation is that these requirements have to be met very quickly to minimize computational time.

When running a typical transient simulation GAS-PASS/CO_2 may easily make over 10^9 calls to the fluid property routine. The vast majority of code runtime will be spent in the fluid property routines, therefore it is key that this code run rapidly if the ultimate design code continues to serve its purpose of rapid scoping calculations.

This chapter will explain the developments that were incorporated into GAS-PASS/CO_2 to meet the above requirements. The chapter is subdivided into four sections:

> *Section 2.1* provides a brief background into the non-linear nature of S-CO_2 fluid properties near the critical point
>
> *Section 2.2* addresses the use and the improvements made to the ultimate source of S-CO_2 fluid properties, NIST RefProp.
>
> *Section 2.3* details how the fluid properties are put into linear and log-indexed fluid property tables.
>
> *Section 2.4* summarizes this chapter.

2.1 Non-linear S-CO_2 Fluid Property Behavior

Carbon dioxide shows highly non-linear property behavior near its critical point (30.978°C, 7.3773 MPa). This non-linear behavior is one of the key enabling features for the S-CO_2 recompression cycle but it presents challenges for the fluid property source code. The reference design for this report is shown in *Figure 2-1*.

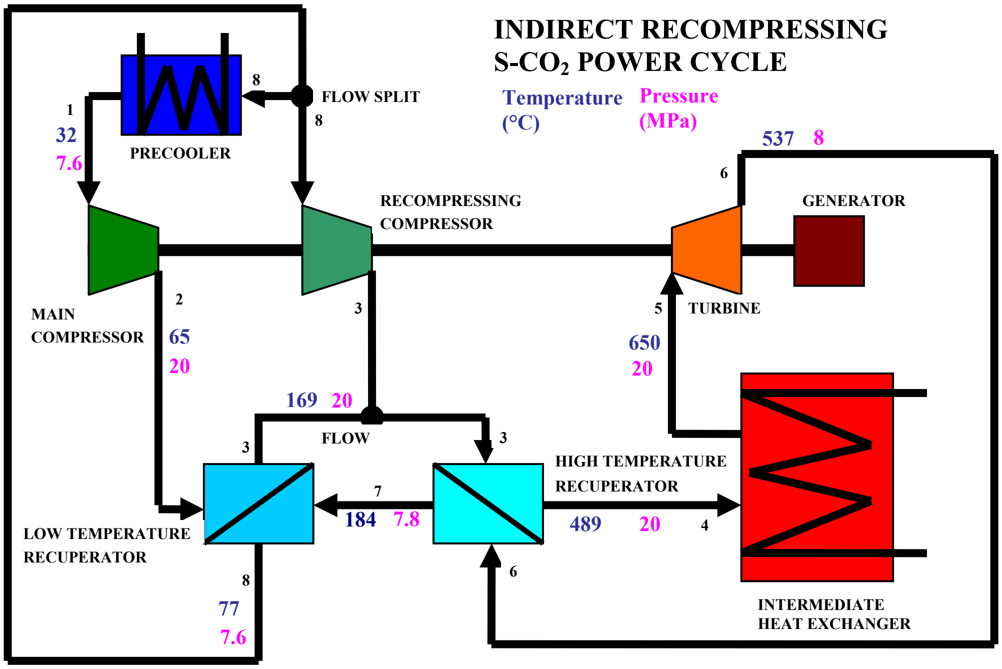


Figure 2-1: S-CO_2 Recompression cycle Diagram

The cycle design points (component outlet states numbered 1-8) in *Figure 2-1* are plotted on a density versus temperature diagram in *Figure 2-2*. The reader should note both the wide spread of densities within the cycle (for a single phase fluid) and especially note the very steep increase in fluid density at lower pressures as the fluid nears it critical temperature. It should be apparent from the plot that a small change in fluid temperature or pressure at this point of the cycle results in a large change in fluid density.

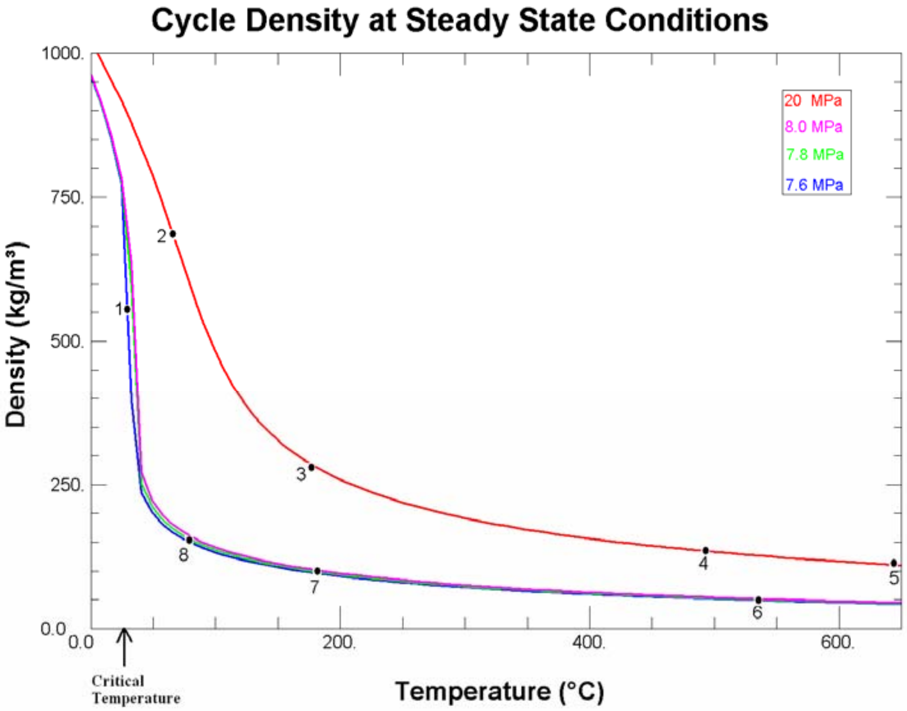


Figure 2-2: S-CO_2 Cycle Density

This impression is further reinforced when one looks at other fluid properties such as isobaric specific heat. The cycle design points from *Figure 2-1* are plotted on a isobaric specific heat versus temperature diagram in *Figure 2-3*. Compared to the density figure this property shows an even steeper rise as the fluid approaches its critical point. This large increase in isobaric specific heat is a primary reason for the complexity of the heat exchanger calculations described in detail in *Chapter 4*.

One of the reasons for the sharp behaviors shown in the property diagrams is this cycle's proximity to the critical point. When one plots the cycle diagram operation points on a temperature entropy plot, the bottom of the cycle is visually indistinguishable from the critical point, as shown in *Figure 2-4*.

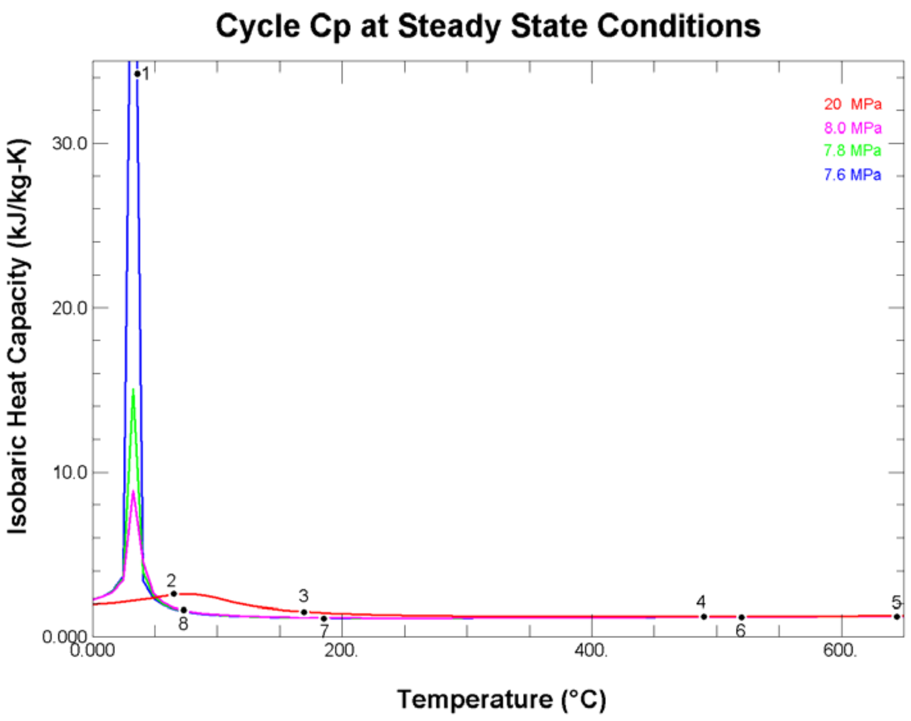


Figure 2-3: S-CO2Cycle Isobaric Specific Heat

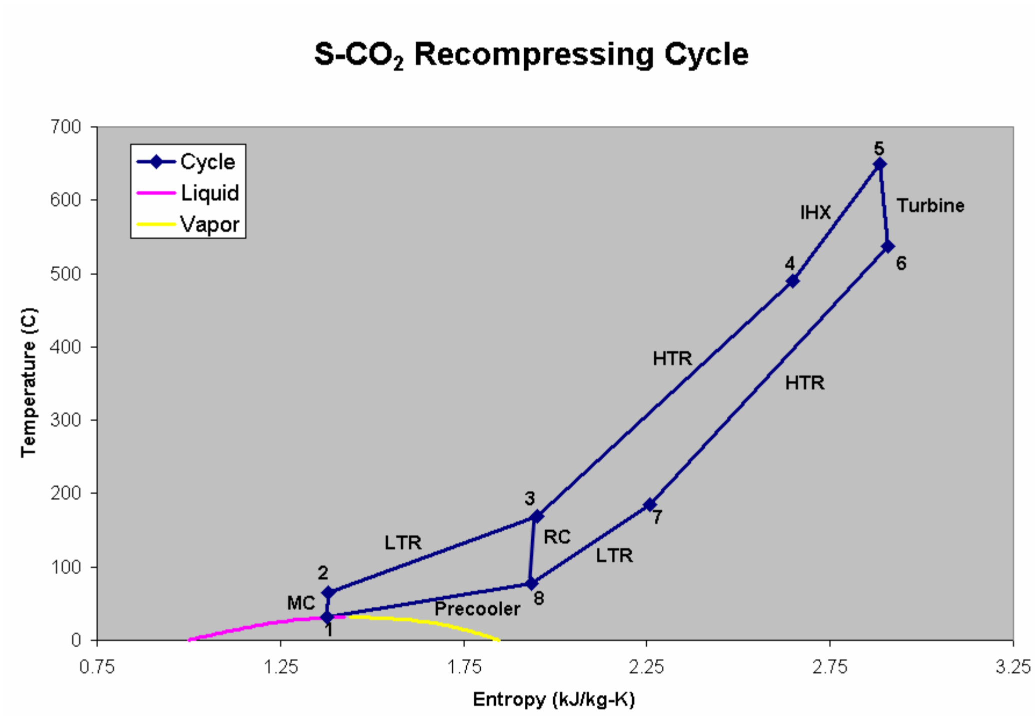


Figure 2-4: S-CO$_2$ Cycle Temperature vs. Entropy Diagram

Unfortunately, (from a property calculation standpoint), CO_2 shows unique behavior in several parts of this cycle. While non-linear behavior is expected close to the critical point, it can also be manifested at pressures well above the critical pressure and temperatures moderately above the critical temperature. An example of this behavior shows up clearly when one plots the main compressor operating region on an enthalpy and entropy diagram, as shown in *Figure 2-5*.

Figure 2-5 clearly shows that there is a very narrow region where the main compressor operates in enthalpy and entropy space compared to temperature and pressure space. Small changes in enthalpy and especially entropy lead to large changes in temperature and pressure. Therefore, fluid property conversions must be very accurate to avoid introducing significant error.

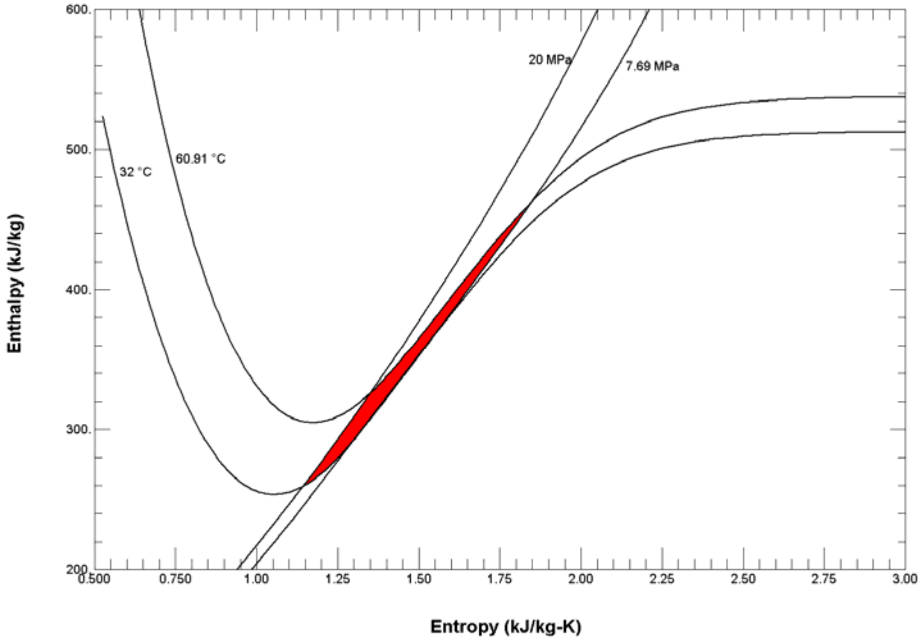


Figure 2-5: Main Compressor Enthalpy vs. Entropy Region

To summarize, CO_2 fluid properties show highly non-linear behavior in the S-CO_2 recompression cycle. This behavior is concentrated near the fluid's critical point but can also present itself in other regions such as the outlet of the main compressor. The sharpness of this behavior requires highly accurate property conversions thus making large demands upon the equation of state.

2.2 Using NIST RefProp with S-CO_2

The ultimate source of real fluid property for GAS-PASS/CO_2 is the NIST RefProp code[7]. The NIST RefProp code can accurately calculate a variety of fluid properties for numerous fluids. The documentation included with the code states:

> REFPROP is based on the most accurate pure fluid and mixture models currently available. It implements three models for the thermodynamic properties of pure fluids: equations of state explicit in Helmholtz energy, the modified Benedict-Webb-Rubin equation of state, and an extended corresponding states (ECS) model. Mixture calculations employ a model which applies mixing rules to the Helmholtz energy of the mixture components; it uses a departure function to account for the departure from ideal mixing. Viscosity and thermal conductivity are modeled with either fluid-specific correlations or an ECS method.[8]

The RefProp code was selected for a variety of reasons but primarily because of its accuracy and wide applicability with carbon dioxide. Other authors have modeled CO_2 properties from a "pseudo-perfect" gas model that fixes the ideal gas law with compressibility factors. Aungier makes a compelling case[9] for using these very rapid calculations in compressor design.

Unfortunately, these pseudo-perfect models require constants that are not readily available, compressibility factors should be recalculated within turbomachinery stages[10] (a level of detail not desired in GAS-PASS/CO_2), and, in general, will not be as accurate as the NIST RefProp models. Furthermore, the varied requirements of simulating fluid properties for simulation of the recompression cycle prohibit the use of a simplified equation.

Therefore, NIST RefProp was selected as GAS-PASS/CO_2's ultimate property source whenever possible. This section will provide a brief background of key RefProp features, recommend an equation of state, and warn the user of several problems encountered and solutions found when calculating S-CO_2 properties with RefProp.

2.2.1 Key NIST RefProp Features

RefProp provides the ability to calculate numerous different fluid properties including common properties like enthalpy to more exotic properties like fugacity. More importantly, it allows the user to input various properties to get the desired value e.g. one may use enthalpy and entropy to calculate temperature. RefProp accomplishes this with an equation of state in temperature and pressure by performing numerical convergence iterations when non-temperature and pressure values are entered. For simulating the S-CO_2 recompression cycle RefProp provides far more property flexibility than is required.

RefProp 7.0 also features the a choice of simulating 52 different fluids as shown in *Table 2-1*. While not offering liquid sodium, RefProp offers all other fluids necessary, including water.

Number	Symbol	Fluid	Molecular Weight [g/cm^3]	Description
1	AMMONIA	AMMONIA	17.03026	
2	ARGON	ARGON	39.948	Noble Gas
3	BUTANE	BUTANE	58.1222	Lighter Fluid
4	CO2	Carbon Dioxide	44.0098	
5	CO	Carbon Monoxide	28.011	
6	D2	Deuterium	4.0282	
7	ETHANE	ETHANE	30.070	
8	ETHYLENE	Ethylene	28.05376	
9	FLUORINE	Fluorine	37.99681	
10	HELIUM	Helium	4.0026	Noble Gas
11	HEPTANE	Heptane	100.202	
12	HEXANE	Hexane	86.17536	
13	HYDROGEN	Hydrogen	2.01594	
14	ISOBUTAN	ISOBUTANE	58.1222	
15	METHANE	METHANE	16.0428	Natural Gas
16	NITROGEN	NITROGEN	28.01348	Noble Gas
17	NEON	Neon	20.179	
18	NF3	Nitrogen Tri-Fluoride	71.019	
19	OXYGEN	OXYGEN	31.9988	
20	PROPANE	PROPANE	44.0956	Hydrocarbon
21	PROPYLEN	PROPYLENE	42.0804	
22	PARAHYD	Parahydrogen	2.01594	
23	PENTANE	Pentane	72.14878	
24	R11	R11	137.368	Refrigerant
25	R113	R113	187.375	Refrigerant
26	R114	R114	170.921	Refrigerant
27	R115	R115	154.4667	Refrigerant
28	R116	R116	138.01	Refrigerant
29	R12	R12	120.913	Refrigerant
30	R123	R123	152.931	Refrigerant
31	R124	R124	136.4762	Refrigerant
32	R125	R125	120.022	Refrigerant
33	R13	R13	104.459	Refrigerant
34	R134A	R134A	102.032	Refrigerant
35	R14	R14	88.0046	Refrigerant
36	R141B	R141B	116.95	Refrigerant
37	R142B	R142B	100.495	Refrigerant
38	R143A	R143A	84.041	Refrigerant
39	R152A	R152A	66.051	Refrigerant
40	R218	R218	188.019	Refrigerant
41	R22	R22	86.468	Refrigerant
42	R227EA	R227EA	170.0289	Refrigerant
43	R23	R23	70.01385	Refrigerant
44	R236EA	R236EA	152.03928	Refrigerant
45	R236FA	R236FA	152.0393	Refrigerant
46	R245CA	R245CA	134.04882	Refrigerant
47	R245FA	R245FA	134.04882	Refrigerant
48	R32	R32	52.024	Refrigerant
49	R41	R41	34.033	Refrigerant
50	RC318	RC318	200.0312	Refrigerant
51	WATER	WATER	18.015268	
52	Xenon	Xenon	131.3	

Table 2-1: NIST RefProp 7.0 Available Fluids

Of importance to future work is the ability of RefProp to calculate fluid mixtures. Fluid mixtures like helium and nitrogen Brayton cycles are an area of active research, and mixtures also allow the simulation of plant performance during accidents such as air ingress. RefProp 7.0 supports mixing up to 20 fluids and includes a mixtures file, *Hmx.bnc*, which provides the mixture rules. The mixtures supported are somewhat limited at the time of this writing but common combinations with fluids like helium, nitrogen, and carbon dioxide are available.

2.2.2 RefProp CO_2 Equation of State

RefProp allows the user to specify the equation of state. This subsection will compare the Benedict-Webb-Rubin, BWR, model to the default Span and Wagner Helmholtz model, FEQ, using CO_2 density since this is the key property of interest. The RefProp CO_2 fluids file, *co2.fld*, states that the FEQ model is more accurate in the critical region of CO_2 while the BWR model is recommended for general use.

Data points ranging from 300 K to 440 K (the maximum temperature in the BWR model is 440.1 K) at every 0.5 K and at each integer MPa value between 1 MPa and 20 MPa were tested.

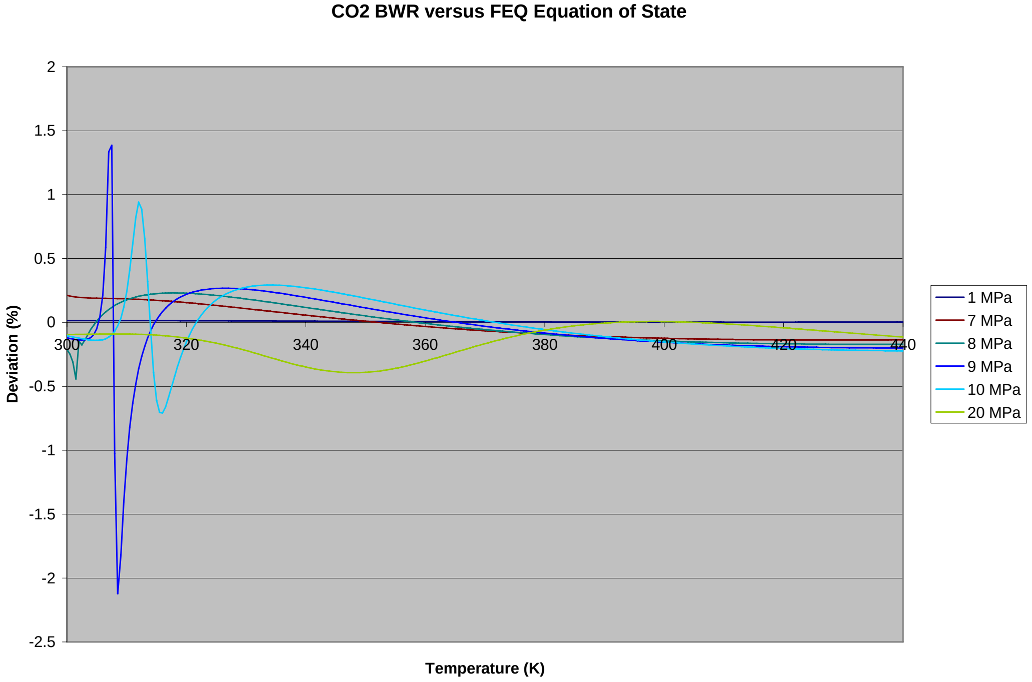


Figure 2-6: CO_2 BWR versus FEQ Equations of State

The results may be seen in *Figure 2-6* (only a few pressure lines are shown for clarity) showing the percent difference between the BWR and FEQ models. The reader should note that there are systematic differences but they are relatively small (note the percent scale). The differences are centered near the critical temperature and slightly above the critical pressure but persist in all the data.

Due to the relatively small differences between these model values, the limited temperature range of the BWR model: to only 440.1 K (the FEQ model goes to 1100 K, a realistic temperature for possible applications), the emphasis on the critical region, and the difficulty in coding a variable model program, only the NIST default model, FEQ, will be used.

2.2.3 RefProp CO_2 Caveats

While RefProp accurately calculates CO_2 properties in the vast majority of circumstance several problems near the critical point have been discovered. In all known cases of failure the RefProp code has provided the user with error flags and warning messages.

Known RefProp problems with S-CO_2 are summarized in *Table 2-2*.

Table 2-2: Known RefProp CO_2 Problems

Version	Problem	Solution
5.0	Non-convergence near CO_2 critical point.	Modified Regula-Falsi root finding technique provided by Dr. Pavel Hejzlar.
7.0	Log error near CO_2 critical point when using enthalpy and entropy inputs.	This was caused by a lack of machine precision coupled with an non-optimal solution technique. It was already fixed in the next version, 7.1 beta, when the problem was discovered.
7.1	Non-convergence very close to CO_2 critical point using non-temperature and pressure inputs.	The modified Regula-Falsi root finding algorithm cannot converge some CO_2 data points. An updated algorithm and patch were provided by Nate Carstens.

The first problem listed in the table was fixed several years ago and the main code now features the solution. The second problem in the table has a fix available for those willing to use the current beta code version. The final problem deservers further attention.

RefProp iteratively converges non-temperature and pressure roots with numerical root finding algorithms, currently a modified Regula-Falsi. While this works well in the vast majority of cases, S-CO_2 is non-linear enough that this technique will not work very close to the critical point.

A solution was developed that changed the root finding algorithm to Brent's method[11]. Brent's method uses inverse quadratic interpolation and the secant method (when possible for speed) and falls back on the bisection method when progress isn't made.

When provided with bracketing roots, convergence is guaranteed for continuous functions.

This method has been applied to RefProp 7.1 and a patch was sent to RefProp's developer. To date this new code has converged in every case tried including very close to the CO_2 critical point and with a variety of fluids and properties. The new code also runs slightly faster than the previous version.

To summarize, when patched the NIST RefProp code provides the required accuracy for all of the fluids (except liquid sodium) and properties required to simulate the S-CO_2 recompression cycle. It is the source of all water and CO_2 fluid properties used in GAS-PASS/CO_2.

2.3 Tabular Fluid Properties

The benefits of the NIST RefProp code come at the cost of considerable complexity and therefore slow runtime. When GAS-PASS was originally updated to allow real fluid properties it used RefProp directly. Unfortunately, this slowed code runtime by quite a few orders of magnitude. While runtime isn't a primary concern, GAS-PASS is designed as a rapid scoping tool and this large an increase in runtime would negate much of its rationale.

The solution adopted for GAS-PASS/CO_2 is to pre-compute fluid property data and store it in tabular form. The user can still have the flexibility to create tabular data for any fluid or fluid mixture and can create tables with any input or output properties. Tables also offer the advantage of easily combining disparate property sources like simple liquid sodium polynomials with the very complex RefProp carbon dioxide Helmholtz equation of state. Finally, tables offer the advantage of unrestricted distribution, whereas NIST RefProp is a commercial code.

GAS-PASS/CO_2 uses indexed property tables to save space and avoid searching. An indexed table contains only the desired property data but by carefully setting up the table one may know the location of the desired data a priori. For example, a table might be indexed by temperature and pressure and contain 1,001 temperature indexes between 0 and 1000 K and 1,001 pressure indexes between 100 kPa and 10,100 kPa with a density calculated at each point. When the interpolation routine is called with a desired temperature and pressure it calculates the position in the table where those values would lie. Once the points surrounding the desired value are found the approximate value is calculated via double interpolation as shown in *Equation 2-1*.

Equation 2-1: Double Linear Interpolation

$$x_t = {(t - T_{low})} \Big/ {(T_{high} - T_{low})}$$

$$x_p = {(p - P_{low})} \Big/ {(P_{high} - P_{low})}$$

$$r_{p-low} = V(P_{low}, T_{low}) + x_t * (V(P_{low}, T_{high}) - V(P_{low}, T_{low}))$$

$$r_{p-high} = V(P_{high}, T_{low}) + x_t * (V(P_{high}, T_{high}) - V(P_{high}, T_{low}))$$

$$V(t, p) = r_{p-low} + x_p * (r_{p-high} - r_{p-low})$$

This code is very fast and can be implemented with little overhead, especially compared to a sorting algorithm.

The user should be aware that GAS-PASS/CO_2 will not extrapolate and returns an error flag if properties are requested outside tabular bounds. This will become important during Jacobian calculation addressed in detail in *Chapter 5*. If necessary, the user may simply recreate the appropriate data table with larger bounds to avoid this problem.

2.3.1 Linearly Indexed Property Tables

The obvious way to index tabular data is linearly. This creates a constant spacing between values and makes it trivial to calculate the appropriate data position within the table by knowing the minimum and maximum index values and grid spacing. When this method was implemented in GAS-PASS the runtime improved significantly compared to the NIST RefProp version as shown in *Table 2-3*. The reader should note that this is an optimal RefProp runtime using assumptions that probably couldn't be maintained during transient simulation.

Table 2-3: Gas-Pass Run Times by Data Source

Run	Time (sec.)	% Change to Original
Ideal Gas Law	236	0
NIST RefProp	15730*	6665
Linearly Indexed Tables	444	88.1

*Estimated from 10% run time

2.3.2 Log Indexed Property Tables

Unsurprisingly, the highly non-linear behavior of S-CO_2 properties leads to difficulty when used with linearly indexed tables. The first CO_2 tables were linearly indexed and numerical problems were encountered during simulation that prevented solver convergence. Upon investigation it was discovered that during the calculation of the main compressor outlet pressure the property tables were creating errors of over 1 MPa.

The reason for this error can be visually seen in *Figure 2-5*. Despite using millions of data points in the property table, the non-linear behavior of the fluid properties in the main compressor required a much tighter table spacing to avoid large property conversion errors when converting from values of enthalpy and entropy to pressure. Unfortunately, tables with this tight of a grid spacing would be far too large to be practical.

Using a variable mesh would effectively solve this problem at the cost of requiring a searching routine to find the appropriate position in table. The large increase in computation time (compared to indexed tables) would remove much of the advantage of using tabular data in the first place. Log-indexed property tables offer many of the benefits of both methods with few drawbacks.

2.3.2.1 Background

One significant advantage of non-linear CO_2 behavior, is that its challenging areas are well defined and furthermore lie in only one region. In many complex problems that would benefit from a variable mesh, it is very difficult to predict the region, or more probably regions, of difficulty. One may take advantage of this specific knowledge by using a non-linearly indexed property data table.

Carbon dioxide’s properties become highly non-linear near its critical point, thus any property table would benefit from a larger number of data points in this area. For the S-CO_2 recompression cycle the temperatures and pressures of interest range from the critical point upwards.

2.3.2.2 Log Function

If one were to plot the desired number of data points versus property value it would create a function that beginning at the critical point bends sharply upward then tapers off towards a limit. The most common simple function that resembles this shape is a natural log. This is clearly shown in *Equation 2-2 & Equation 2-3.*

Equation 2-2: Number of Points as a Function of Property Value

$$Np = \alpha * \ln(P*) + \beta$$
$$P* = P - \gamma$$

Equation 2-3: Property Value as a Function of the Number of Points

$$P* = e^{\frac{Np-\beta}{\alpha}}$$
$$P = P* + \gamma$$

Equation 2-2 can calculate the desired data point a given property value will be at. This will not be exact because we are not using an infinite number of data points but with expected mesh densities the answer should be close. Correspondingly, when creating the data tables, *Equation 2-3* may be used to calculate the property value for a given data point.

The requirements of rapidly solving turbomachinery and converging a non-linear system with a Newtonian solver dictate (discussed in detail in *Chapter 5*) that pressure, enthalpy, and entropy are the properties that must be log-indexed.

2.3.2.3 Property Indexes

The outlined process will be used for CO_2 pressure in the S-CO_2 recompression cycle by using *Equation 2-2* & *Equation 2-3* to find appropriate coefficients. Experimentation has shown that a multiplicative coefficient, alpha, near 100 provides a reasonable degree of high initial density and reasonable end of table density. The value of beta and gamma may be iteratively solved for. First, by deciding upon the number of data points in the table, i.e. 500, and then solving for beta by setting the last point to be equal to the largest desired property value. Then one may solve for gamma by setting point 1 equal to the minimum desired pressure. By iterating several times to keep both points close to their targets one may arrive at the desired values, which for the S-CO_2 cycle are:

$$\alpha = 100.0$$

$$\beta = -1194.1$$

$$\gamma = 7230035$$

One finds that by starting near 7,385 kPa (note that the coefficients are sized in terms of Pascals), where there is a data point every 1.56 kPa, that there are 500 data points by 30,000 kPa, where there is one data point every 226.6 kPa. This encompasses our range of interest, provides more detail in the critical region, and provides a reasonable guess of an appropriate mesh density. The total number of data points per unit pressure is graphically shown in *Figure 2-7*.

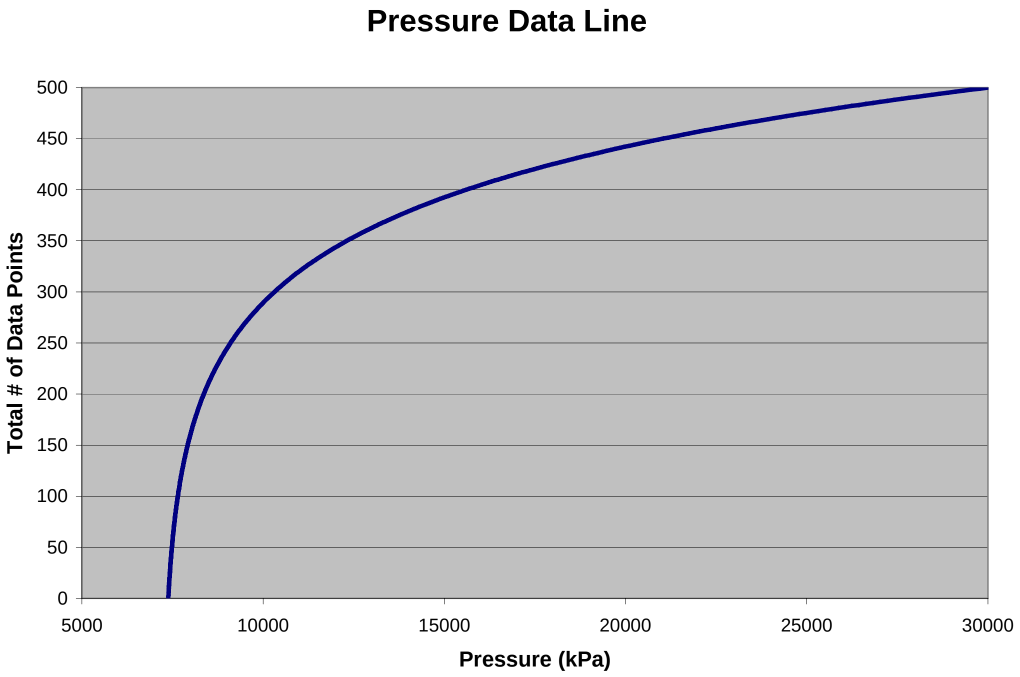


Figure 2-7: Pressure Data Density

A similar process may be used with enthalpy. Using the previously equations shown for enthalpy (in J/kg units) we find:

$$\alpha = 100.0$$
$$\beta = -889.555$$
$$\gamma = 267627$$

Using these values, the table begins at 275 kJ/kg, where there is one data point every 0.074 kJ/kg. There are 500 data points by the maximum table value at 1351 kJ/kg, where there is one data point every 10.78 kJ/kg. It should be noted that enthalpy is also a function of pressure and thus this table must cover both expected temperatures and pressures.

While temperature determines enthalpy more strongly than pressure, the wide range of pressures of interest above the critical pressure (7.385-30 MPa) provides a significant enthalpy range. For example, at the critical point the enthalpy is 326.08 kJ/kg. However, at 31°C (just above T_c=30.978°C) and 30 MPa the enthalpy is 242.68 kJ/kg. To be conservative, this table starts at 31°C and 10 MPa which has an enthalpy of 275 kJ/kg. This trades table density of pressures near the critical point for table completeness. Future use should establish the correctness of this tradeoff. The upper bound is simply set to 800°C and 30 MPa which are well above expected operating parameters. The plot of the number of data points versus enthalpy is shown in *Figure 2-8*.

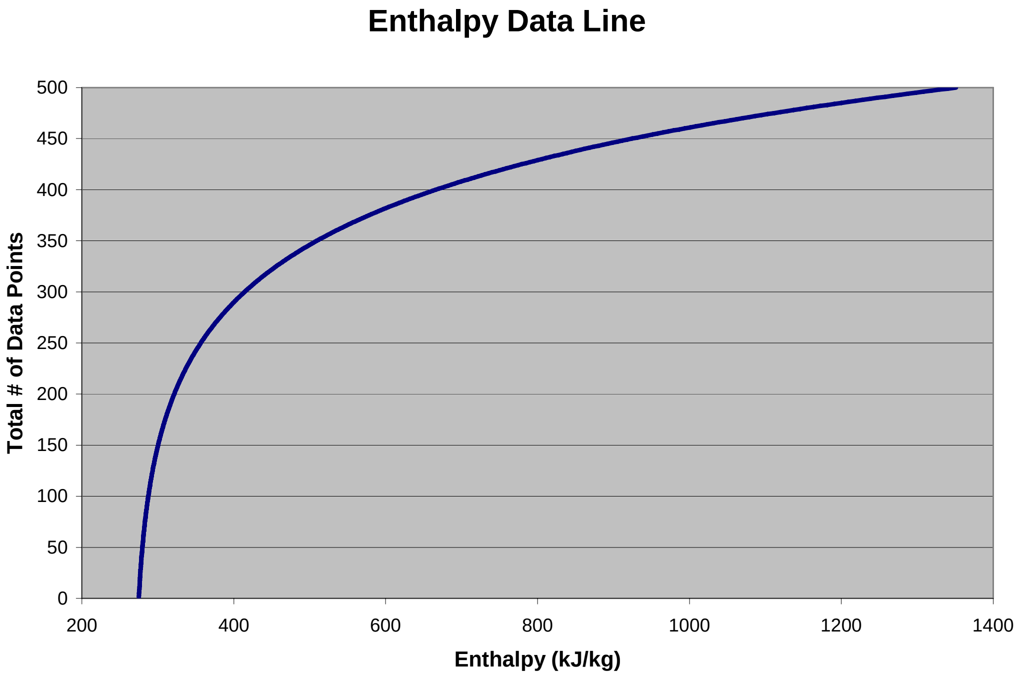


Figure 2-8: Enthalpy Data Density

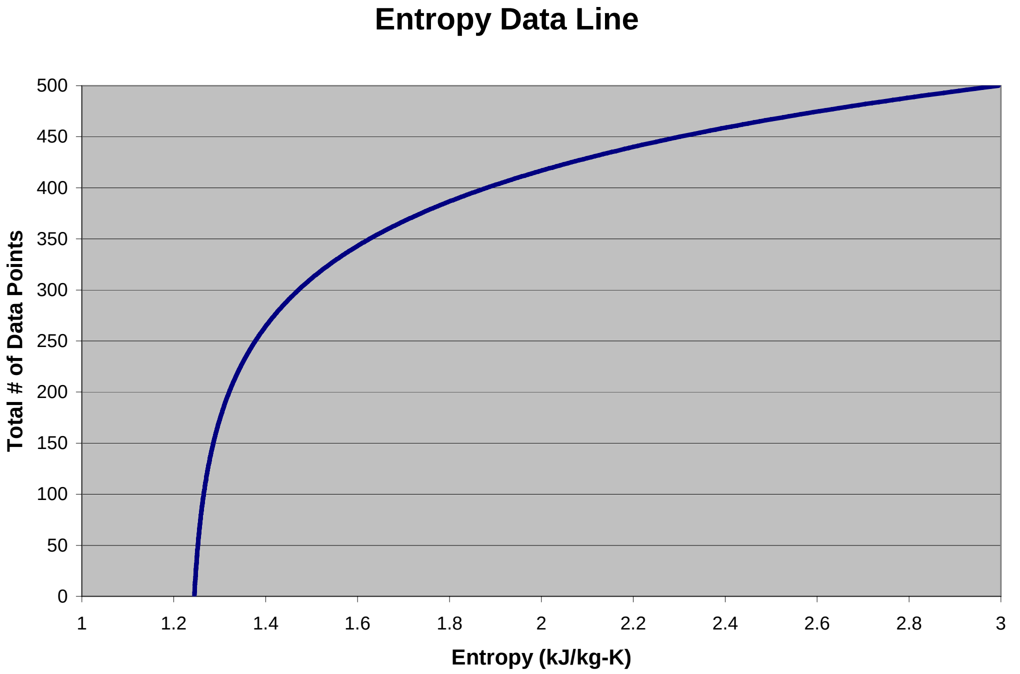


Figure 2-9: Entropy Data Density

A similar process may be used with entropy. Using the previously shown equations shown for enthalpy (in J/kg-K units) we find:

$\alpha = 100.0$

$\beta = -247.418$

$\gamma = 1232.809$

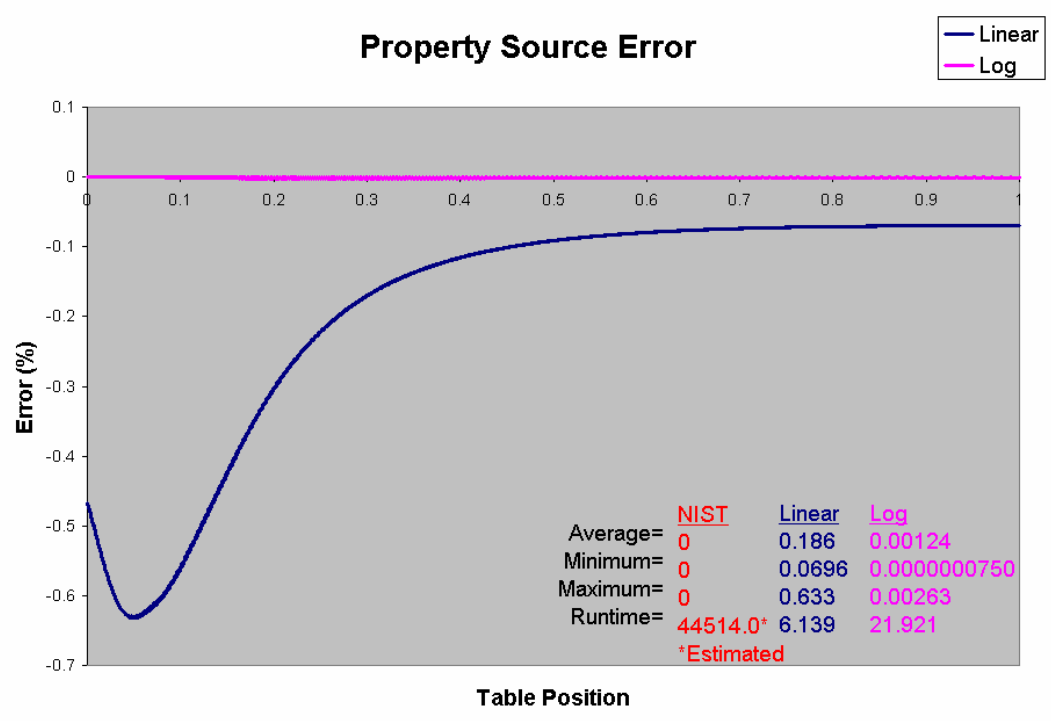
This table encompasses the same range as the minimum to maximum pressures and enthalpies as previously noted. The plot of the number of data points versus entropy is shown in *Figure 2-9*.

These tables encompass our range of interest, provide greater detail in the critical region, and provide a reasonable guess of an appropriate mesh density.

2.3.2.4 Speed and Accuracy

The speed and accuracy of the log-indexed tables were compared to linearly-indexed tables. The linearly-indexed tables had approximately 4x as many data points as the log-indexed tables and covered the same range (the S-CO_2 recompression cycle with extra margin). Virtually the full range of both tables was tested and compared to the NIST RefProp code to assess their accuracy.

Figure 2-10 shows the error of the log-indexed properties versus the error of the linearly indexed properties. It is apparent that log-indexed properties are much more accurate (150x more accurate on average) especially near the critical region (towards the beginning of the table) despite the much larger amount of data in the linear table. Unfortunately, the log tables ran approximately 3.6x slower than the linear tables (a log is more difficult to calculate than a linear function). However, the log tables were still about 2,300x faster than the NIST RefProp code with default inputs.



	NIST	Linear	Log
Average=	0	0.186	0.00124
Minimum=	0	0.0696	0.0000000750
Maximum=	0	0.633	0.00263
Runtime=	44514.0*	6.139	21.921

*Estimated

Figure 2-10: Relative Tabular Error

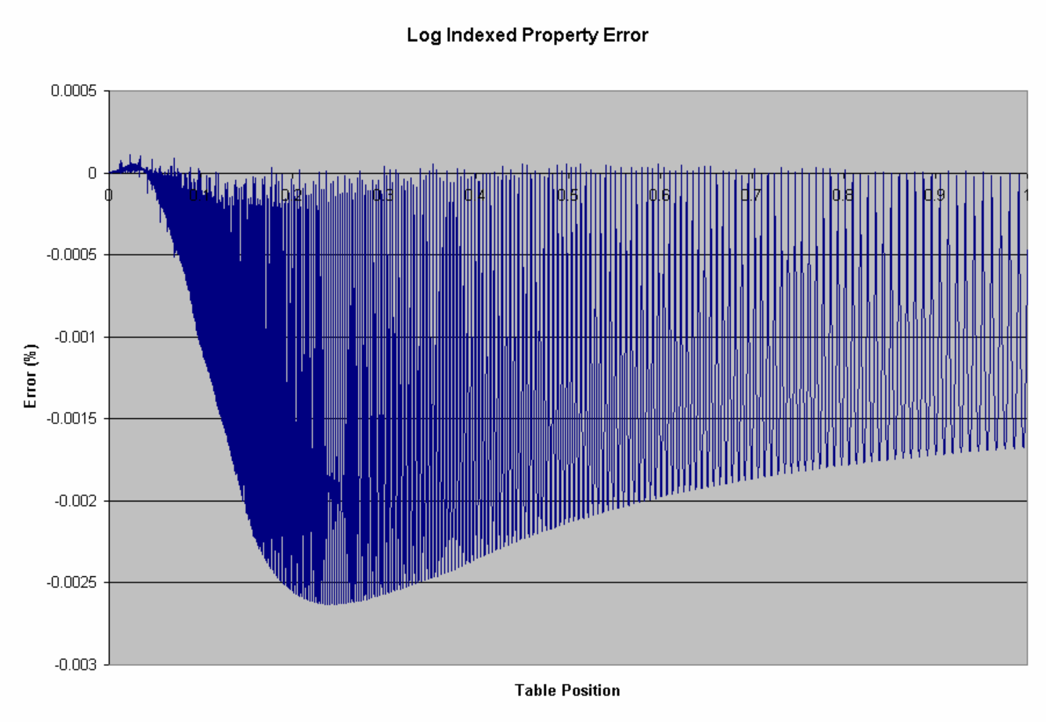


Figure 2-11: Absolute Log-Indexed Tabular Error

The error present in the log-indexed tables is more clearly seen in *Figure 2-11*. The critical region is extremely accurate but the log function, as currently configured, does not place enough emphasis on the transition between the critical region and higher values. One might desire approximately the same error throughout the table but this table appears appropriate for our purpose where the critical region is the key area of interest.

In any case, a maximum error of less than 0.003% is probably more accuracy than necessary for dynamic analysis. Unless future needs arise, these tables will be considered complete.

2.33 GAS-PASS/CO_2 Property Tables

The complete list of fluid property tables required for GAS-PASS/CO_2 is shown in *Table 2-4*. While this list is specific to the current plant model it should be readily applicable to any indirect power cycle; only the property table needs to be updated. For example, the liquid sodium tables could be replaced by helium tables using the exact same input and output properties to simulate a gas-to-gas heat exchange cycle.

Table 2-4: GAS-PASS/CO_2 Property Tables

Fluid	Input Properties	Output Property	Number
Carbon dioxide	Enthalpy/Pressure	Isobaric Specific Heat	1
Carbon dioxide	Enthalpy/Pressure	Density	2
Carbon dioxide	Enthalpy/Pressure	Internal Energy	3
Carbon dioxide	Enthalpy/Pressure	Thermal Conductivity	4
Carbon dioxide	Enthalpy/Pressure	Temperature	5
Carbon dioxide	Enthalpy/Pressure	Viscosity	6
Carbon dioxide	Temperature/Pressure	Enthalpy	7
Carbon dioxide	Enthalpy/Pressure	Entropy	8
Carbon Dioxide	Entropy/Pressure	Enthalpy	9
Water	Enthalpy/Pressure	Isobaric Specific Heat	1
Water	Enthalpy/Pressure	Density	2
Water	Enthalpy/Pressure	Internal Energy	3
Water	Enthalpy/Pressure	Thermal Conductivity	4
Water	Enthalpy/Pressure	Temperature	5
Water	Enthalpy/Pressure	Viscosity	6
Water	Temperature/Pressure	Enthalpy	7
Sodium	Enthalpy/Pressure	Isobaric Specific Heat	1
Sodium	Enthalpy/Pressure	Density	2
Sodium	Enthalpy/Pressure	Internal Energy	3
Sodium	Enthalpy/Pressure	Thermal Conductivity	4
Sodium	Enthalpy/Pressure	Temperature	5
Sodium	Enthalpy/Pressure	Viscosity	6
Sodium	Temperature/Pressure	Enthalpy	7

The authors would like to note the significance of the input properties chosen. Enthalpy is used in place of temperature where possible to prevent the non-linearity of the temperature to enthalpy conversion being included in most solution equations as described in greater detail in *Chapter 4*.

Second, entropy and pressure are used in the turbomachinery performance curves to avoid two problems. If the turbomachinery performance curves used enthalpy change (instead of pressure ratio) one would need an enthalpy/entropy->pressure table that would introduce further non-linearity with two property conversions and require RefProp to perform a much more difficult double root find. Using pressure ratio performance curves avoids these problems and is the more common practice, therefore it was adopted for GAS-PASS/CO_2.

2.4 Fluid Property Summary

The highly non-linear behavior of carbon dioxide near its critical point and the varied requirements of simulating the S-CO_2 recompression cycle place difficult demands upon any fluid property source. When patched the NIST RefProp code can provide all of the necessary fluid (except liquid sodium) and property flexibility and accuracy required for simulation.

Combining RefProp output in indexed tables offers the added benefit of a reasonable runtime with little loss in accuracy or flexibility. While linear indexing of water and liquid sodium properties works well, S-CO_2 required log indexing to achieve the necessary accuracy in a reasonable manner.

Overall, these property methods and sources allow GAS-PASS/CO_2 to flexibly, accurately, and rapidly calculate fluid properties. This will enable this simulation code to continue to meet its design goal of being a rapid scoping tool.

3 Turbomachinery Performance Mapping

Due to the complexity of modeling turbomachinery GAS-PASS/CO_2 will use recomputed performance curves to estimate turbomachinery performance. This has the advantage of removing a significant computational burden from the transient code as well as allowing rapid changes to turbomachinery such as switching from radial to axial turbomachines by simply switching input files. Furthermore, it will allow accurate experimental data to be easily introduced into simulations as this data becomes available.

This chapter will detail how GAS-PASS/CO_2 handles the turbomachinery curves, show the currently used axial and radial (compressor only) performance curves, and derive a method to handle the S-CO_2 fluid property variation effect for the compressors.

3.1 Turbomachinery Performance Maps

The turbomachinery performance maps used in GAS-PASS/CO_2 are displayed in this section. The slopes of these maps are quite important during transient simulations since they will determine which direction pressures and enthalpies want to move to satisfy the turbomachinery conservation equations.

The radial compressor maps are based upon Dr. Yifang Gong's radial compressor report[12]. The radial main and recompressing efficiency maps are shown in *Figure 3-1* and *Figure 3-2* respectively. They show a definite efficiency peak with a relatively steep drop off on either side. The curves show a wider mass flow rate operating range at higher shafts speeds but are otherwise identical. This stems from extrapolating from a single data curve as detailed in the referenced report.

The radial main and recompressing compressor pressure ratio maps are shown in *Figure 3-3* and *Figure 3-4*. These curves have a flat pressure ratio at low mass flow rates that eventually smoothly slopes to a minimum at high mass flow rates. The shaft speed curves become every wider space and have ever wider mass flow rates as the speed increases. Once again, it is clear that these curves are physically extrapolated from a single shaft speed curve.

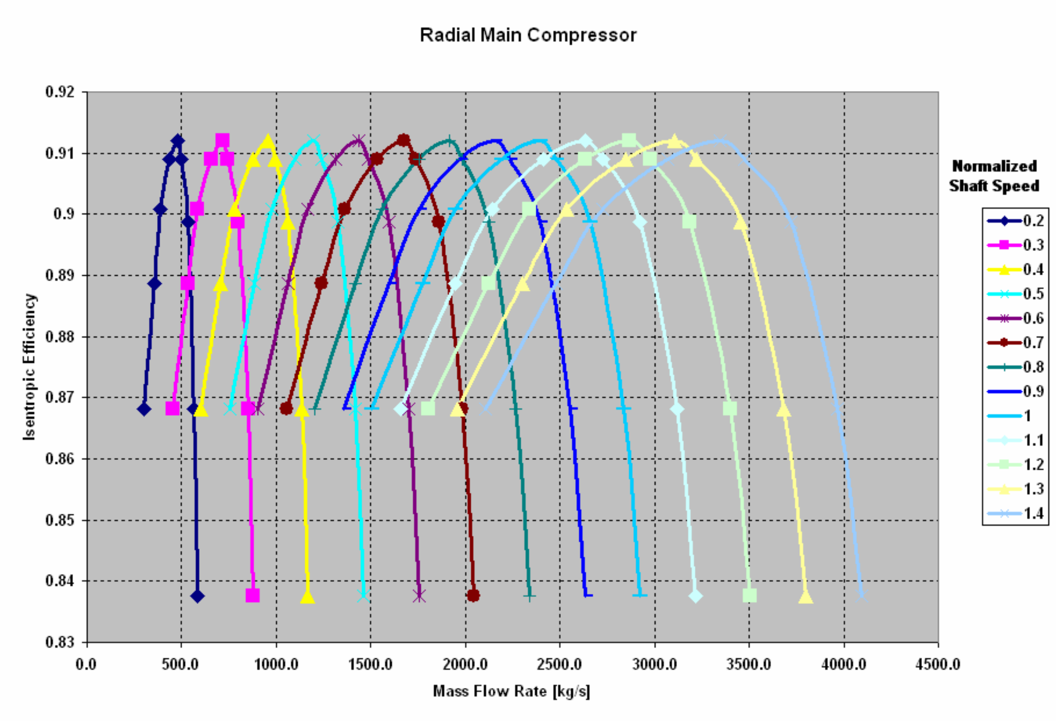


Figure 3-1: Radial Main Compressor Efficiency Map

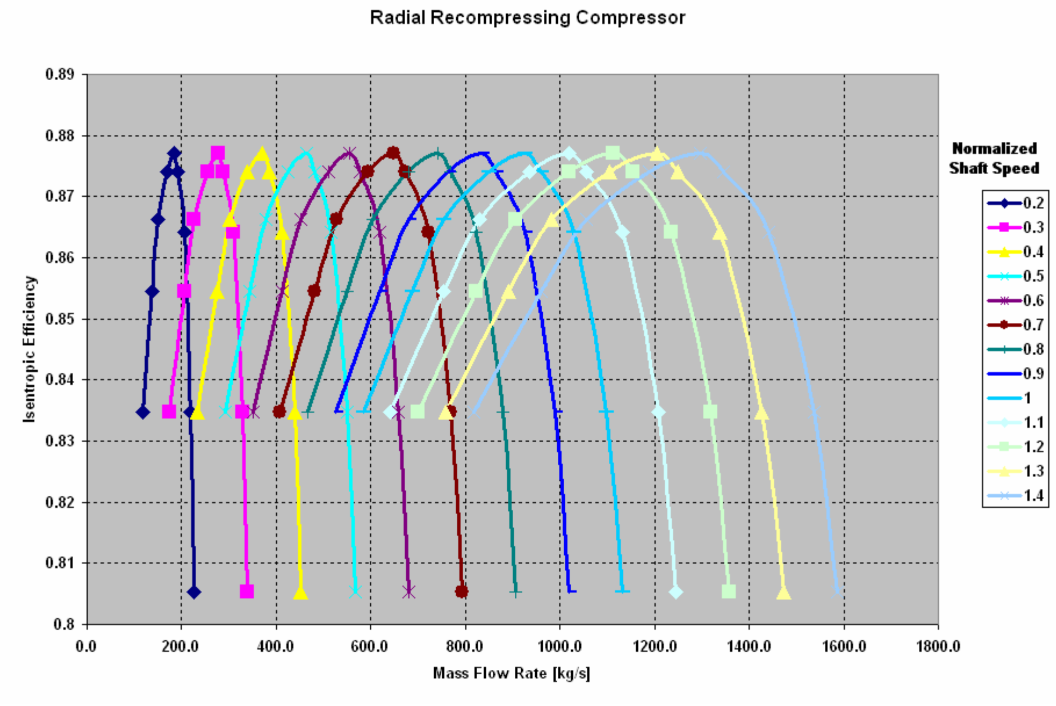


Figure 3-2: Radial Recompressing Compressor Efficiency Map

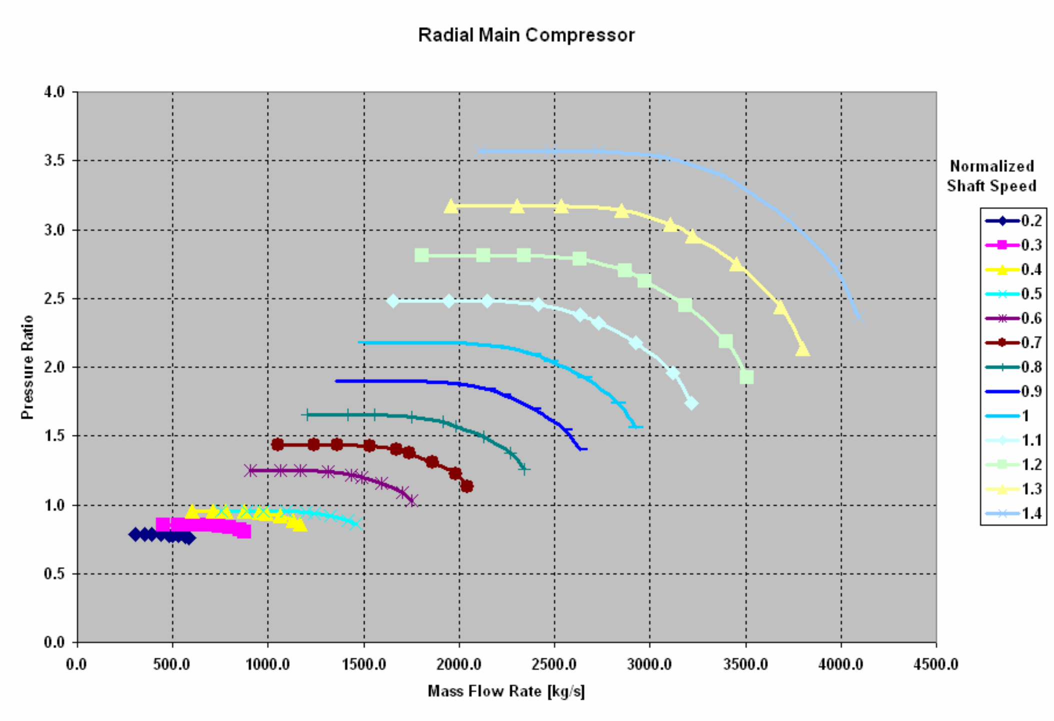


Figure 3-3: Radial Main Compressor Pressure Ratio Map

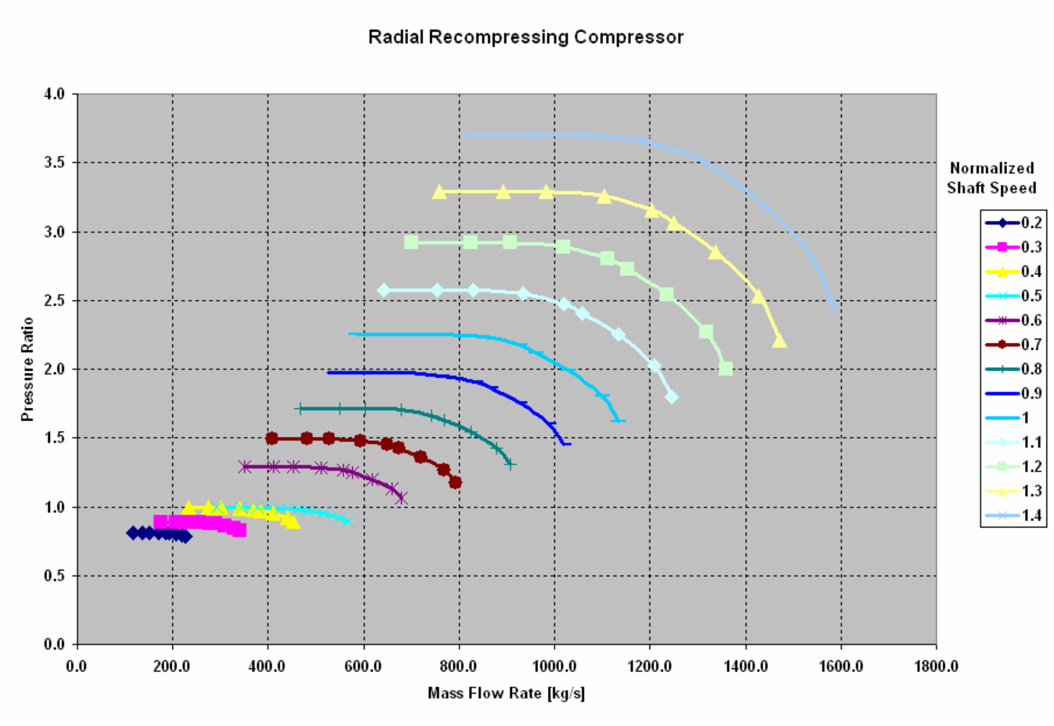


Figure 3-4: Radial Recompressing Compressor Pressure Ratio Map

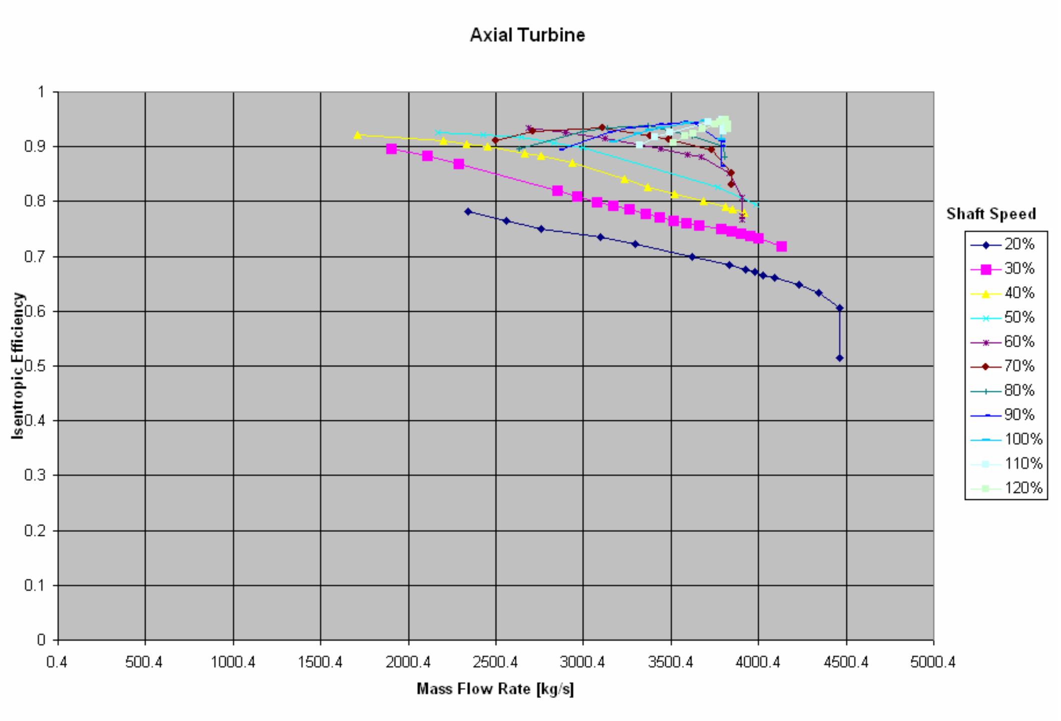


Figure 3-5: Axial Turbine Efficiency Map

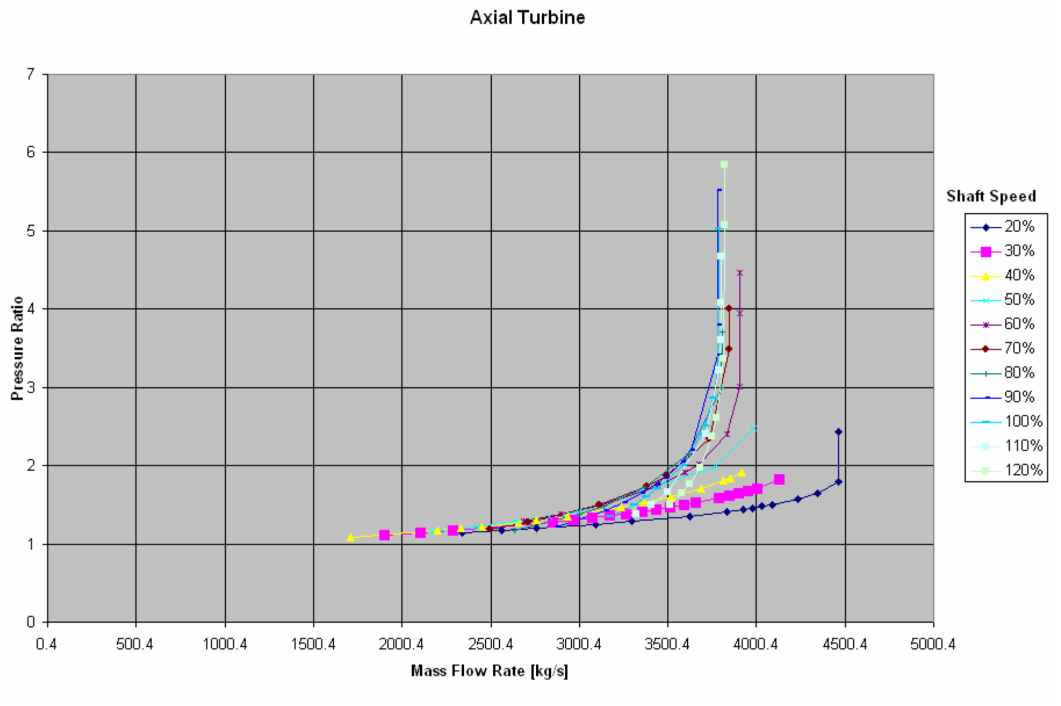


Figure 3-6: Axial Turbine Pressure Ratio Map

The axial data are based originally upon Jeff Wang's Report[13] and then modified by Dr. Yifang Gong. Due to the difficulty of S-CO_2's highly non-linear fluid properties near the critical point the main compressor shown here was designed at 42°C.

The axial turbine efficiency curves are shown in *Figure 3-5*. At high shaft speeds these curves show a linear increase in efficiency as mass flow rates increase until the peak efficiency is met then a rapid decrease. At low shaft speeds these curves show a linearly decreasing efficiency with increasing mass flow rates and a sudden final drop in efficiency. The shaft speed curves operate over a wider mass flow rate range at lower shaft speeds. It is worth noting that if the turbine is operated at the peak efficiency then small increases in mass flow rate will rapidly drop efficiency and move beyond the known data.

The axial turbine pressure ratio curves are shown in *Figure 3-6*. These curves show a gently sloping increase in pressure ratio that becomes increasingly steep as mass flow rate increases. At the end of the curve the increase in pressure ratio is nearly straight up. The mass flow rate operating rage increases as the shaft speed decreases.

The axial main compressor efficiency map is shown in *Figure 3-7*. At low shaft speeds these curves begin near their peak efficiency and rapidly decrease in efficiency as mass flow rates are increased. At high shaft speeds the curves become nearly flat with an efficiency peak towards the middle of the curve. The widest range of mass flow rates is found towards the middle shaft speeds.

The axial recompressing compressor efficiency map is shown in *Figure 3-8*. This map shows similar behavior to the main compressor at low shaft speeds. At high shaft speeds the curves show a lot more curvature than in the main compressor. The efficiency begins near the peak shaft speed, gently slopes to the peak as mass flow rate increases, and gently but increasingly drops off after the peak.

The axial main compressor pressure ratio curves are shown in *Figure 3-9*. At low mass flow rates these curves are nearly flat but linearly slope downwards with increasing mass flow rates. At high shaft speeds the curves become somewhat steeper but still linearly decrease as mass flow rate increase. The widest range of mass flow rates is found towards the middle shaft speeds.

The axial recompressing compressor pressure ratio curves are shown in *Figure 3-10*. These curves show similar behavior to the axial main compressor curves at low shaft speeds but have a much steeper slope at high shaft speeds. It is worth noting that the recompressing compressor has a relatively small mass flow rate operating range at high shafts speeds.

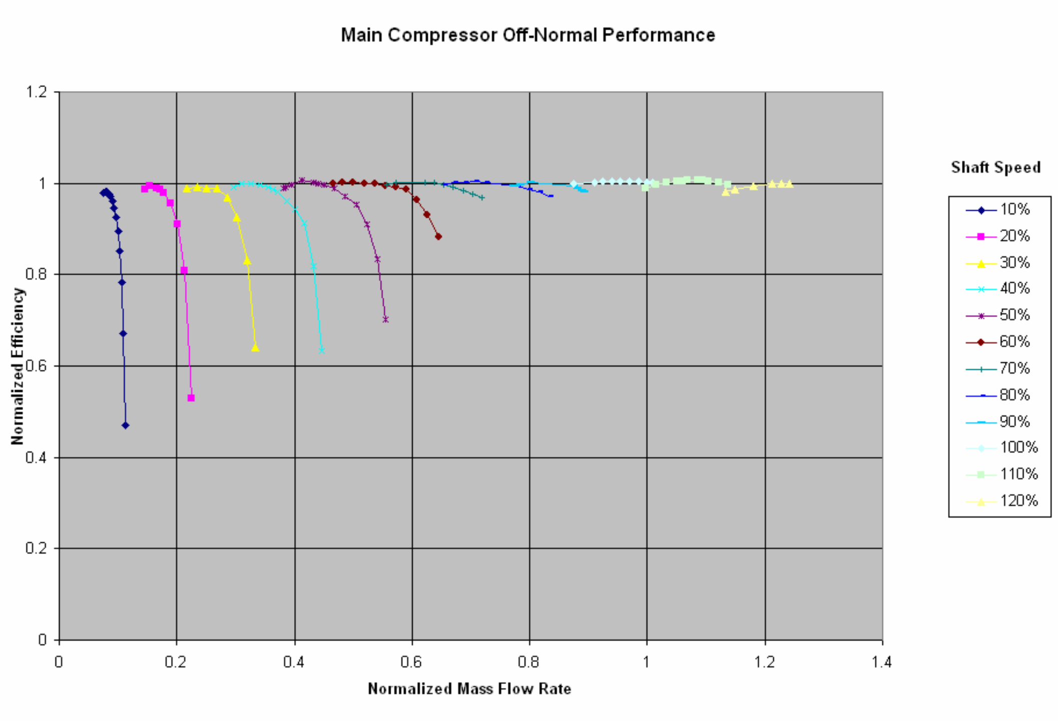


Figure 3-7: Axial Main Compressor Efficiency Map

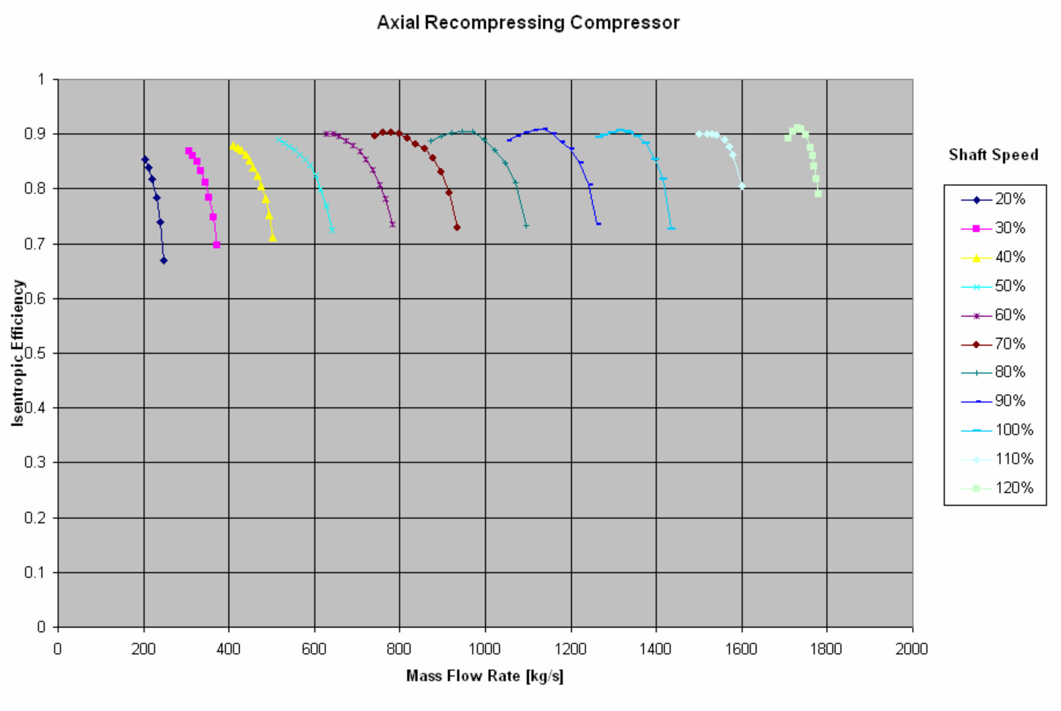


Figure 3-8: Axial Recompressing Compressor Efficiency Map

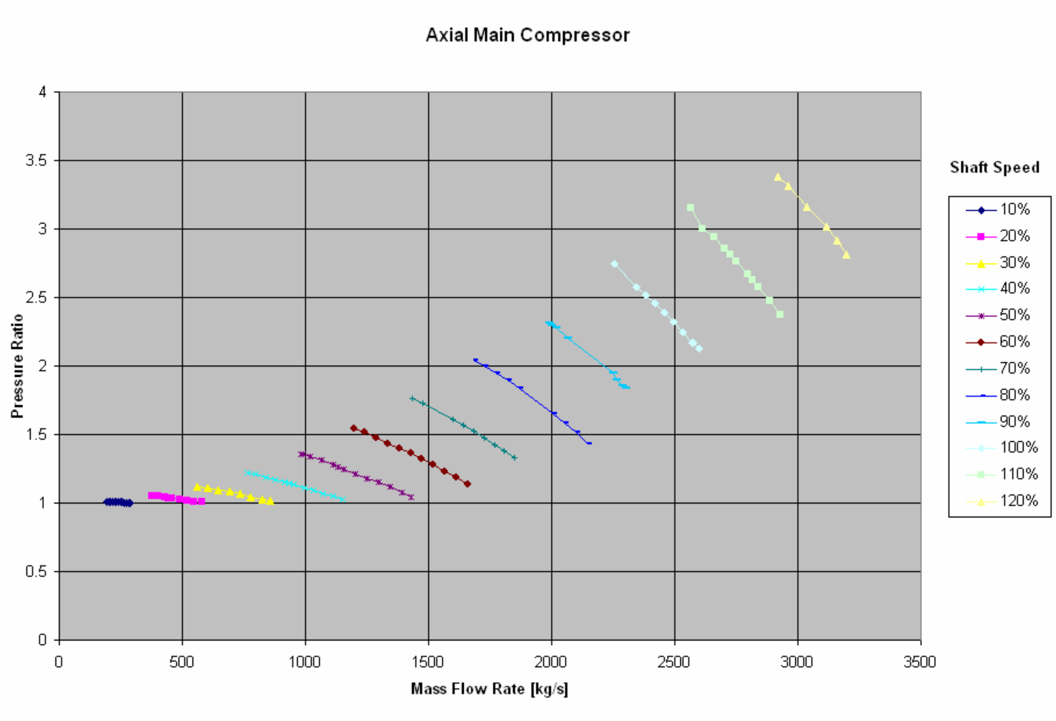


Figure 3-9: Axial Main Compressor Pressure Ratio Map

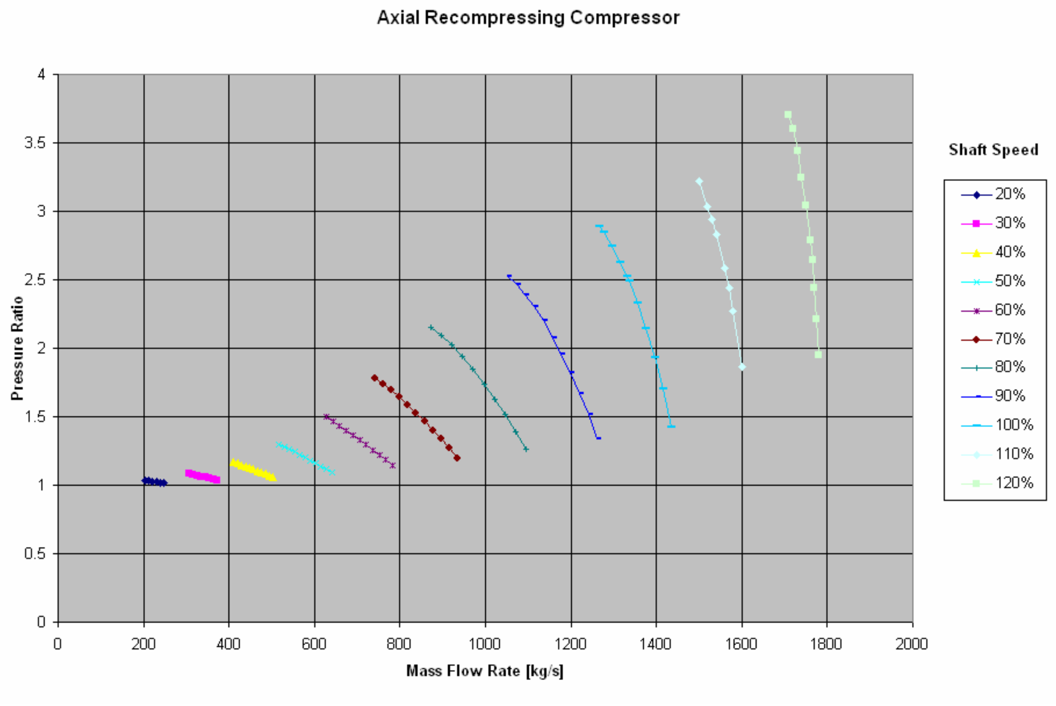


Figure 3-10: Axial Recompressing Compressor Pressure Ratio Map

3.2 Handling Performance Curves

Due to the complexity of S-CO_2 turbomachinery performance it is difficult if not impossible to normalize and collapse off-normal performance curves into a single functional relation as is sometimes done with helium (i.e. ideal) turbomachinery in the original GAS-PASS as shown in *Figure 3-11*.

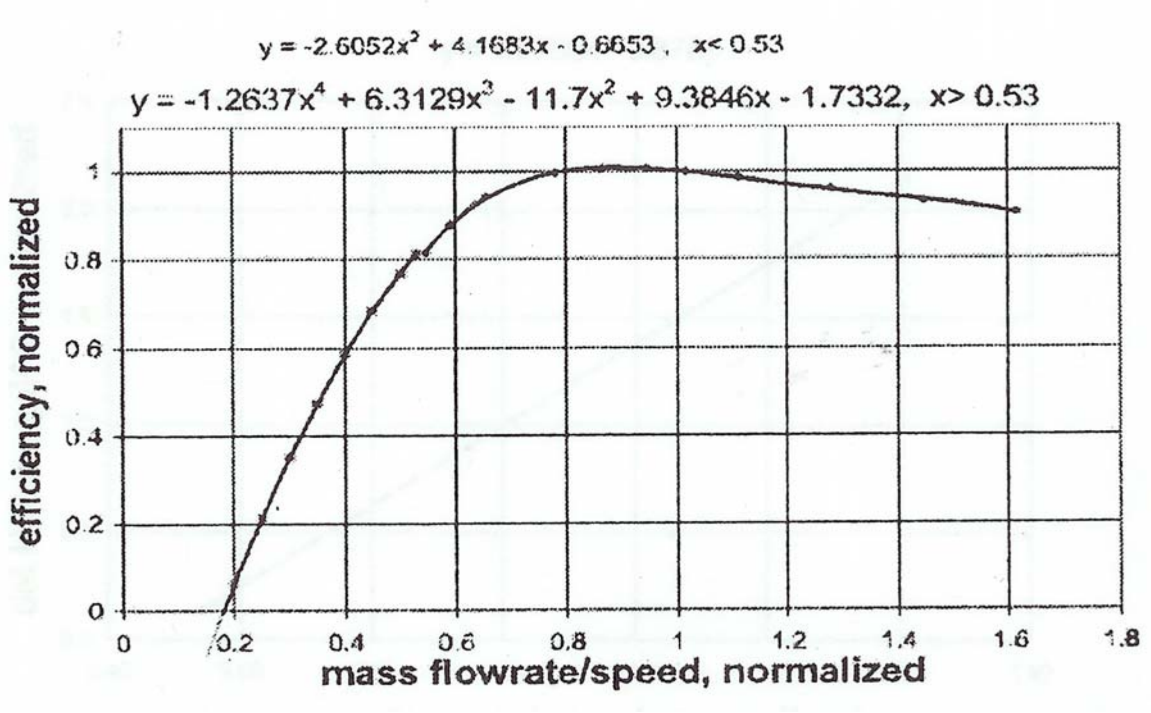


Figure 3-11: Original Gas-Pass Axial Turbine Efficiency Off-Design Performance Map[14]

This figure shows a rapid parabolic rise in efficiency, peaking near 1 then a gradual almost linear decrease following the peak. However, this turbomachine is expected to perform in this manner regardless of shaft speed.

This process can be made more accurate by fitting functions to shaft speed curves as shown in *Figure 3-12*. In this figure two performance curves at different shaft speeds are modeled with one equation. Due to the simple and similar nature of these curves little error is introduced.

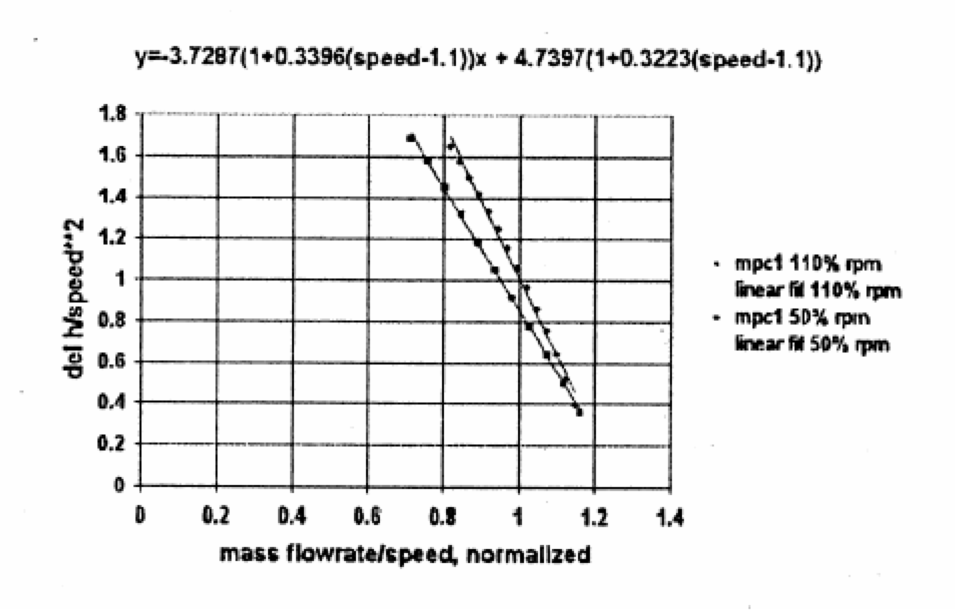


Figure 3-12: Original Gas-Pass Helium Axial Compressor Shaft Speed Lines[15]

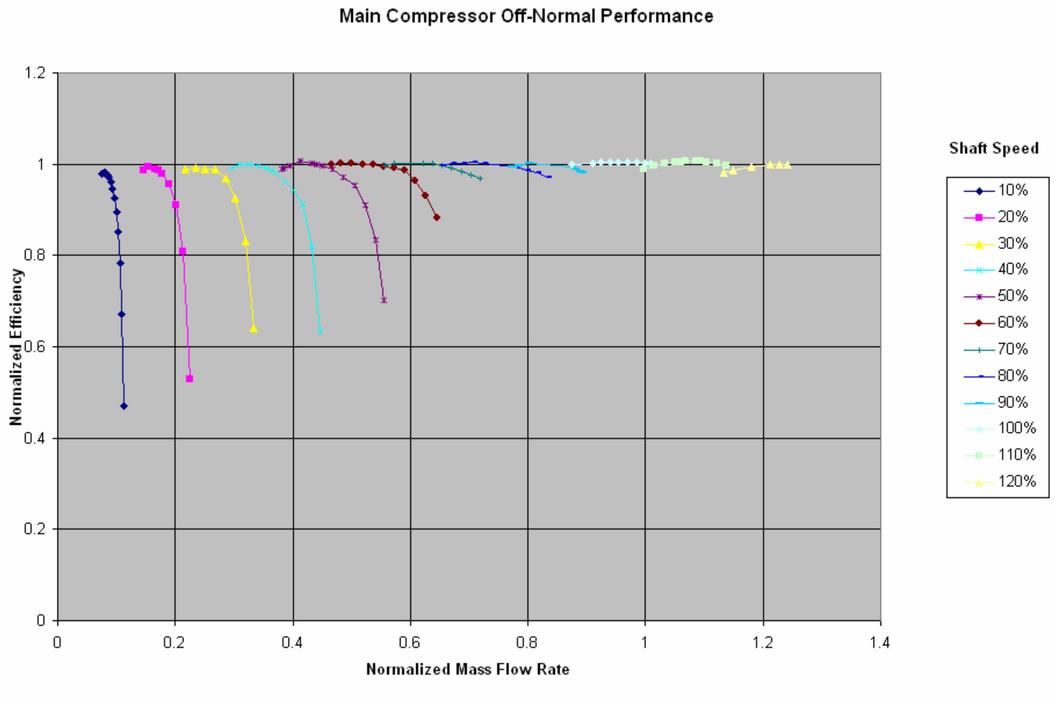


Figure 3-13: Axial Main Compressor Efficiency Curves

Unfortunately, the complex shapes of the performance curves for the S-CO_2 recompression cycle prevent such simple treatment as show in *Figure 3-13*. These curves were generated by Jeff Wang[16] and Dr. Yifang Gong for 42°C. While curves at nearby shaft speeds resemble one another it would be difficult and probably impossible to accurately model all of these shaft speed curves with a single simple equation. Therefore, GAS-PASS/CO_2 will avoid collapsing shaft speed curves in favor of interpolating between them. GAS-PASS/CO_2 will first interpolate on each surrounding shaft speed curve then interpolate between them.

3.2.1 Interpolating on a Shaft Speed Performance Curve

Interpolating on shaft speed curves is non-trivial due to the variety of curve shapes encountered in practice. A technique that works well on a radial compressor curve is unlikely to work well on an axial turbine curve without significant modification. GAS-PASS/CO_2 will use a simple but general approach that arose from numerous attempts at modeling shaft speed curves.

3.2.1.1 Notes on Correlations

In the work reported here we have followed the common practice of using normalized parameters to characterize turbomachinery performance over a wide range of operating parameters. It should be noted that there are several approaches employed in the literature, and many are suitable primarily for pedagogical purposes, being applicable with any degree of accuracy only to ideal gas and highly idealized turbomachines. The reader is referred to references Mathews[17], Vilim[18], Aungier[19], Japikse[20] for relevant discussions.

In the present work a more utilitarian approach is taken, to semi-empirically correlate off-normal performance with accuracy sufficient for the end-use application in mind.

The convention adopted for Gas-Pass employs two correlations:

1) normalized efficiency versus normalized mass flow rate divided by normalized shaft speed, for a series of constant rotational speeds
2) normalized pressure ratio versus normalized mass flow rate divided by normalized shaft speed, for a series of constant rotational speeds

The normalization of all variables is to values at full power and flow and normal rotational speed: thus both correlations pass through point (1,1).

3.2.1.2 Monotonic Trimming

The first step used to model shaft speed curves was trimming functions that are non-monotonic. At the highest pressure ratios, for a given shaft speed, the mass flow rate may decrease by a fraction of a percent despite the increase in the pressure ratio and the power. Unfortunately, having a non-monotonic function would add significant complexity and removing these points loses little real data.

In all observed cases this phenomena occurs when the efficiency is decreasing very rapidly and only at the end of the data set. A representative example is shown in *Figure 3-14*, where the last data point decreases on the **X** axis (probably not visually observable) by a fraction of a percent. Aside from the problems of non-monotonic functions this point creates a very steep slope that causes problems in many interpolation methods as well. Therefore any point at the end of a shaft speed data set that would create a non-monotonic curve is removed.

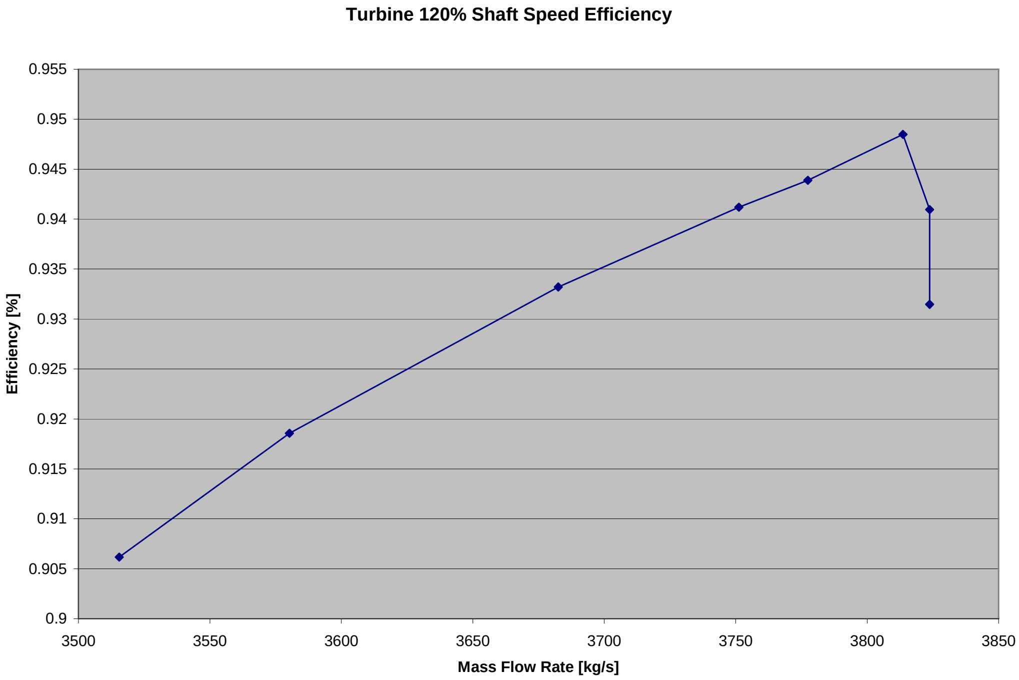


Figure 3-14: S-CO$_2$ Turbine 120% Shaft Speed Efficiency versus Mass Flow Rate

3.2.1.3 Linear Interpolation

Linear interpolation offers the simplest shaft speed modeling method and performs surprisingly well. *Figure 3-15* shows linear interpolation and cubic spine interpolation for the recompressing compressor at 120% of the normal shaft speed. While both curves match well, at most points, the linear curve shows deviation from the cubic spline near 1.105 on the **X** axis where there is a greater slope change between points. The cubic spline curve more faithfully mimics expected physics, while the linear undershoots significantly.

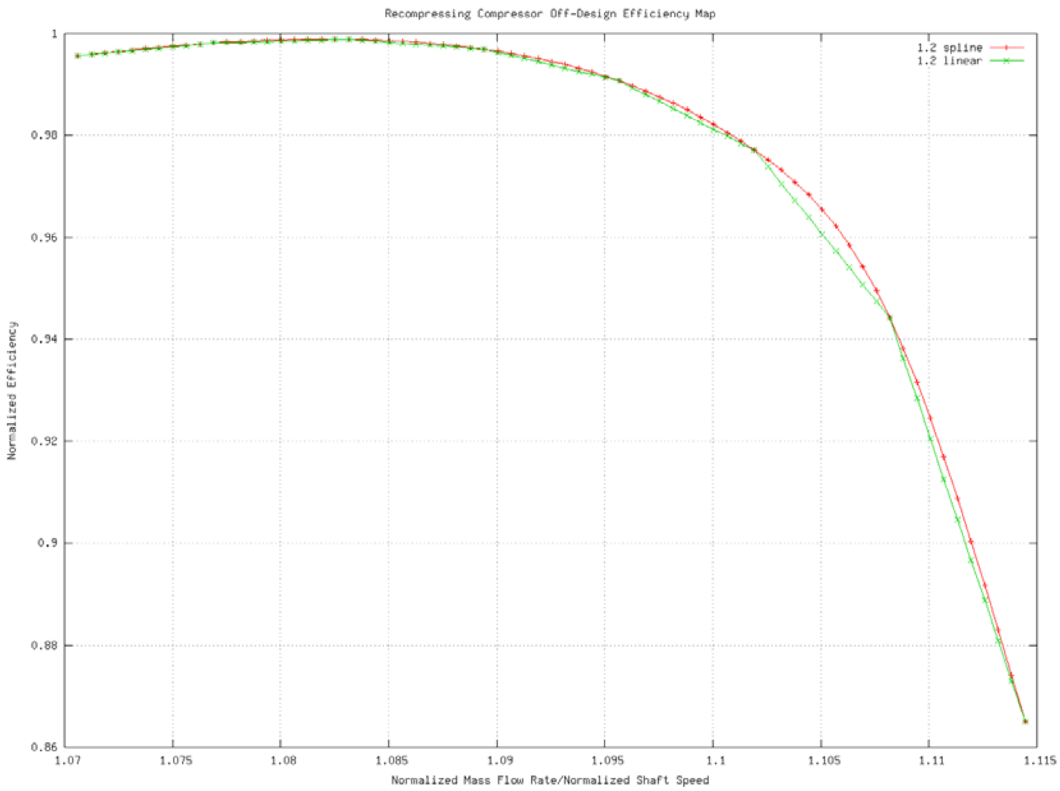


Figure 3-15: Linear and Cubic Spline Interpolation for the S-CO2 Axial Recompressing Compressor at 120% of Normal Shaft Speed

Linear interpolation may offer reasonable performance from a graphical point of view but it will always underestimate points near significant slope changes such as around a peak (a key area of interest). Therefore, it will be used as a backup method when more advanced methods fail.

Pure linear interpolation also has instant changes in slope due to discontinuous first derivatives at points, which has the potential to create convergence problems later, aside from being a non-physical phenomena. When linear interpolation is used this problem is avoided by smoothing the transition between lines with the first 10% of a line fractionally combined with the next line.

3.2.1.4 Cubic Spline Interpolation

More advanced interpolation methods may yield better results than linear interpolation. Two properties of the shaft speed curves stand out when picking an interpolation method:

-every data point is equally accurate (best estimate)
-all the normalized performance curves (generally) gently curve between each point

These two properties suggest cubic spline interpolation (fitting a polynomial via regression will be addressed later) which exactly models each point and adds curvature between points. Cubic is selected because higher order splines could add unwanted oscillation between points.

Cubic spline interpolation gives a formula which is "smooth in the first derivative, and continuous in the second derivative, both within an interval and at its boundaries."[21] The method creates a cubic function between each consecutive data point that passes exactly through the data point, matches the first derivatives, and has a continuous second derivative. In short, it makes a smooth curved line between each known data point.

Since all the data points are known, only two additional pieces of information are required to solve this system. In our case we may estimate the first derivative of the first and last point to solve for the cubic splines. A code for this purpose is available[22], very fast (linear, tridiagonal system, *O(N)* operations), and need be run only once. These equations can then be used to calculate the actual curve values by finding the appropriate position via bisection and solving the cubic equation. This code is also available[23], very fast, and is called each time a value on a shaft speed curve is desired. Using cubic spline interpolation one may calculate any position along a smooth performance curve with significant accuracy.

An example of cubic spline interpolation is shown in *Figure 3-15*. The reader may observe that all data points match exactly, there are no wild oscillations in the curve, and that the cubic splines closely match linear interpolation while adding slightly more curvature as desired.

However, cubic splines present two special challenges. The first challenge is to correctly estimate the first derivatives at the end points and the second is to handle rapid slope changes.

After some experimentation it was decided that the best general method for setting the end point derivatives is to use a natural spline. Natural splines set the **second derivative** equal to 0 at the end points which allows the curve to move freely at other points

3.2.1.5 Cubic Spline Problems with Rapid Slope Changes

In the axial turbine efficiency curves the cubic splines oscillate due to the extremely large slope changes. In some shaft speed curves the last few data **X** points change little while the last few **Y** points change greatly, producing slopes that cubic splines cannot handle appropriately. The problem is best described with *Figure 3-16*.

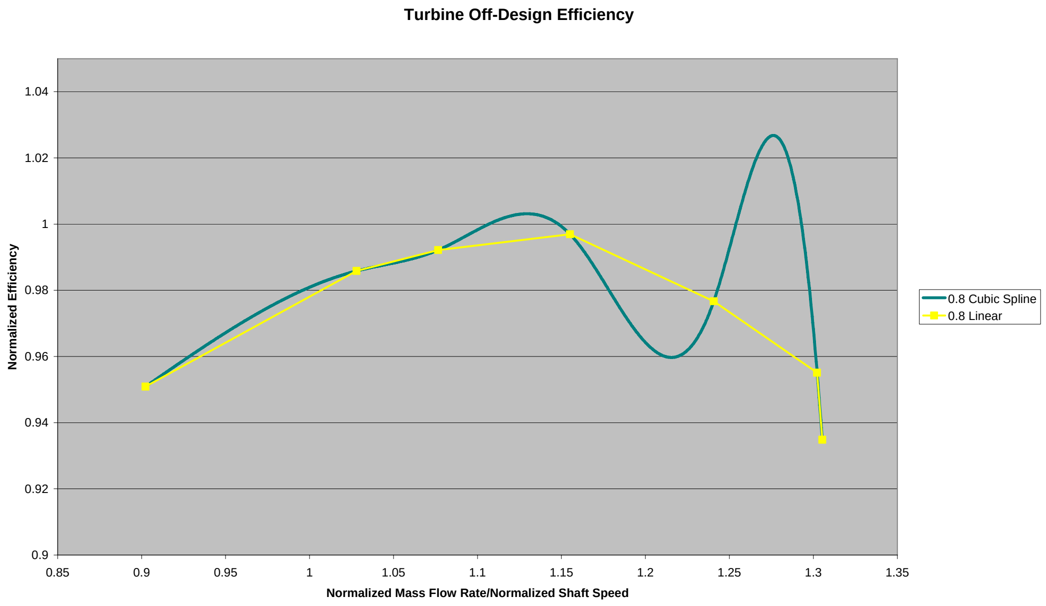


Figure 3-16: Cubic Spline Slope Oscillation

The reader will notice that the last two data points produce a relatively large slope. The last two data points are:

X	Y
1.302256	0.955096
1.305341	0.93484

These points create a linear slope of -6.56 compared to the median slope of the whole curve of -0.088. The large difference between these slopes between consecutive points creates problems when matching first derivatives and making continuous second derivatives with the cubic spline method.

Since this high slope region represents a small part of the **X** range (6% in *Figure 3-16*) it is probably sufficient to simply use linear interpolation here. One could argue that this region may be neglected without significantly affecting the turbine curve however; this may not be true when it comes to studying extreme transients. It is expected that turbomachinery efficiency will rapidly decrease at certain mass flow rates and shaft speeds, thus this could be a real physical phenomenon that may be needed in dynamic studies. On the other hand such deviations could be non-physical, and arise because the originally computed "data" is somehow flawed. Current turbomachinery design codes typically have great difficulty working with S-CO_2 near the fluid's critical point.

3.2.1.6 Shaft Speed Curve Modeling Summary

GAS-PASS/CO_2 models a shaft speed curve in a three step process:

1. Remove any (always final) data points that make the curve non-monotonic.

2. If any set of points creates a linear slope that is greater than 4x the median linear slope use linear interpolation here.
3. Create natural cubic splines between the remaining data points.

This relatively simple method will accurately capture the complex physics used in creating the shaft speed curves while still be general enough to apply to any axial or radial turbomachine tested thus far.

3.2.2 Interpolating Between Shaft Speed Performance Curves

Interpolating between turbomachinery shaft speed curves is difficult due to the complexity and diversity of the shape of the curves, the lack of available data, and the lack of theory to accurately tie together known curves. While good solutions exist for interpolating on a single shaft speed curve, it is difficult to accurately interpolate between shaft speed curves.

This difficultly is easily shown with an example. *Figure 3-17* shows the axial main compressor efficiency shaft speed curves at 10% and 20% of nominal shaft speed. If the shaft speed curves covered the same mass flow rates then one could simply interpolate for the appropriate efficiency (by mass flow rate) on each curve then by the shaft speed between the curves. Unfortunately, this is never the case and, as seen in *Figure 3-17*, it is possible that mass flow rates will not overlap, thus making it impossible to interpolate for the desired mass flow rate on at least one shaft speed curve and possibly on both.

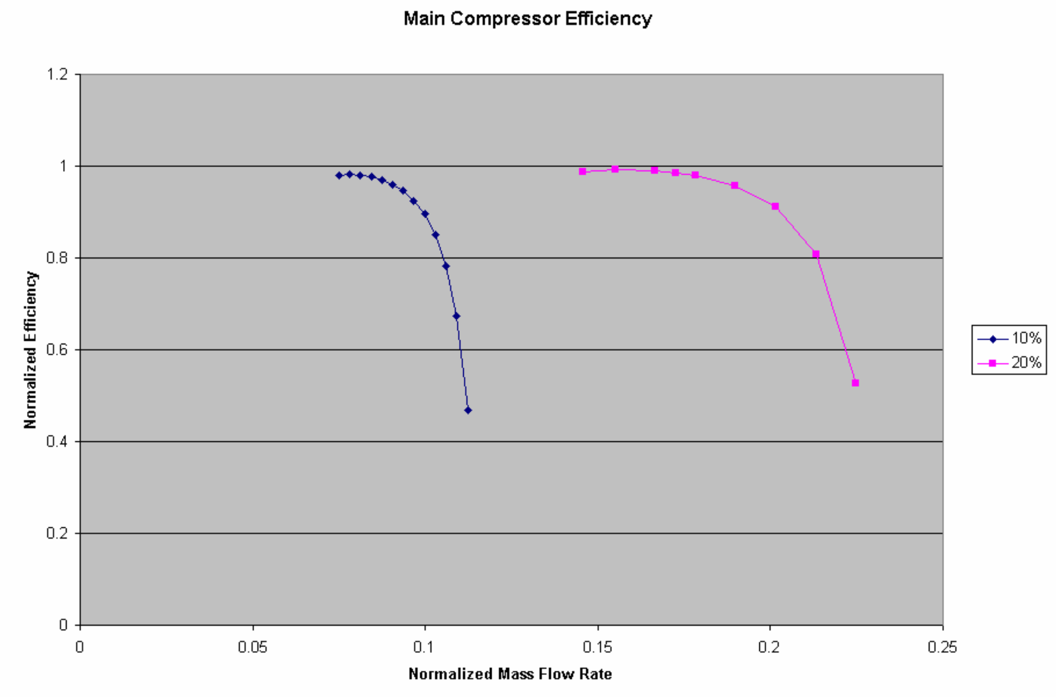


Figure 3-17: Shaft Speed Interpolation Example

A better approach would connect lines between points on the curves to estimate intermediate values. By connecting points between the curves one can estimate an intermediate shaft speed curve and use this new line to interpolate by mass flow rate.

Figure 3-18 shows this by connecting the first, middle, and last data points between both curves. Unfortunately, there is no easy method to determine which points on the curves to connect. Currently, the data points on each shaft speed curve need only converge within the modeling code and there must be enough of them to get an idea of the shape of that shaft speed curve. The curves are neither complete (i.e. extend from stall to surge) nor are there the same number of data points, nor are they in the same places on the curves. As seen in *Figure 3-18* the first and last connecting lines look correct but the middle data points' line has an inaccurate slope. Fundamentally, there is no physical reason to connect these points.

The only point that is physically defined on these curves is the maximum efficiency. This is the point where a reduction or an increase in mass flow rate would cause a decrease in compressor efficiency, which is a well defined physical process. By drawing a line between the maxima, as shown in *Figure 3-19*, we may then use this slope to map one curve to the other.

It is worth noting that some shaft speed curves may have many data points close to the maximum i.e. the curve is flat. This is not important in our case because if the curve is flat, and we are only using the maxima to estimate the slope between the curves, then the slope will be accurate regardless of the actual point chosen.

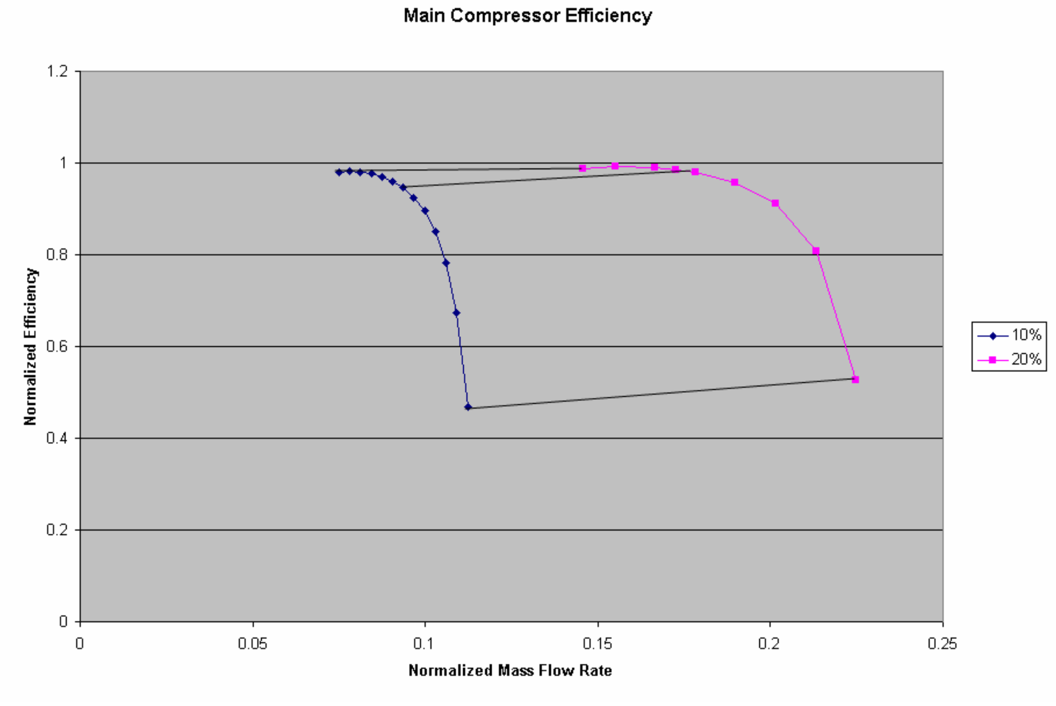


Figure 3-18: Connecting Points Example

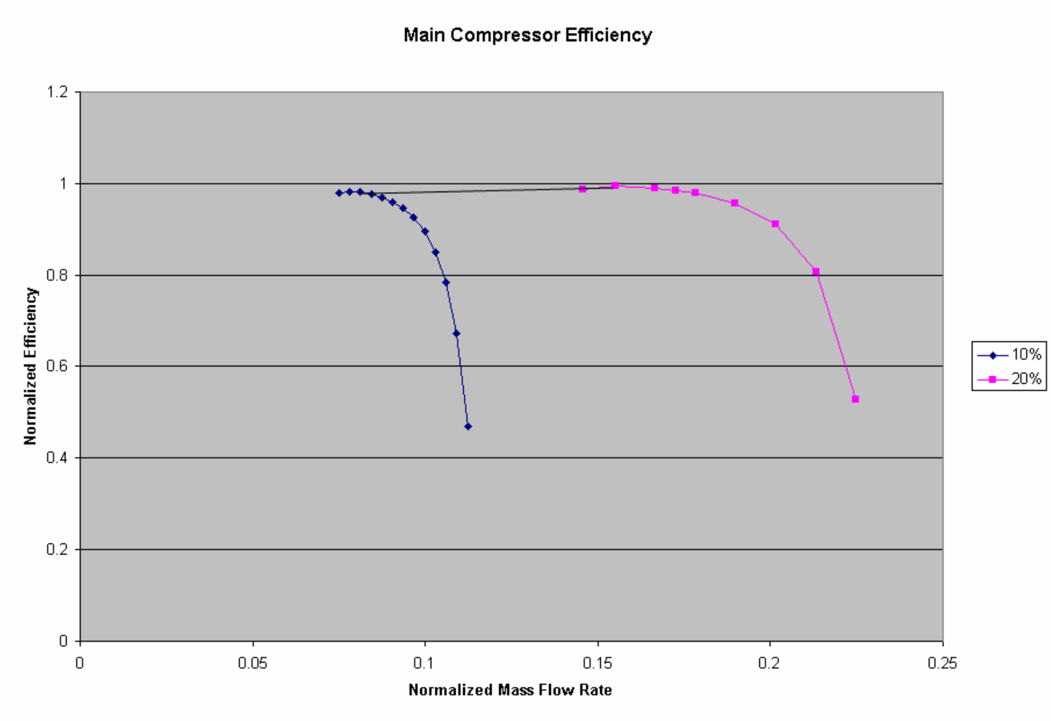


Figure 3-19: Maxima Line

If this connecting line's slope is used to map the transition between the shaft speed curves at several points one arrives at something like *Figure 3-20*. The connecting lines are parallel, start from the first shaft speed curve, and the length of each line is determined by the point at which it intersects the second shaft speed curve.

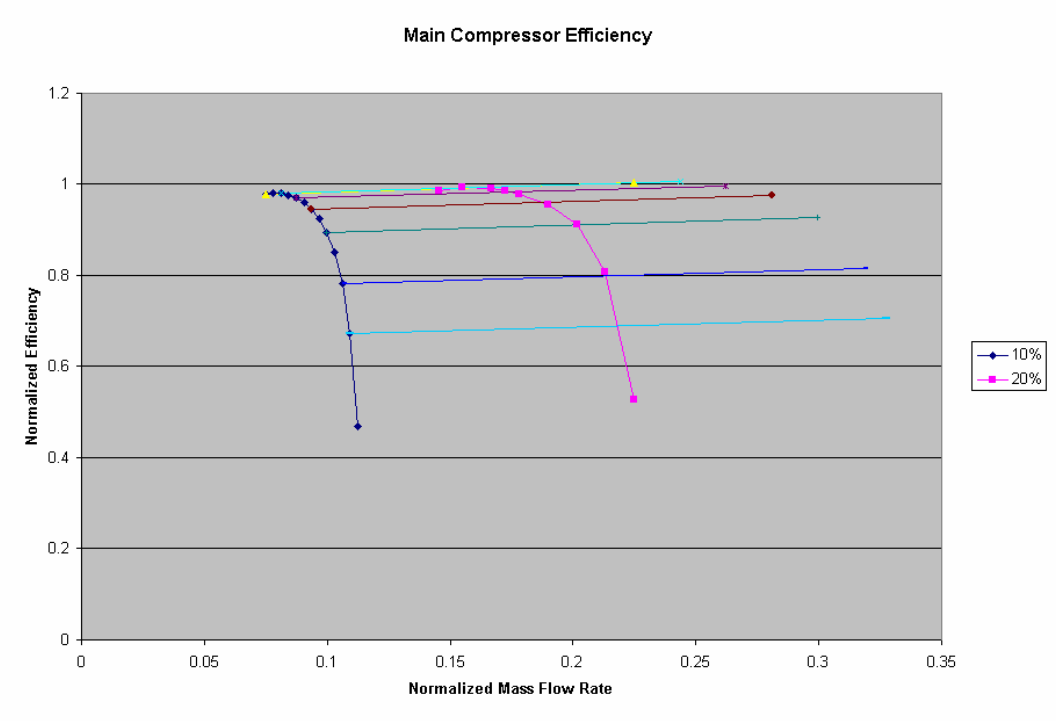


Figure 3-20: Parallel Maxima Lines

Unfortunately, this method does not work well due to the diversity of curves encountered, especially in the turbine. The difficulty begins when one examines the turbine efficiency curves shown in *Figure 3-21*. The efficiency curves approximate a parabola around the maximum. Any shaft speed interpolation mechanism must decide which side of the maximum to interpolate from, determine if the upper curve features the same parabolic behavior, and then create a connecting line between the appropriate sides of the maximum.

Even if this is successfully accomplished, the shaft speed curves shown in *Figure 3-22* present new problems. These curves have little mass flow rate separation and cross each other at least twice. To date no simple method has been found to handle multiple line crossings between shaft speeds while simultaneously handling parabolic behavior around the maximum in the compressors and turbine.

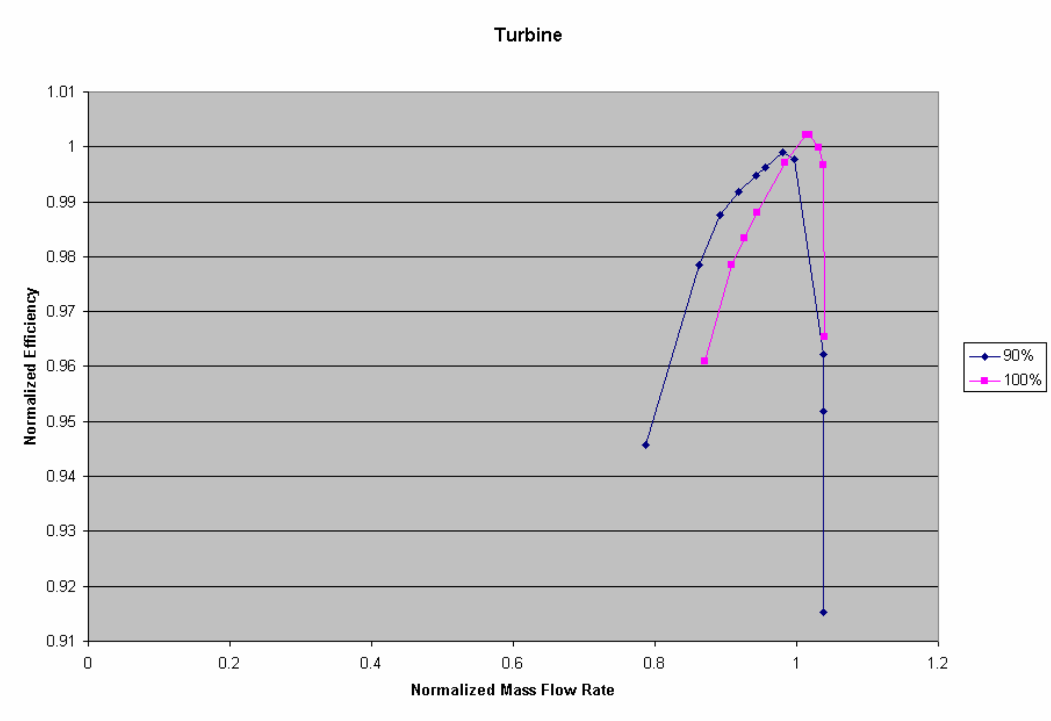


Figure 3-21: Turbine Curve Efficiency Difficulty

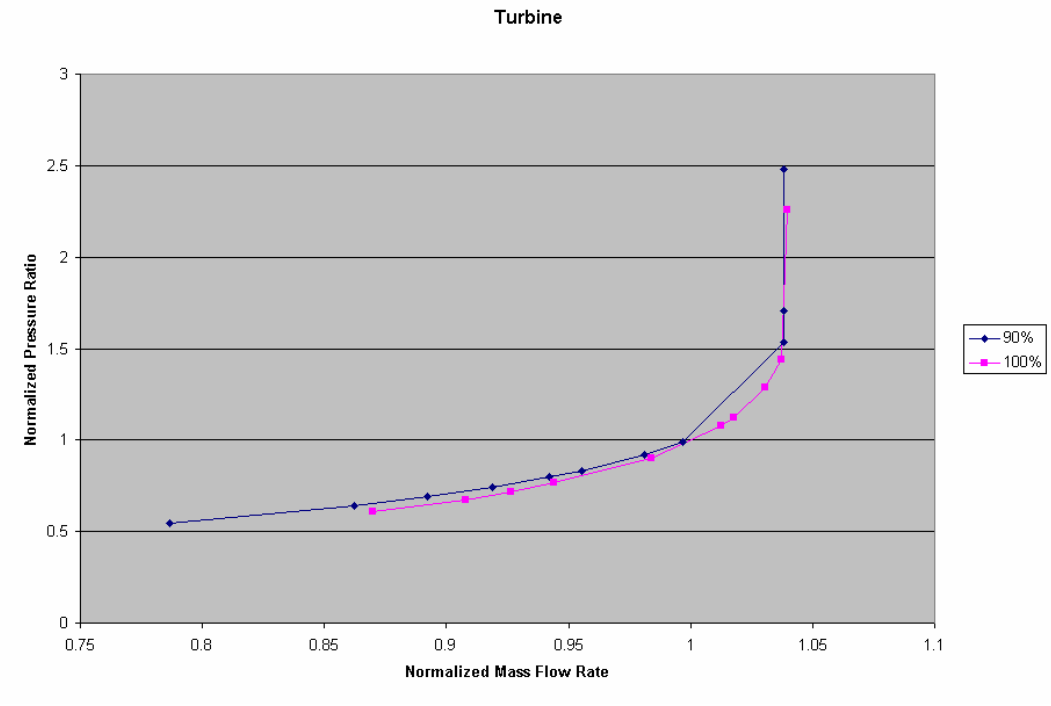


Figure 3-22: Turbine Curve Pressure Ratio Complexity

A simple solution that avoids the complexity of merging two shaft speed curves yet captures much of the information from both curves is based upon interpolating between shaft speed efficiency maxima. This approach is a several step process:

1. Find the mass flow rates where two shaft speed efficiency curves peak.
2. Determine the slope and distance between the curves on both the efficiency and pressure ratio curves at the previously determined mass flow rates.
3. Assume the shaft speed curves are parallel and use the lower shaft speed curve with the maxima connecting line to interpolate between them.

This avoids the many problems of merging the two curves at the cost of only using the maxima data to interpolate by shaft speed -- the upper shaft speed curve is completely ignored except for its maximum efficiency value (and pressure ratio at that point).

While crude, this method should allow realistic shaft speed interpolation until more accurate curve data are known. When accurate shaft speed curves are known it may be much easier and more worthwhile to capture data from the shaft speed curves equally. At this point, this method allows reasonably accuracy with few drawbacks.

3.2.2.1 Interpolating Between Shaft Speed Curve Summary

Interpolating between shaft speeds is difficult due to the lack of available data and the variety of curve shapes found in the S-CO_2 compressors and turbine. A simple and reasonably accurate solution uses the slope and distance between the maximum efficiencies (or pressure ratios) on bounding shaft speed curves to interpolate between them. The first curve is accurately modeled and then the connecting line is used to move between the curves. For example, in *Figure 3-19* take the maximum to maximum line, slide it down keeping the slop and length the same, then interpolate along its length by shaft speed. This method looses much of the upper shaft speed curve shape but avoids many potential pitfalls (including placing too much emphasis on initial curve estimates) and is very fast. It will be adopted for all shaft speed interpolation for the foreseeable future.

3.3 Turbomachine Fluid Property Variation Effect

The performance of a turbomachine depends not only on the fluid mass flow rate and rotational speed but upon the incoming fluid properties. As fluid temperatures and pressures vary so will turbomachine performance. This section will briefly outline this problem and how Gas-Pass models fluid properties changes.

For a turbomachine the important varying fluid properties are viscosity and density. As viscosity changes the flow profile will change and frictional losses will change, both of which affect turbomachine performance. As density changes the turbomachine's ease of compressing or expanding the fluid will change therefore the pressure rise will change. To compute these effects is non-trivial and in general must be done with detailed finite element codes in a computationally expensive manner.

Gas-Pass allows the accurate and rapid calculation of these effects by allowing interpolation between multiple sets of performance curves. Each set of performance curves will allow interpolation by mass flow rate and shaft speed. Between performance curve sets one could interpolate by the changing fluid properties such as temperature and pressure.

Unfortunately, these sets of performance curves are not currently available. As turbomachinery modeling codes evolve to incorporate S-CO_2 and as experimental data are generated these performance curve sets will be incorporated into Gas-Pass. Until then fluid property effects will be approximated by taking into account the density difference encountered in the compressors.

3.3.1 Compressor Density Equation

This section will derive an equation which allows simple approximation of the a fluid property effect for compressors. The first law of thermodynamics for a control volume where shaft and flow work are separated is:

$$dU_{cv} = dH_{in} - dH_{out} + \delta Q - \delta W_{shaft}$$

For a steady state system which is adiabatic to the environment and where work is input:

$$H_{out} - H_{in} = W_{shaft}$$

If we separate the enthalpy into internal energy (neglecting kinetic energy changes and gravitational potential both of which are small for our compressors) and flow work:

$$U_{out} + P_{out}V_{out} - U_{in} - P_{in}V_{in} = W_{shaft}$$

If we assume that the fluid density is constant (which is almost the case for the main compressor as discussed shortly) then:

$$P_{out} - P_{in} = \frac{W_{shaft} - \left(U_{out} - U_{in}\right)}{V}$$

This equation states that a compressor's pressure rise is proportional to the shaft work minus the energy lost to increasing fluid temperature (viscous forces) which is conceptually what we expect. Dividing both sides by the fluid density times gravity:

$$\frac{P_{out} - P_{in}}{\rho g} = \frac{W_{shaft} - \left(U_{out} - U_{in}\right)}{V\rho g}$$

The right hand side is also known as actual pump head (*h* now refers to head not enthalpy – they will not appear in the same equation):

$$\frac{P_{out} - P_{in}}{\rho g} = h_{actual}$$

This can be restated:

$$\Delta P = h_{actual}\rho g$$

This equation expresses the pressure rise as proportional to density where the only assumption is an incompressible fluid and steady state flow. If we further assume that the actual head term does not change with changing properties then we can say:

$$\Delta P_1 = \alpha\rho_1$$

$$\Delta P_2 = \alpha\rho_2$$

Therefore we can compute a pressure rise at a new fluid density by:

$$\Delta P_2 = \Delta P_1 \frac{\rho_2}{\rho_1}$$

This equation thus assumes:

- Inviscid flow
- Incompressible flow
- Steady flow.

For fluid flow, inviscid forces become important when they become significant compared to inertial forces – this is the definition of the Reynolds number. Therefore, one may generally approximate a fluid with a high Reynolds number i.e. turbulent flow as inviscid (over short flow distances). The S-CO_2 recompression cycle turbomachinery experience highly turbulent flow therefore this analysis will assume inviscid flow. Furthermore, we are only neglecting the difference in viscous effects between the design and new fluid properties. Viscous effects are properly accounted for at design fluid conditions.

The assumption of incompressible flow is an approximation. However, when the S-CO_2 leaves the compressors it has largely stopped increasing in density and is relatively incompressible especially when compared to an ideal gas system. Furthermore, even within the compressors the fluid is relatively incompressible. *Table 3-1* compares the density ratios of an incompressible fluid and an ideal gas to S-CO_2 for the main compressor inlet and outlet design conditions. The relative compressibility of the fluids suggests that the S-CO_2 behaves roughly like an incompressible fluid.

Table 3-1: Density Ratios at Main Compressor Operating Conditions

Fluid:	**ρ_2/ρ_1**
Incompressible	1
Ideal Gas	2.38
S-CO_2	1.24

Finally, the steady state assumption will be applied solely to the fluid property corrections in the turbomachine performance curves. Any transient effects in the turbomachines will occur on a much shorter timescale than those occurring within the cycle due to these machine's relatively small volumes and masses. The whole plant will continue to account for transient energy and mass storage.

3.3.2 Implementing Property Correction

The turbine behaves more like an ideal gas than an incompressible fluid thus its property fluid corrections will be neglected until accurate sets of property performance curves become available.

Compressor pressure rise may be computed from a single set of shaft speed curves (at the design point fluid properties) then corrected by dividing the non-design fluid density by the design fluid density. This simple factor allows Gas-Pass to easily and rapidly account for much of the effects of fluid property variation in the compressors until performance curves become available at different fluid properties.

To summarize, the process for computing compressor pressure rise is:

1. Using the GAS-PASS/CO_2 supplied inlet mass flow rate and shaft speed interpolate for pressure ratio using the shaft speed curves.
2. Using the GAS-PASS/CO_2 compressor outlet conditions (enthalpy and pressure) compute the outlet density.
3. Adjust the compressor pressure ratio by the ratio of the calculated outlet density to the compressor's steady state outlet density.

The GAS-PASS/CO_2 solution process will be satisfied only when all conservation equations are simultaneously correct. Therefore the inlet and outlet values supplied to the compressor module described here will be accurate.

4 Printed Circuit Heat Exchangers

Modeling printed circuit heat exchanger performance (PCHE) is a key area of research in the super-critical carbon dioxide (S-CO_2) recompression cycle. Due to the large variation in carbon dioxide properties (especially near the critical point in the precooler) it is not trivial to accurately calculate PCHE performance.

This chapter offers steady state and transient equations and methods for assessing straight channel PCHE performance in the S-CO_2 recompression cycle. Both methods rely on one-dimensional, nodalized, iterative computational solutions. Complete fluid dynamic simulations are computationally expensive (runtimes of hours to weeks). Fortunately, for most applications this level of analysis is not necessary. Using the proposed methods one may solve for heat transfer and other properties of a PCHEs at a time step with runtime on the order of a millisecond. This tool allows rapid yet accurate assessment of internal PCHE performance that will be used in the overall Gas-Pass code.

This chapter is subdivided into seven sections:

1. Section one provides a brief background on PCHEs.
2. Section two provides an overview of the heat transfer and pressure drop correlations.
3. Section three describes how a PCHE is nodalized and computed efficiently.
4. Section four provides a graphical overview of the steady state performance of the various PCHEs.
5. Section five analysis the precooler thermal inertia.
6. Section six derives the transient equation implicitly applied to the PCHEs in Gas-Pass
7. Section seven summarizes this chapter.

Unfortunately, dynamic simulation (for applications like control studies) may require millions of simulations and a much faster means to simulate PCHE heat transfer is desirable. While analytic equations may be derived, their solution generally requires numerous calculations. By simplifying the previously used 1D heat transfer code and using tabular fluid properties, this code will calculate the total heat transferred with a runtime on the order of a couple of milliseconds and a relatively high degree of accuracy.

4.1 Background

The S-CO$_2$ recompression cycle's advantageous performance is primarily accomplished by taking advantage of CO$_2$ property changes as the fluid nears its critical point (at the bottom of the cycle) and by using high efficiency yet compact heat exchangers to significantly recuperate the cycle. The cycle layout is shown in *Figure 4-1: S-CO2 Recompression cycle* which shows the four heat exchangers which must be modeled in Gas-Pass.

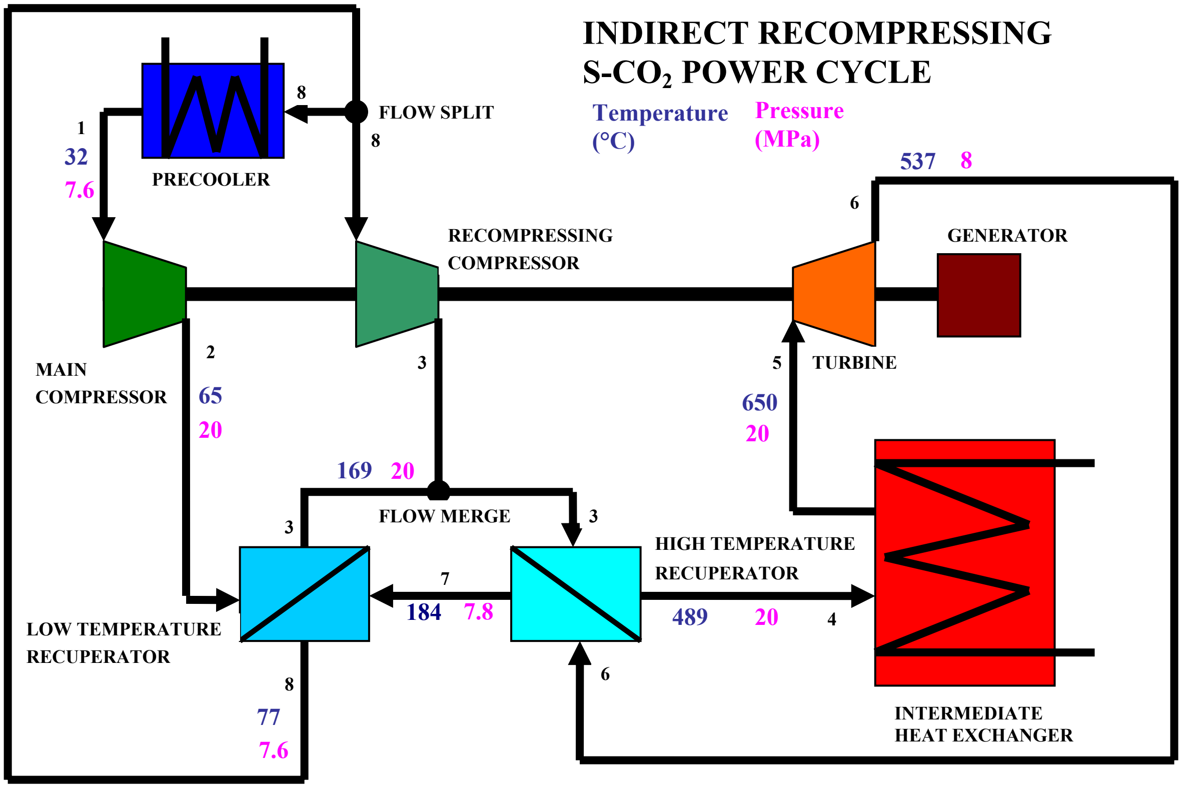


Figure 4-1: S-CO$_2$ Recompression cycle

Unfortunately, while large CO$_2$ property variations allow high efficiency they also complicate standard analysis assumptions. This is especially true in the heat exchangers where standard assumptions like log-mean temperature heat transfer are not valid.

New cycle designs[24] have taken advantage of new heat exchanger technology in the form of printed circuit heat exchangers (PCHEs). These heat exchangers have been described elsewhere[25] but the basic idea of a channel layout is shown in *Figure 4-3: PCHE Channel Layout*. Note that in an actual PCHE there are many thousands of channels.

4.2 1-D Straight Channel Correlations

The appropriate correlation to use with S-CO_2 heat exchange is an area of active research at this time. Due to the variety of designs and the lack of published data (see Ishizuka et. al.[26] for initial wavy channel experimental results and correlation factors) only semi-circular (current available PCHE designs use semi-circular channels) straight channels will be considered in this chapter.

4.21 Applicable Literature

There is a long history of property correlations, but according to Wang and Hihara[27] many turbulent correlations stem from Petukhov et. al.[28] (1974). The Petukhov correlation was improved upon by Gnielinski[29] (1976) into what has become a widely applied correlation. According to Wang and Hihara[30] (2002) Gnielinski's constant property correlation has been cited by numerous authorities as the most accurate for constant property turbulent flow in a tube but was found inaccurate for variable property flows by Olsen[31] 2000.

Wang and Hihara computationally compared several correlations for S-CO_2 PCHE heat transfer and found that that the Gnielinski constant property correlation significantly underestimates the heat transfer compared to other correlations. Wang and Hihara conclude that the more recently developed methods use variable property models and predict higher heat transfer coefficients.

Oslen also compared several correlations that attempt to correct for supercritical property variation. Unfortunately, he concluded that no correlations "were found to adequately predict the measurements over the entire range of experimental parameters." The Gnielinski constant property correlation was found to consistently under-predict heat transfer. It should be noted that Olsen used the correlations to predict integral heat transfer (i.e. only based upon inlet and outlet conditions) and did not attempt to apply these correlations in a nodalized fashion. However, it is likely that even if the Gnielinski constant property correlation was analyzed in a nodalized fashion it would under-predict the true heat transfer.

Unfortunately, the author has found no analysis of Gnielinski's recommended correlations which include temperature correction. Gnielinski's commonly cited correlation is shown in *Equation 4-1*.

Equation 4-1: Gnielinski Constant Property Turbulent Heat Transfer Correlation

$$Nu = \frac{\xi/8\,(\mathrm{Re}-1000)\mathrm{Pr}}{1+12.7\sqrt{\xi/8}\left(\mathrm{Pr}^{2/3}-1\right)}$$

However, Gnielinski's recommended and concluding correlations include property temperature correction factors for liquids and gases and are shown in ***Error! Reference***

source not found. and ***Error! Reference source not found.***. These correlations are numerically identical to ***Error! Reference source not found.*** for smooth ducts with the addition of developing length correction factors and property temperature dependence factors.

Equation 4-2: Gnielinski Turbulent Gas Heat Transfer Correlation: 0.6<Pr<1.5

$$Nu = 0.0214\left(\mathrm{Re}^{0.8} - 100\right)\mathrm{Pr}^{0.4}\left[1 + \left(\frac{d}{L}\right)^{2/3}\right]\left(\frac{T}{T_w}\right)^{0.45}$$

Equation 4-3: Gnielinski Turbulent Liquid Heat Transfer Correlation: 1.5<Pr<500

$$Nu = 0.012\left(\mathrm{Re}^{0.87} - 280\right)\mathrm{Pr}^{0.4}\left[1 + \left(\frac{d}{L}\right)^{2/3}\right]\left(\frac{\mathrm{Pr}}{\mathrm{Pr}_w}\right)^{0.11}$$

It is worth noting that more recent correlations use a “variable property model” which is identical to a temperature dependence factor in practice. While S-CO_2 properties vary greatly with both temperature and pressure near the critical point, in a PCHE the pressure difference will be miniscule between the center of the flow and the wall of a PCHE but the temperature difference can be more significant.

For laminar flow Hesselgreaves (2001)[32] recommends ***Error! Reference source not found.*** for our conditions.

Equation 4-4: Laminar Heat Transfer Correlation

$$Nu = 4.089$$

Hesselgreaves further recommends a set of tabular data from Shah and Batti (1987) for the laminar thermal entrance length effect which will be linearly interpolated in practice. This data is plotted in *Figure 4-2.*

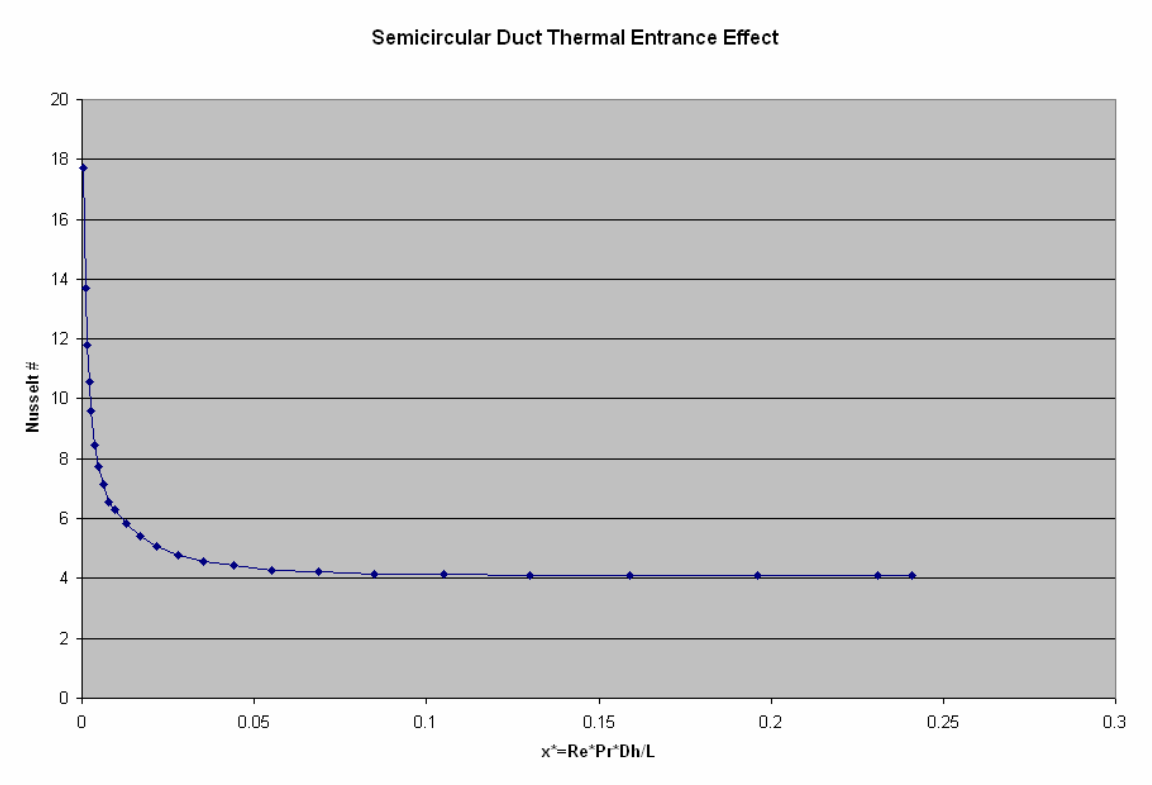


Figure 4-2: Semi-circular Tube Thermal Entrance Length

4.22 Straight Channel Heat Transfer Correlation

The method presented here is modified from code originally created by Dr. Pavel Hejzlar at MIT and is based upon Williams[33]. It is presented here in an abbreviated form with basic computer logic and loop constructs for brevity and clarity.

In summary, there are no known published experimental data for S-CO_2 heat transfer in a straight channel PCHE in support of correlations accurate for this purpose. Therefore the Gnielinski correlation shown in ***Error! Reference source not found.*** with entrance length correction will be used for turbulent flow in water and CO_2 since it is simple, experimentally based, widely used, and a recommended correlation that includes entrance length effects. The wall temperature effect was dropped due to its observed small effect and additional complication during transient analysis.

For laminar fluid flow in CO_2 and water the Hesselgreaves recommendations will be followed including interpolating tabular data for the entrance length effects.

Finally, a transition region will be used to create a smooth change from laminar to turbulent correlations. Since the turbulent correlation begins at 2300 a transition region was chosen to range from 500 above and below the transition point i.e. 1800 < Re < 2800. This region will simply use linear interpolation to create a smooth transition between the laminar and turbulent Nusselt numbers to calculate the heat transfer coefficient.

To summarize findings, for CO_2 or water PCHE heat transfer coefficients in pseudo-code:

```
If (Re < 1800) Then
    Xmod = Re*Pr*Dh/L
    If (Xmod < 0.159) Then
        Nu = (Linearly Interpolate Thermal entrance data)
    Else
        Nu = 4.089
    Endif
ElseIf (Re < 2800) Then
    Xmod = Re*Pr*Dh/L
    If (Xmod < 0.159) Then
        NuLam = (Linearly Interpolate Thermal entrance data)
    Else
        NuLam = 4.089
    Endif

    $f=(1.8*LOG10(2800)-1.5)^{-2} / 4$
    $NuTurb = f/2*((2800-1000) * Pr / ( 1 + 12.7 * (f/2)^{0.5} * (Pr^{2/3}-1)$
    $NuTurb=NuTurb*(1+(d/L)^{2/3}$

    Nu = (Re – 1800)/(2800-1800) * (NuTurb – NuLam) + NuLam
Else
    $f=(1.8*LOG10(2800)-1.5)^{-2} / 4$
    $NuTurb = f/2*((2800-1000) * Pr / ( 1 + 12.7 * (f/2)^{0.5} * (Pr^{2/3}-1)$
    $NuTurb=NuTurb*(1+(d/L)^{2/3}$
Endif
```

4.23 Straight Channel Pressure Drop

Again the method presented here is modified from code originally created by Dr. Pavel Hejzlar at MIT and is based upon Williams[34]. It is presented here in an abbreviated form with basic computer logic and loop constructs for brevity and clarity.

The heat transfer coefficient will also be calculated as a function of Reynolds Number and tube smoothness. The Reynolds Number flow transition points depend upon tube smoothness. The relative smoothness is the ratio of deviation peak size versus tube diameter:

$$\delta_{rel} = \frac{\delta}{D}$$

- The Reynolds Number transition points are:

```
If $\delta_{rel} < 0.007$ Then
        $\mathrm{Re}_0 = 2000$
        $\mathrm{Re}_1 = 2000$
ElseIf $\delta_{rel} \ge 0.007$ Then
        $\mathrm{Re}_0 = 754e^{0.0065/\delta}$
        $\mathrm{Re}_1 = 1160*(1/\delta)^{0.11}$
EndIf
$\mathrm{Re}_2 = 2090*(1/\delta)^{0.0635}$
$\mathrm{Re}_3 = 441.19*\delta^{-1.1772}$
```

- If Re < Re_0 then flow is before the first transition region it will be considered laminar:

$$f = \frac{64}{\mathrm{Re}}$$

- If Re > Re_0 and Re < Re_1 then the flow is within the first transition region (reducing with Re):

```
If $\delta_{rel} < 0.007$ Then
        $f = 0.032 + 3.895*10^{-7}*(\mathrm{Re} - 2000)$
Else
        $f = 4.4*\mathrm{Re}^{-0.595}*e^{-0.00275/\delta}$
EndIf
```

- If Re > Re_1 and Re < Re_2 then the flow is within the second transition region (increasing with Re):

```
$f = 0.11*\left(\delta_{rel} + 68/\mathrm{Re}_2\right)^{0.25}$
Do
        $f_n = \left(\frac{1}{2*\log_{10}\left(\frac{2.51}{\mathrm{Re}_2\sqrt{f}} + \frac{\delta_{rel}}{3.7}\right)}\right)^2$
    If $ABS\left(\frac{f_n - f}{f_n}\right) < 0.01$ Then
            $f = f_n$
            Exit Loop
        EndIf
        $f = f_n$
    EndDo Loop

    If ( $\delta_{rel} < 0.007$ ) Then
        $f_1 = 0.032$
        $f_2 = f$
    Else
        $f_1 = 0.0758 - \frac{0.0109}{\delta}$
        $f_2 = f$
    EndIf
    $f = (f_2 - f_1) * e^{-(0.0017*(\mathrm{Re}_2 - \mathrm{Re}))^2} + f_1$
```

- If Re > Re_2 and Re < Re_3 then the flow is within the third region (decreasing with Re):

```
    $f = 0.11 * \left( \delta + \frac{68}{\mathrm{Re}} \right)^{0.25}$
    Loop2: Do
        $f_n = \left( \frac{1}{2 * \log_{10}\left( \frac{2.51}{\mathrm{Re}\sqrt{f}} + \frac{\delta_{rel}}{3.7} \right)} \right)^2$
        If ABS$\left( \frac{f_n - f}{f_n} \right) < 0.01$ Then
            $f = f_n$
            Exit Loop2
        EndIf
        $f = f_n$
    EndDo Loop2
```

- If Re > Re_3 then the flow is highly developed (independent of Reynolds Number):

```
    $f = 0.11 * \left( \delta_{rel} + {}^{68}\!/\!_{\mathrm{Re}_3} \right)^{0.25}$
    Loop3: Do
```

```
        $f_n = \left( \frac{1}{2*\log_{10}\left( \frac{2.51}{\mathrm{Re}_3 \sqrt{f}} + \frac{\delta_{rel}}{3.7} \right)} \right)^2$
        $If\ ABS\left( \frac{f_n - f}{f_n} \right) < 0.01\ Then$
                $f = f_n$
                Exit Loop3
        EndIf
        $f = f_n$
    EndDo Loop3
```

Once the friction factor has been calculated it is applied as follows:

$$\Delta P_{node} = \frac{f_{node} * \Delta X_{node} * G_{node}{}^2}{2 * D_{eq-node} * \rho_{node}}$$

4.24 Conclusion

This section presents a quick one-dimensional method to calculate heat transfer and pressure drop within a straight channel printed circuit heat exchanger using super-critical carbon dioxide or water. This method should be useful for applications that require repeated and rapid analysis such as dynamic simulation until more accurate correlations become available.

4.3 Computational Application

Computationally modeling a S-CO_2 PCHE provides many opportunities for efficiency that should be recognized. While runtime may not be a primary consideration in a steady state analysis, any transient will typically require many thousands of the individual evaluations occurring in a steady state code, therefore making computational efficiency very important.

This section will provide an overview of the highlights of efficient PCHE modeling including simplifying a PCHE to a single axial channel, solving in enthalpy space, averaging in nodes, using nodal memory, and avoiding surrounding loops.

4.31 Using a Single Nodalized Axial Channel

One may take advantage of the regularity in a PCHE's channels by assuming it to be adiabatic to the environment and knowing that there is (typically) a one to one correspondence between hot and cold channels as shown in *Figure 4-3*. This allows the whole PCHE to be reduced to single hot and cold channels as shown in *Figure 4-4: PCHE Single Channel*.

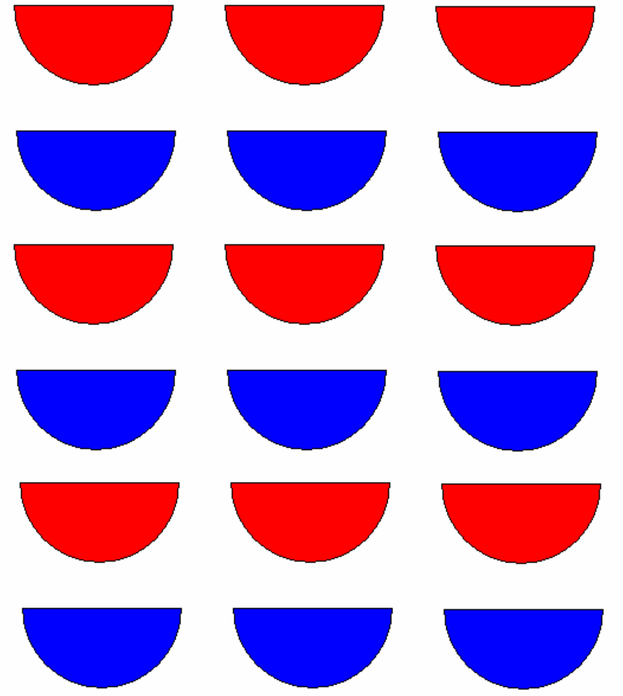


Figure 4-3: PCHE Channel Layout

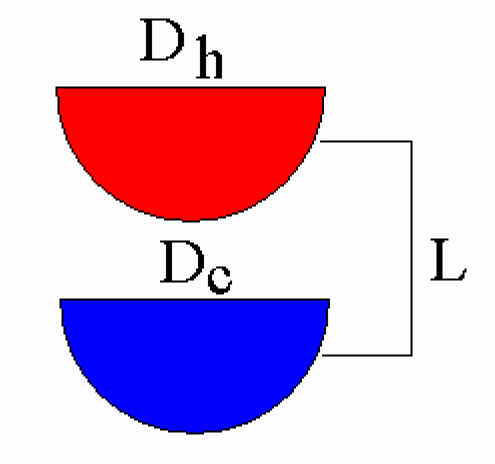


Figure 4-4: PCHE Single Channel

The user may solve these single channels with a one-dimensional (1D) code by assuming that the conditions within a channel are uniform at any given distance into the PCHE. Although this is a significant assumption, the flow is typically highly turbulent, a typical channel diameter is 2 mm and therefore small, and the heat transfer has been found to be insensitive to this assumption[35].

The one-dimensional quick code can then axially break up the PCHE into small sections. By using small enough sections one may assume that properties are constant (within a section) and it becomes relatively simple to arrive at a solution. An example of splitting up the PCHE axially is shown in *Figure 4-5: PCHE Single Channel Nodalized Axially*.

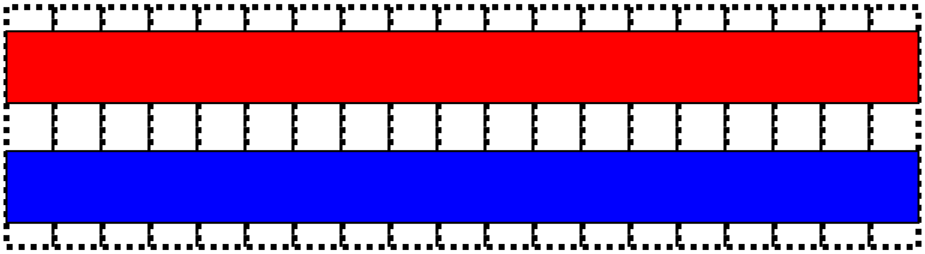


Figure 4-5: PCHE Single Channel Nodalized Axially

A diagram showing the overall computational aspects is shown in *Figure 4-6: PCHE Computation*. The code need only provide the pressure drop and heat transferred in each zone. This data is then used to calculate pressure and enthalpy data points. The pseudo-code for these points is shown in *Table 4-1* and *Table 4-2*. Finally, the pressure and enthalpy are used to calculate all the other useful data at these points that is provided to the user.

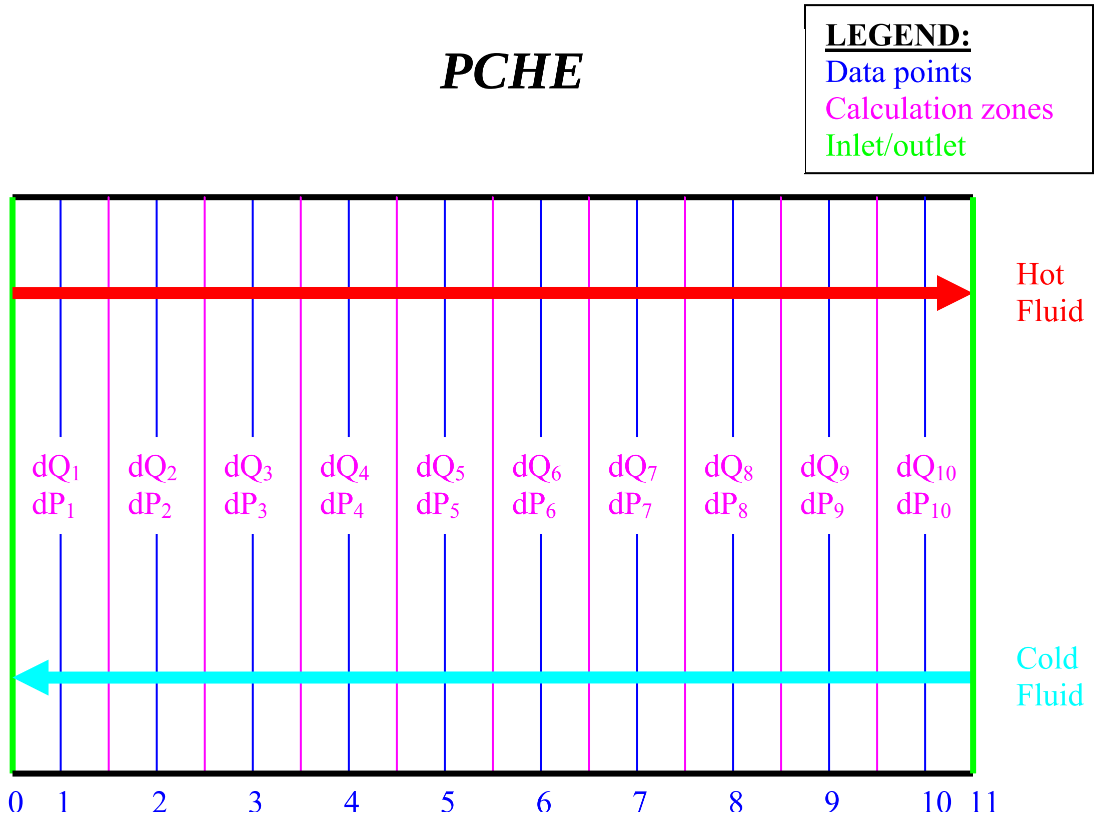


Figure 4-6: PCHE Computation Method

Point	Hot Pressure	Hot Enthalpy
0	$=P_{in}$	$=H_{in}$
1	$=P_0-0.5*dP_1$	$=H_0-0.5*dQ_1/ M_{Cold}$
2	$=P_1-0.5*(dP_1+dP_2)$	$=H_1-0.5*(dQ_1+dQ_2) / M_{Cold}$
3	$=P_2-0.5*(dP_2+dP_3)$	$=H_2-0.5*(dQ_2+dQ_3)/ M_{Cold}$
. . .		
10	$=P_9-0.5*(dP_9+dP_{10})$	$=H_9-0.5*(dQ_9+dQ_{10})/M_{Cold}$
11	$=P_{10}-0.5*dP_{10}$	$=H_{10}-0.5*dQ_{10}/M_{Cold}$

Table 4-1: Hot Fluid Data Points

Point	Cold Pressure	Cold Enthalpy
0	$=P_1-0.5*dP_1$	$=H_1-0.5*dQ_1/M_{Cold}$
1	$=P_2-0.5*(dP_2+dP_1)$	$=H_2-0.5*(dQ_2+dQ_1) /M_{Cold}$
. .		

.		
8	$=P_9-0.5*(dP_9+dP_8)$	$=H_9-0.5*(dQ_9+dQ_8)\ /M_{Cold}$
9	$=P_{10}-0.5*(dP_{10}+dP_9)$	$=H_{10}-0.5*(dQ_{10}+dQ_9)\ /M_{Cold}$
10	$=P_{11}-0.5*dP_{10}$	$=H_{11}-0.5*dQ_{10}/M_{Cold}$
11	=Pin	=Hin

Table 4-2: Cold Fluid Data Points

Two different one-dimensional straight channel printed circuit heat exchanger (PCHE) design codes were previously developed based upon this basis created by Dr. Pavel Hejzlar at MIT, used by Dostal[36], and based upon Williams et. al[37]. This approach will adopted here.

4.32 Solving in Enthalpy Space

The typical person thinks in terms of temperature and pressure but it is far more advantageous to calculate in enthalpy and pressure especially in the S-CO_2 cycle. Where temperature is highly non-linear and relatively rarely used in typical solution code thermodynamic equations, enthalpy is frequently linear and shows up most solution equations.

The precooler clearly shows the advantage of calculating in enthalpy versus temperature. *Figure 4-7* shows the steady state precooler temperature profile. While the water profile is relatively linear the CO_2 profile drops rapidly to a nearly asymptotic limit. Any numerical approach will difficulty in predicted the appropriate CO_2 temperature even when many nodes are used thus requiring numerous evaluations to achieve an accurate solution.

Figure 4-8 shows the enthalpy profile of the precooler in the steady state. This profile shows nearly linear behavior for both the water and the CO_2. Simple linear estimation will provide a relatively accurate enthalpy guess even on the first try. Furthermore, this benefit will extend to subsequent iterations since each guess will be more accurate in a linear profile than a non-linear.

Figure 4-9 is included to show the reader that pressure is relatively linear in a PCHE as well. It shows the pressure profile in the precooler during the steady state.

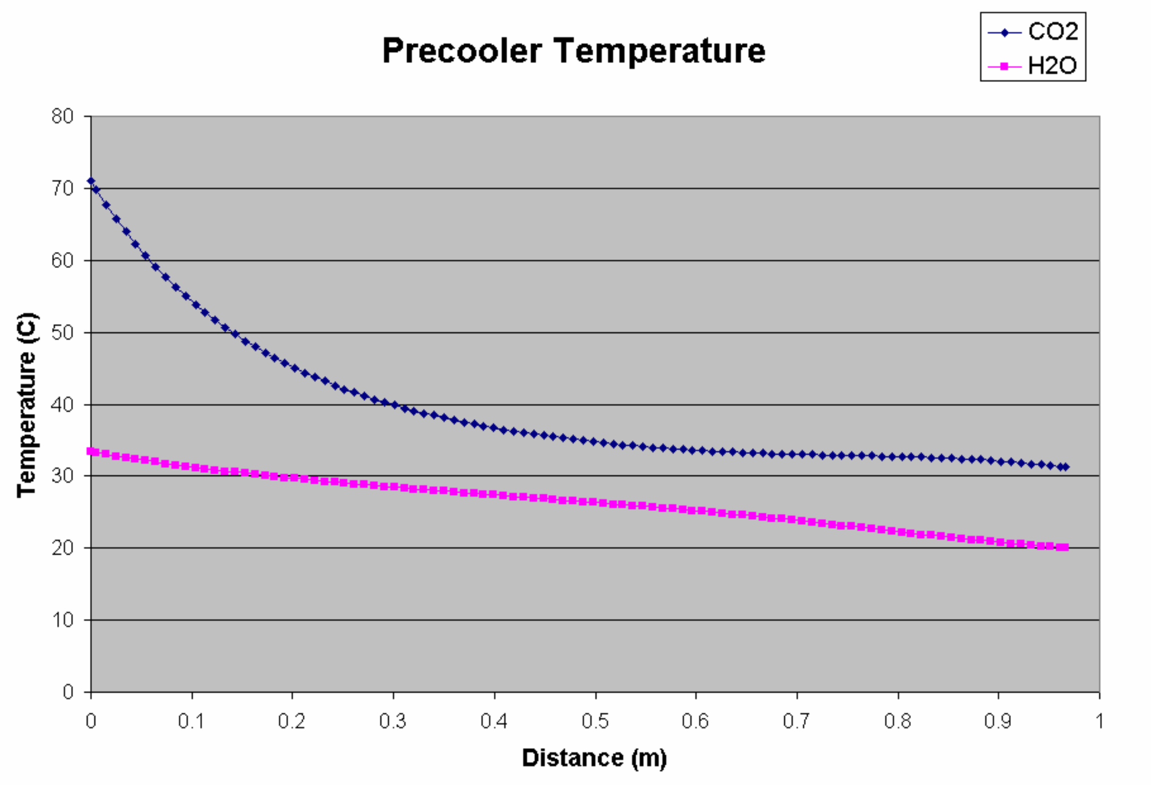


Figure 4-7: Precooler Temperature Profile

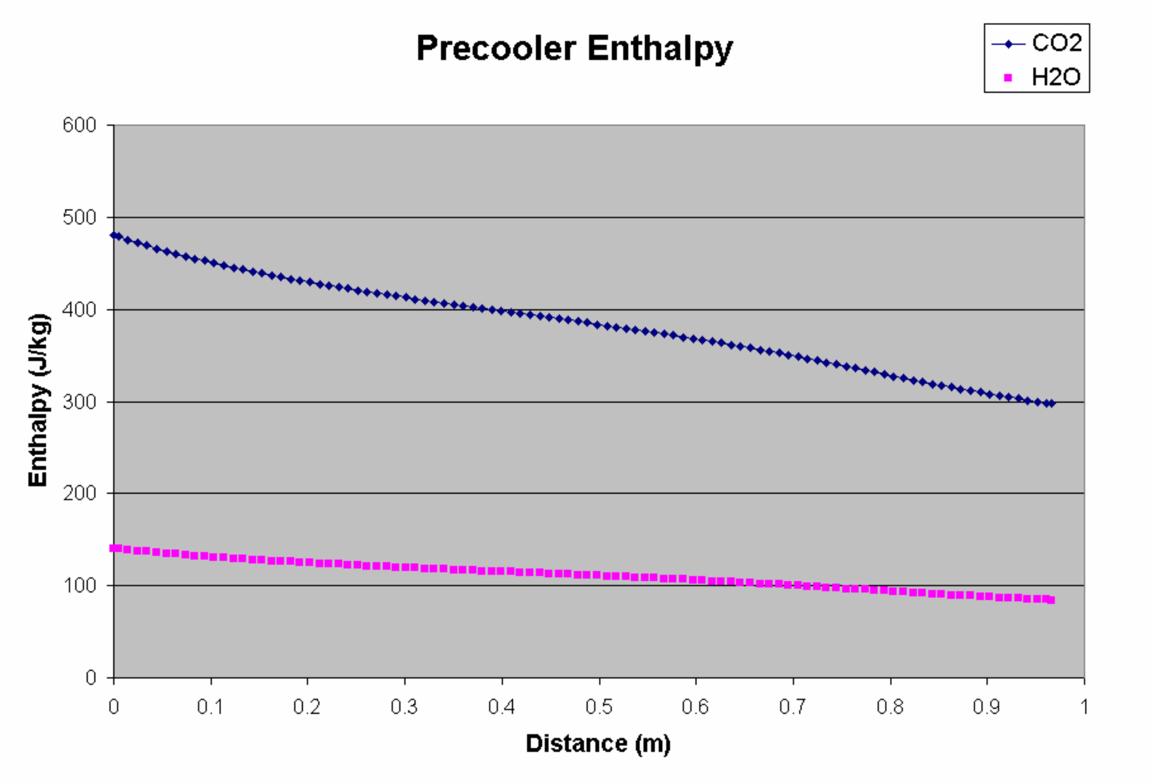


Figure 4-8: Precooler Enthalpy Profile

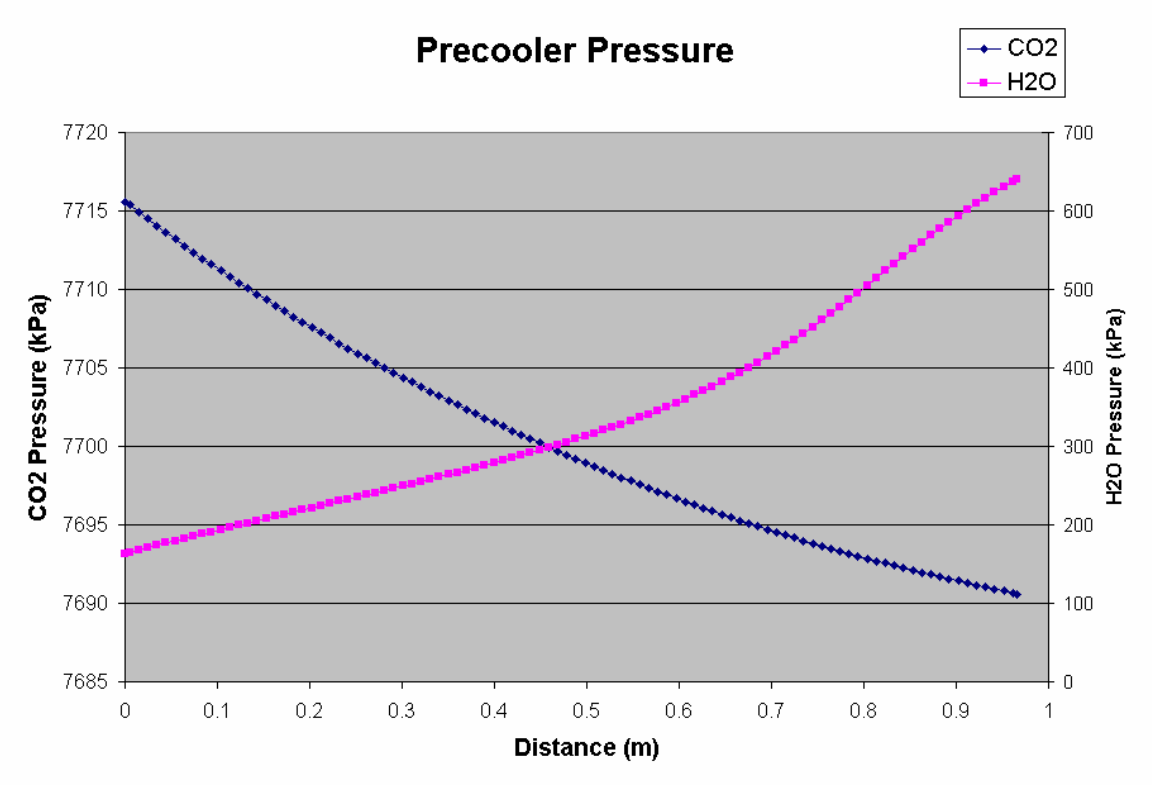


Figure 4-9: Precooler Pressure Profile

It is also important to note that one may greatly decrease the non-linearity of a solution process by solving in enthalpy instead of temperature. If most of the solution equations are in enthalpy (like typical representations of the first law) and one is solving in temperature than this does a great disservice to the solver. While an ideal gas converts proportionally to temperature, S-CO_2 converts highly non-linearly.

Figure 4-10 shows the steady state operating region of the main compressor (in red) on an enthalpy and pressure graph. Incoming and outgoing pressure and temperature lines are show as well. The reader need only glance at the graph to see that small changes in enthalpy can produce relatively large changes in pressure and temperature. Therefore, any solution process that attempts to convert between temperature and enthalpy will need to be highly accurate *solely* because of the property conversion.

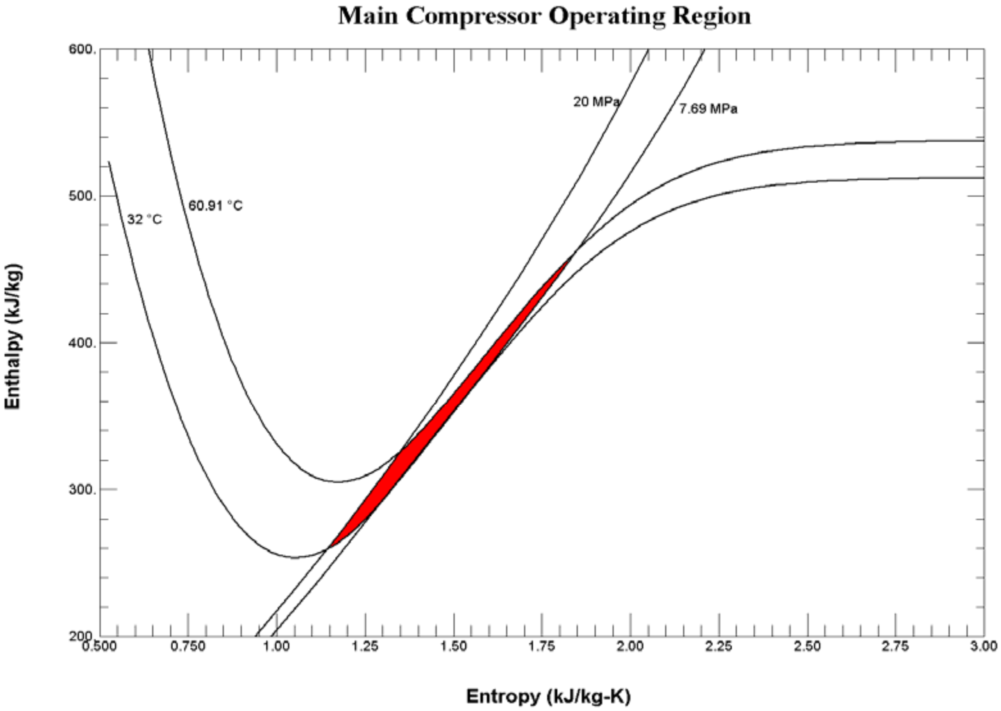


Figure 4-10: Main Compressor Operating Space

This property non-linearity can be clearly seen by comparing the Gas-Pass Jacobians of the S-CO_2 recompression cycle solved in temperature and enthalpy space. *Figure 4-11* shows the original Jacobian when the system was solved in temperature. Large peaks suggest that small changes in the solution variables produce large changes in the solution equations i.e. the system is non-linear. *Figure 4-12* shows a later Gas-Pass Jacobian whose primary change is solving the system in enthalpy and pressure instead of temperature and pressure. The reader will note that there are many few peaks that are much smaller in magnitude i.e. the system is more linear.

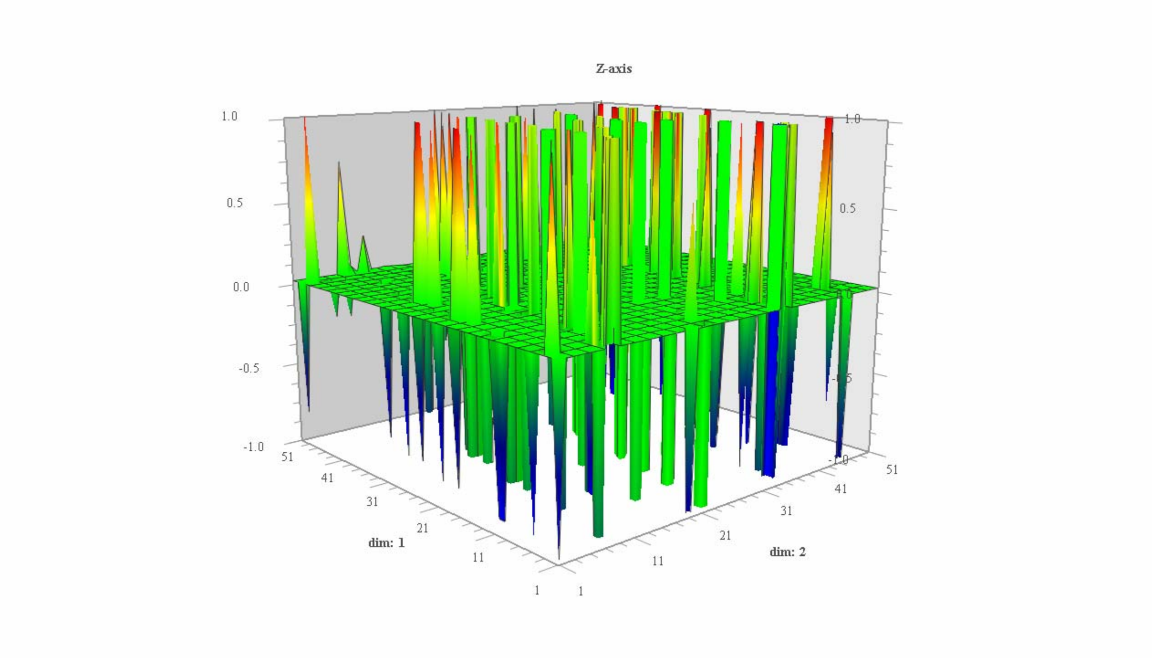


Figure 4-11: Gas-Pass Temperature Jacobian

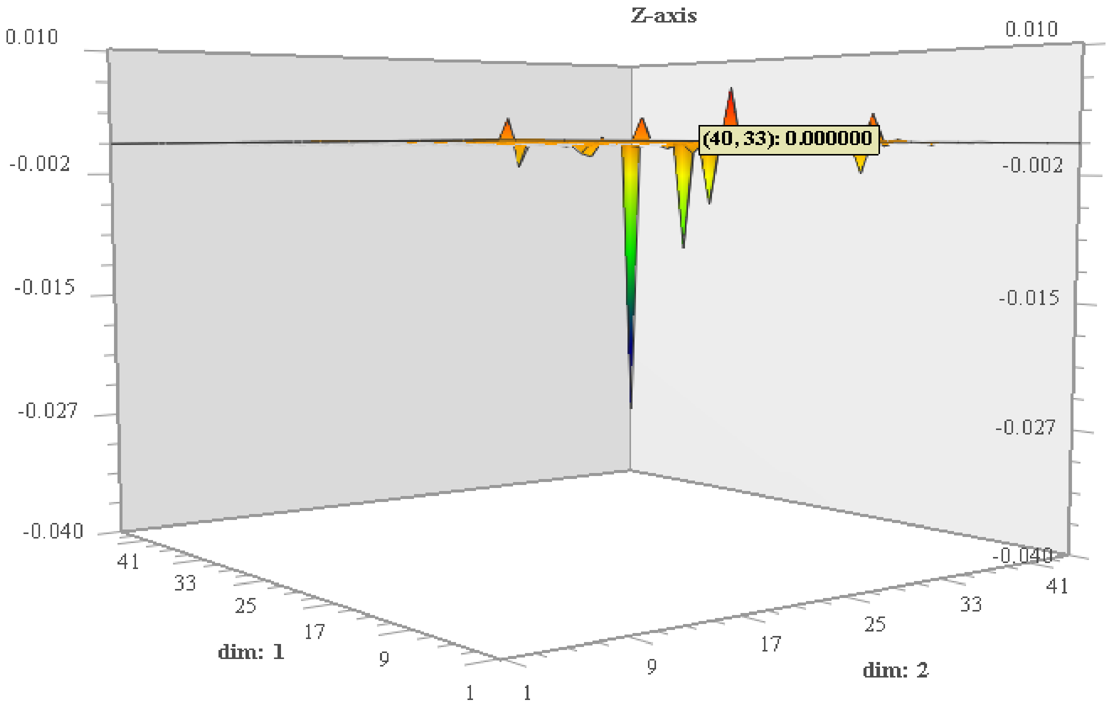


Figure 4-12: Gas-Pass Enthalpy Jacobian

In conclusion one may take advantage of the linear behavior of enthalpy and pressure even in S-CO_2 PCHEs to accurately predict fluid properties. This linearity will allow each guess of a fluid property to be much more accurate than those offered by a highly non-linear property like temperature. Furthermore, by solving in primary variables used in the solution equations one may avoid introducing non-linearity solely from property conversion which can be significant in the S-CO_2 cycle.

4.33 Nodal Averaging

This section is elementary but is included for completeness. As previously mentioned the primary output of the nodalized PCHE code is the heat transfer and pressure drop in each node. Assuming one starts a node with a known solution and a guess of the node's heat transfer and pressure drop it is important to choose and appropriate position within the node to evaluate the guess. The position in the nodes determines the enthalpy and pressure that the correlations will be evaluated at.

The appropriate position in the node is where average values are found. Choosing any other position than the average will introduce an inherent bias in the correlation inputs which will make it more difficult for the node to converge thereby requiring more iterations or more nodes. By choosing enthalpy and pressure (which vary linearly) as the state variables the average values in a node are likely be at the midpoint. This mistake is commonly made since it is more convenient for the programmer to use the node inlet or outlet when first coding a problem.

4.34 Node Memory

One may significantly speedup the calculation of PCHE performance by remembering past nodal values like enthalpy, pressure, and heat transfer. Initial guesses of PCHE performance are likely to be quite crude and require many iterations to converge accurately. Correspondingly, past solutions are likely to be quite similar (and not infrequently identical) to current solutions therefore likely providing a good initial guess. In practice this was observed to speed runtime by nearly an order of magnitude despite using a relatively sophisticated initial performance guess.

However, the reader is warned that memory updating should NOT be done during system Jacobian evaluation. Having PCHE nodes start with different values during each step of the Jacobian calculation will almost surely introduce a significant error in Jacobian that may prevent root finding. Gas-Pass updates the PCHE node memory the first time a new time step is evaluated (but before the numerical solver is called or Jacobian calculated) and then holds the memory constant until the next time step.

4.35 Avoiding Surrounding Loops

Typically, a PCHE will be calculated from the fluid inlets to outlets. For a counter flow heat exchanger a standalone code will require a three step iterative process:

1) Starting with total heat transfer and pressure drop guesses move to one end of the heat exchanger and iteratively solve for each node's pressure drop and heat transfer.
2) Evaluate the error on the pressure drop guess and iteratively refine it.
3) Evaluate the error on the heat transfer guess and iteratively refine it..

Thus three iterative loops are required in a standalone code. However, in an integrated plant code like Gas-Pass it may not be necessary to use the final two loops.

Gas-Pass uses a general non-linear numerical solver which does not distinguish variables by such things as component inputs and outputs – every variable is simply a knob to twist until the equations are satisfied. Therefore, one does not need to supply a PCHE within Gas-Pass with both fluid inlet conditions. One could, and Gas-Pass does, supply a PCHE with fluid properties at a single end of the heat exchanger. Since Gas-Pass is only supplying a guess of the PCHE inputs to look at the PCHE outputs (to see if the system equations are satisfied) there is *no* reason to solver for both fluid outlet instead of one fluid outlet and one fluid inlet – they are both system variables.

By solving from one end of the PCHE Gas-Pass does not need the two outer loops required of a standalone code and therefore should calculate PCHE performance by at least one order of magnitude faster than it would by evaluating from the fluid inlets.

4.35 Conclusion

One may use basic observations to greatly speed the calculation of PCHE performance. By carefully setting up the system one may benefit significantly. An example of the estimated speedups of the various factors are offered below.

Starting with the axial 1D nodalized code outline in section FILL IN:

1. Using enthalpy instead of temperature and calculating at the node midpoint instead of at the beginning of the node allows one to reduce the number of computational nodes from approximately 40 to approximately 2 for the same accuracy therefore it is a 20 fold speedup.
2. Using node memory to speedup PCHE calculations decreases runtime about 10x.
3. Avoiding the outer two iteration loops speeds up the PCHE calculation by about 50x.

Thus we have speedup our PCHE calculation by roughly 10,000x without any loss in accuracy. Since the PCHE is the most computationally demanding component in Gas-Pass one can probably scale Gas-Pass runtime by the same factor. Furthermore, if one counts the greater linearity of solving a S-CO_2 system where variables are in the same property as is commonly used in solution equations (enthalpy instead of temperature) then we can probably add at least another order of magnitude in speedup thus suggesting that Gas-Pass may run 5 orders of magnitude faster than it might by simply adopting already simplified standalone codes.

4.4 Steady State PCHE Results

This section shows the performance of S-CO_2 recompression cycle PCHEs in both steady state and several anticipated transient conditions. The conditions are meant to provide an overview of possible performance during plant operation. All cases are for the 650°C turbine inlet temperature design.

4.41 Precooler Heat Exchanger Performance

The precooler has hot S-CO_2 coming into the precooler and being cooled by liquid water that is external to the thermodynamic cycle. It is the only heat sink for the recompression cycle.

Several cases were computed:

- Normal/steady state
- CO_2 inlet pressure = 8 MPa
- CO_2 inlet pressure = 7.5 MPa
- Mass flow rates from 0.1-1.1 x normal

In all cases the inlet CO_2 temperature was varied between 40°C and 100°C (nominal is 70.953°C). Highlights of these runs are shown in *Figure 4-13: Steady State CO2 Cp* through *Figure 4-21: CO2 Cp Profiles at 110% Mass Flow Rate*.

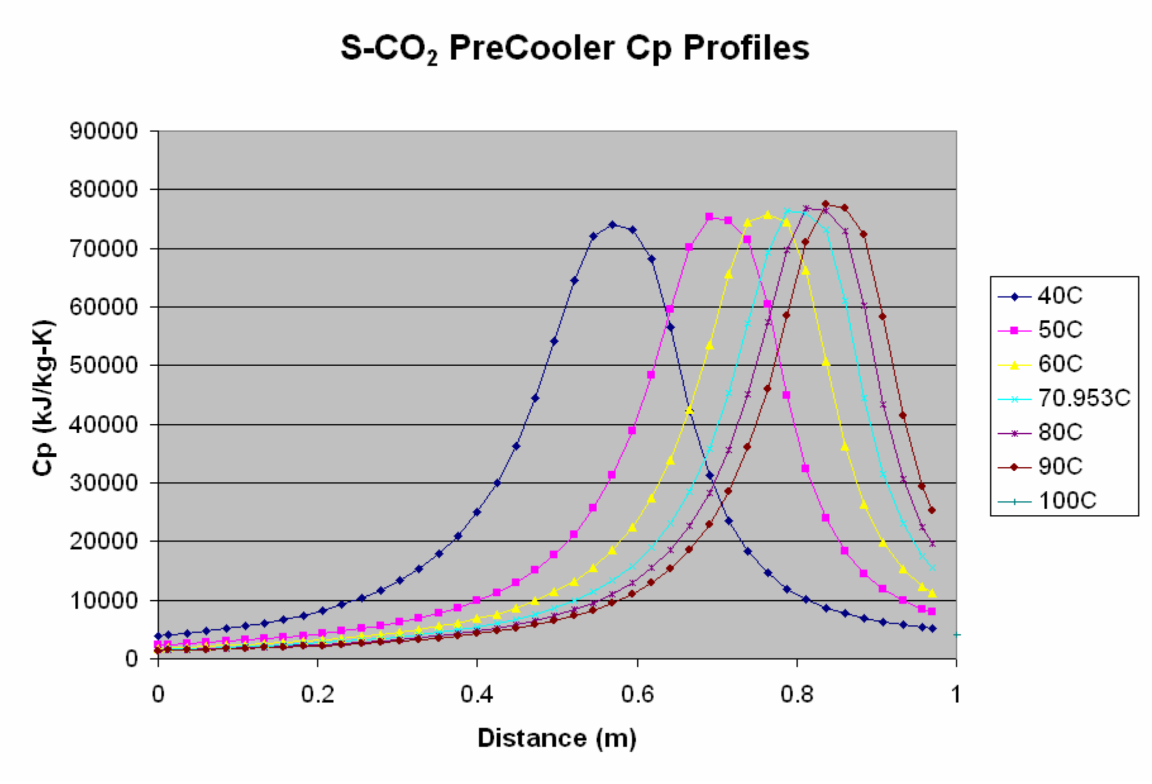


Figure 4-13: Steady State CO_2 Cp

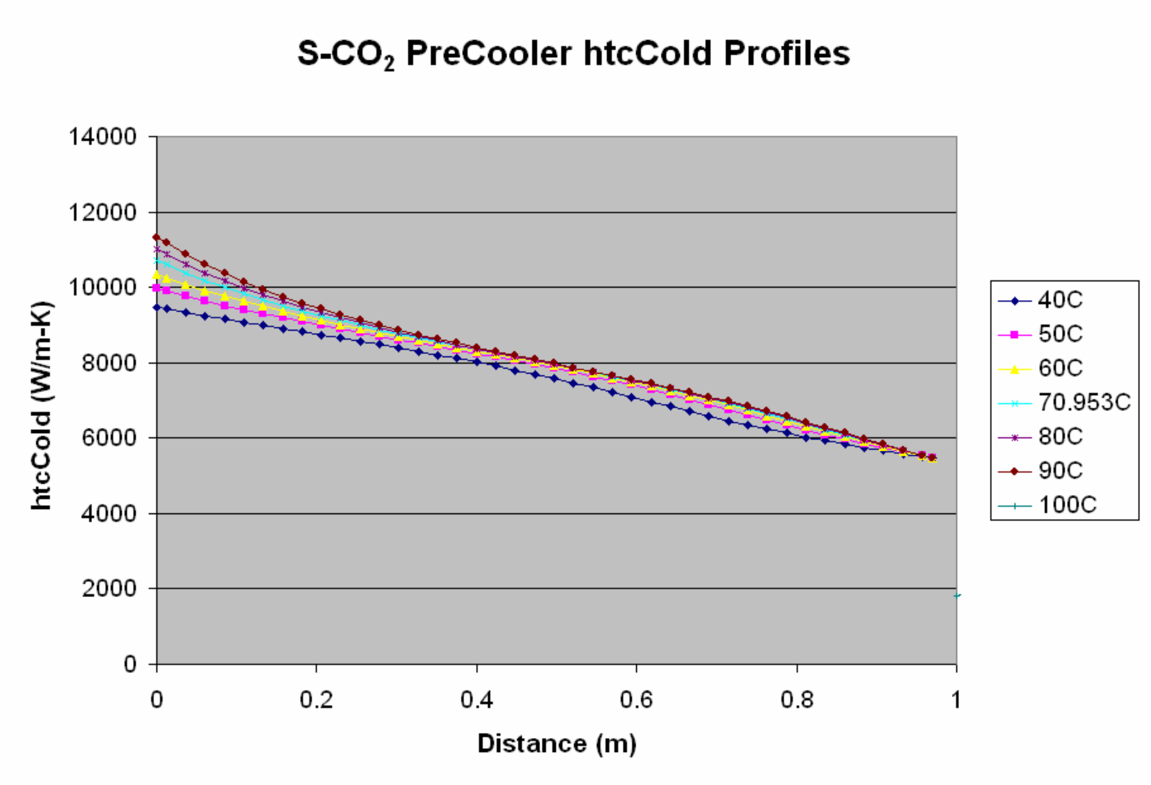


Figure 4-14: Steady State Water HTC

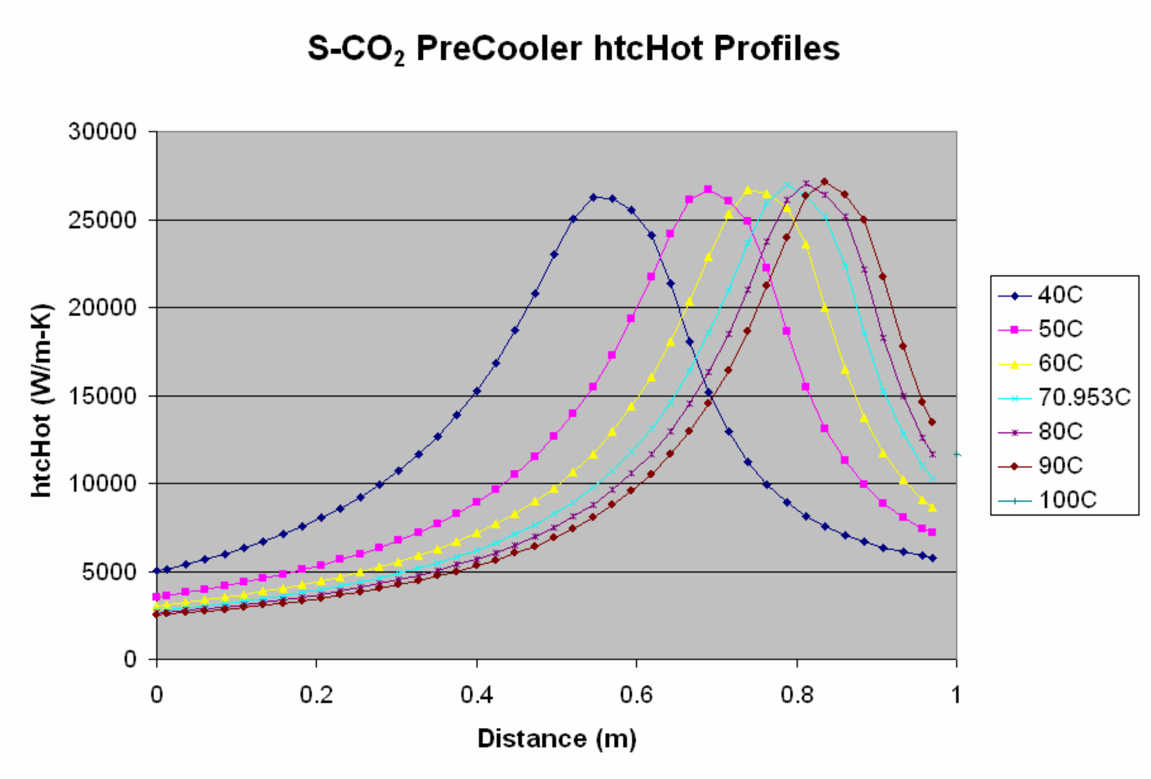


Figure 4-15: Steady State CO$_2$ HTC

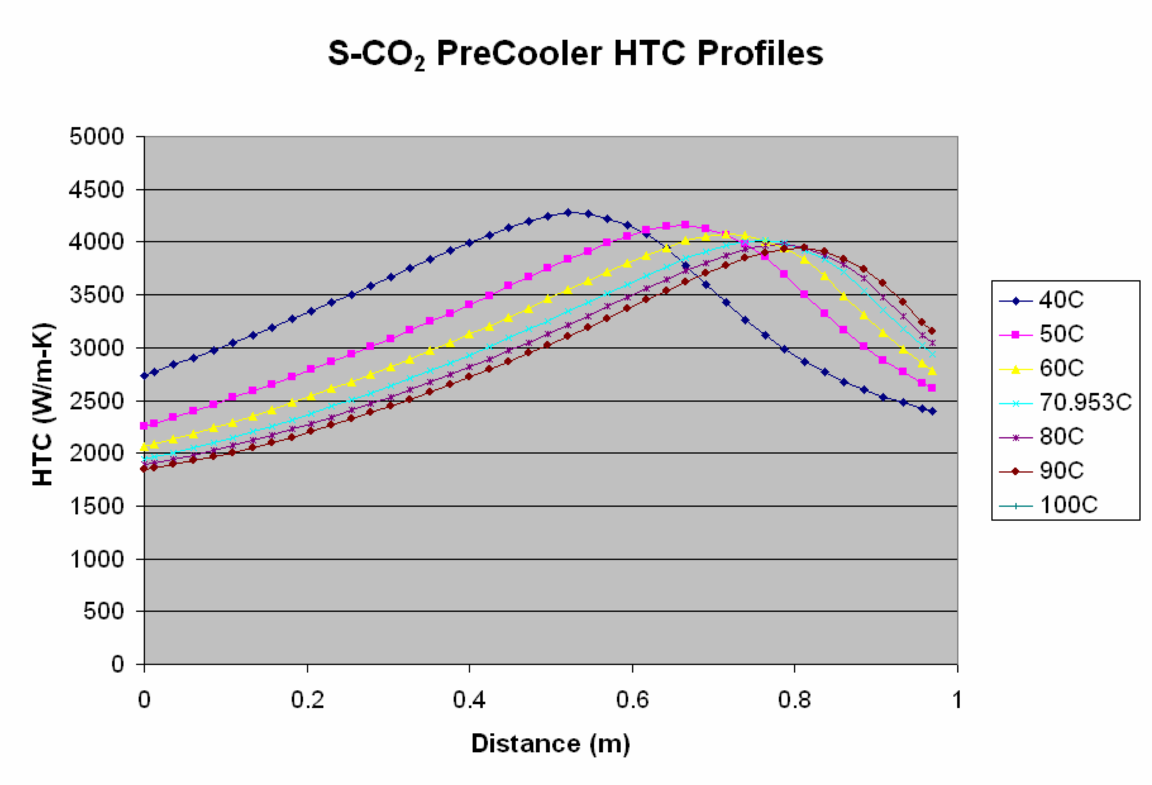


Figure 4-16: Steady State Overall HTC

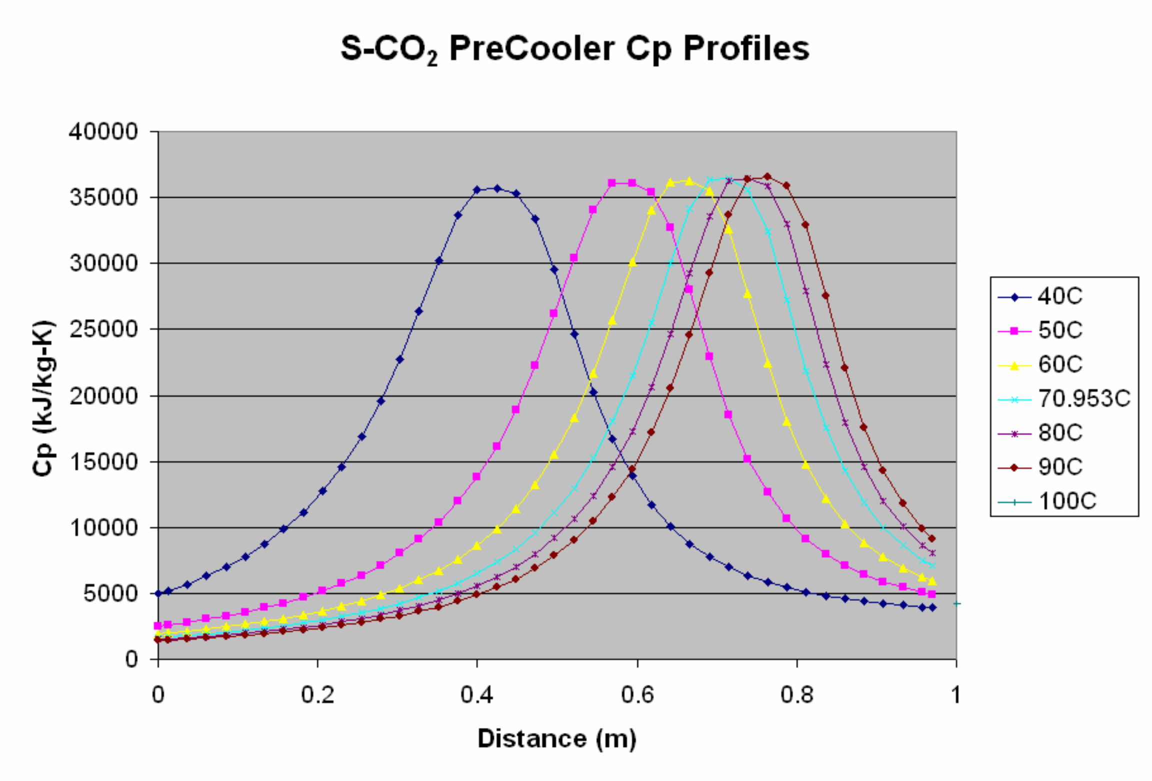


Figure 4-17: CO$_2$ Pin=8MPa Cp

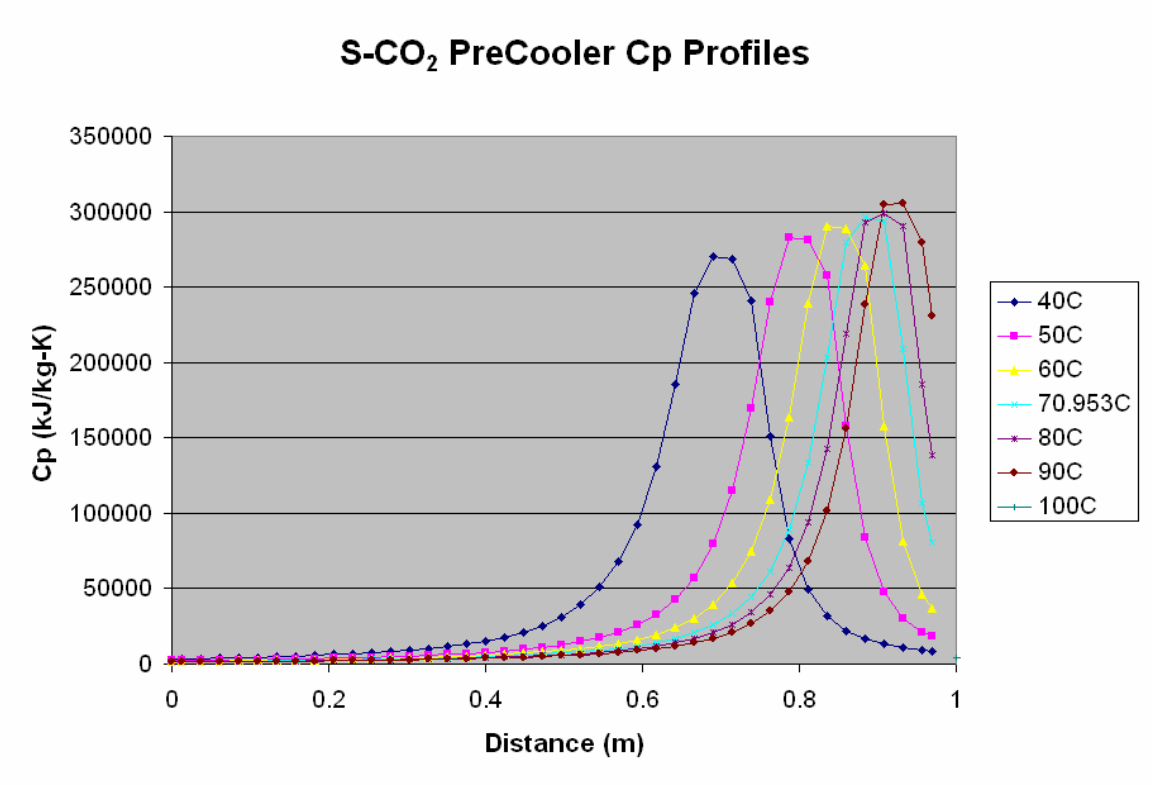


Figure 4-18: CO$_2$ Pin=7.5MPa Cp

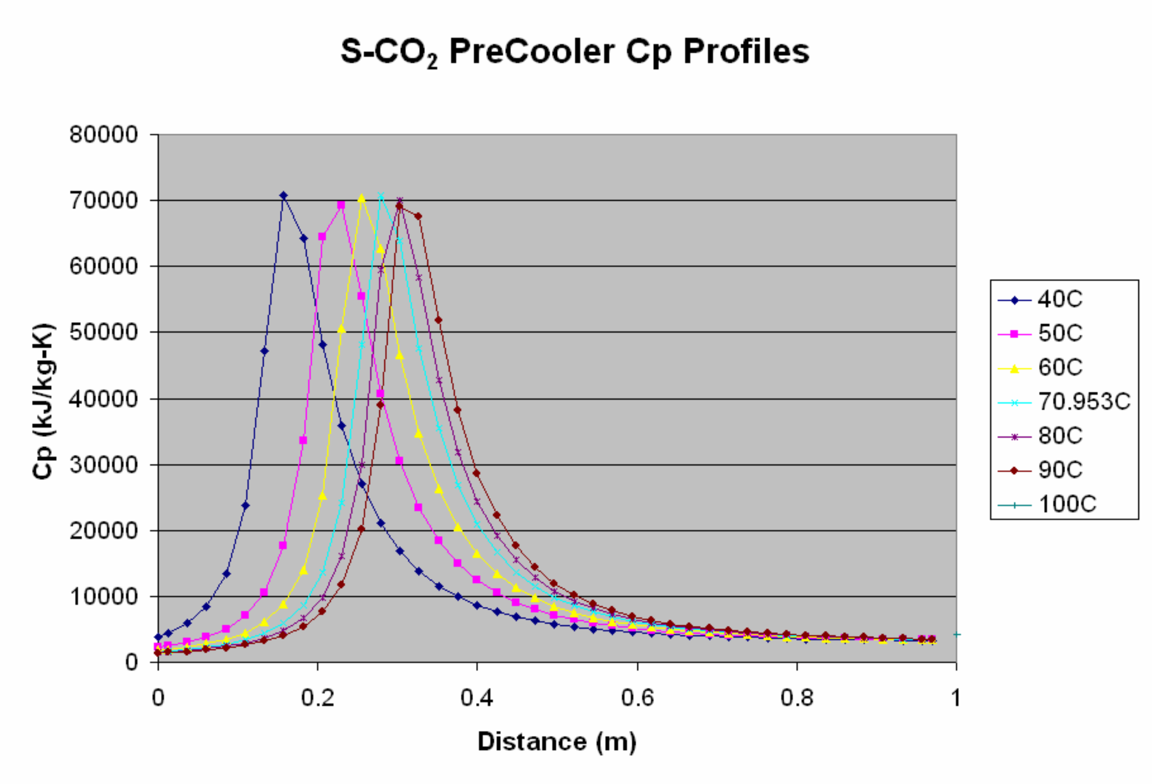


Figure 4-19: CO$_2$ Cp Profiles at 10% Mass Flow Rate

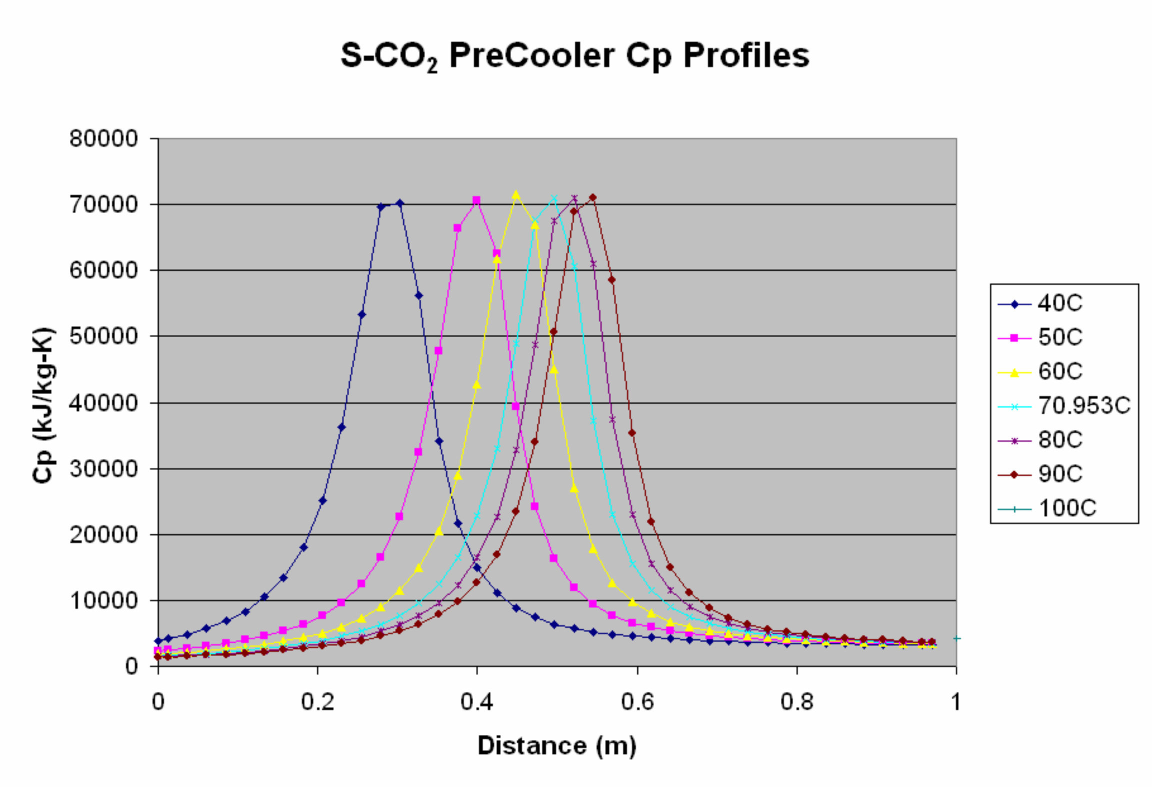


Figure 4-20: CO$_2$ Cp Profiles at 50% Mass Flow Rate

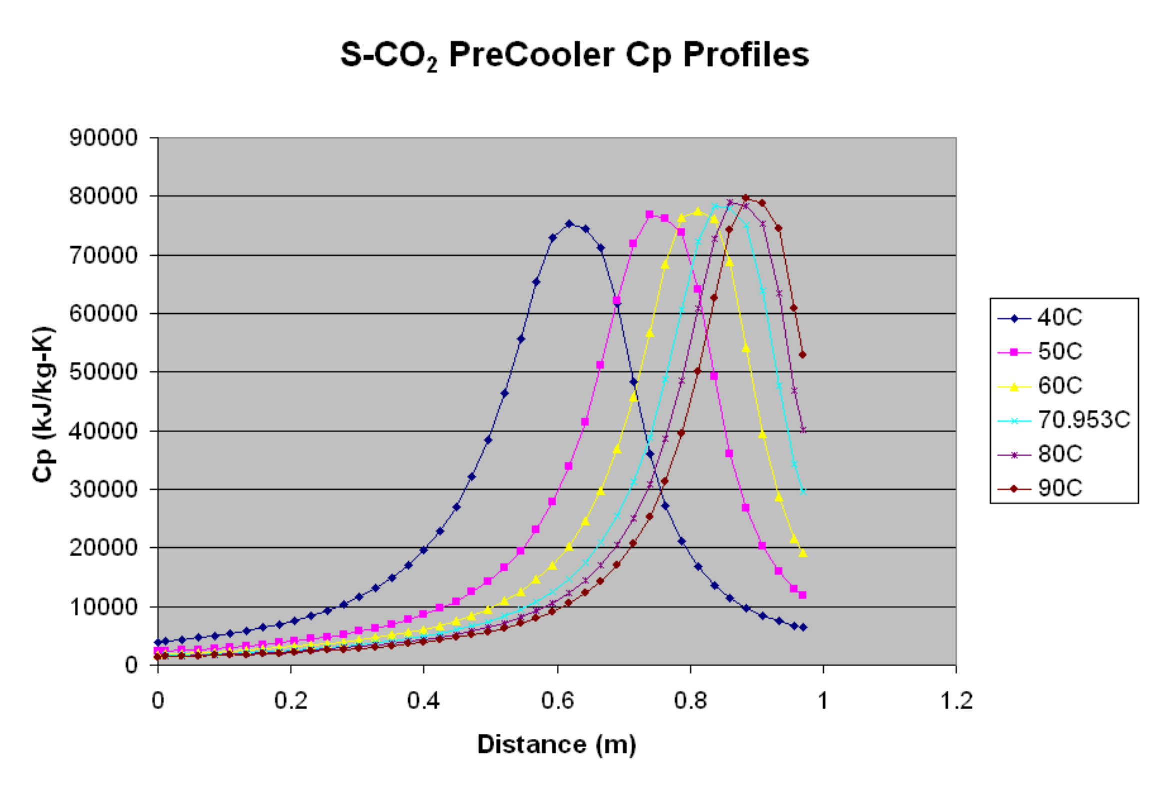


Figure 4-21: CO$_2$ Cp Profiles at 110% Mass Flow Rate

The plots suggest several features. CO_2 in the precooler features an isobaric specific heat peak that moves within the PCHE with temperature and mass flow rate. Lower temperatures and lower mass flow rates move the peak towards the CO_2 inlet and vice versa. The operating pressure largely determines the height of the Cp peak. As the pressure approaches the critical pressure (the critical point occurs at 30.978 °C, 7.3773 MPa)[38] the peak magnitude increases rapidly, and the reverse occurs as well.

The water heat transfer coefficient is roughly linear. It slightly peaks at the CO_2 inlet where the temperature difference is large. The CO_2 heat transfer coefficient shows peaks similar to that of the isobaric specific heat. Its behavior is significantly affected by the Cp profile and will show related behavior. The overall heat transfer coefficient is a combination of the two heat transfer coefficients.

4.42 Low Temperature Recuperator Performance

The low temperature recuperator has hot S-CO_2 at the full mass flow rate being cooled by cold S-CO_2 at approximately 60% of the full flow rate. The cold side flow split avoids a recuperator pinch point that would significantly decrease cycle efficiency.

Several cases were computed:

- Normal/steady state
- Mass flow rates from 0.1-1.1 x normal (assuming each side's mass flow rate is changed by the same percentage)

In all cases the inlet cold CO_2 temperature was varied between 40°C and 80°C (nominal is 60.9°C). Highlights of these runs are shown in *Figure 4-22: LTR Steady State HTCs* through *Figure 4-26: LTR 110% Mass Flow Rate HTRs*.

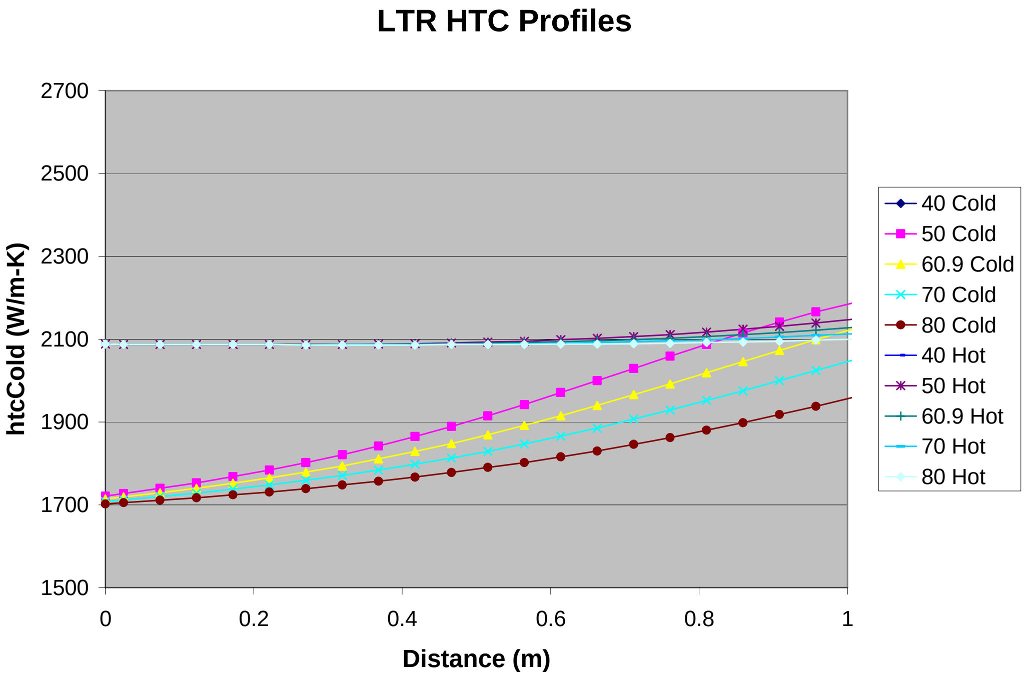


Figure 4-22: LTR Steady State HTCs

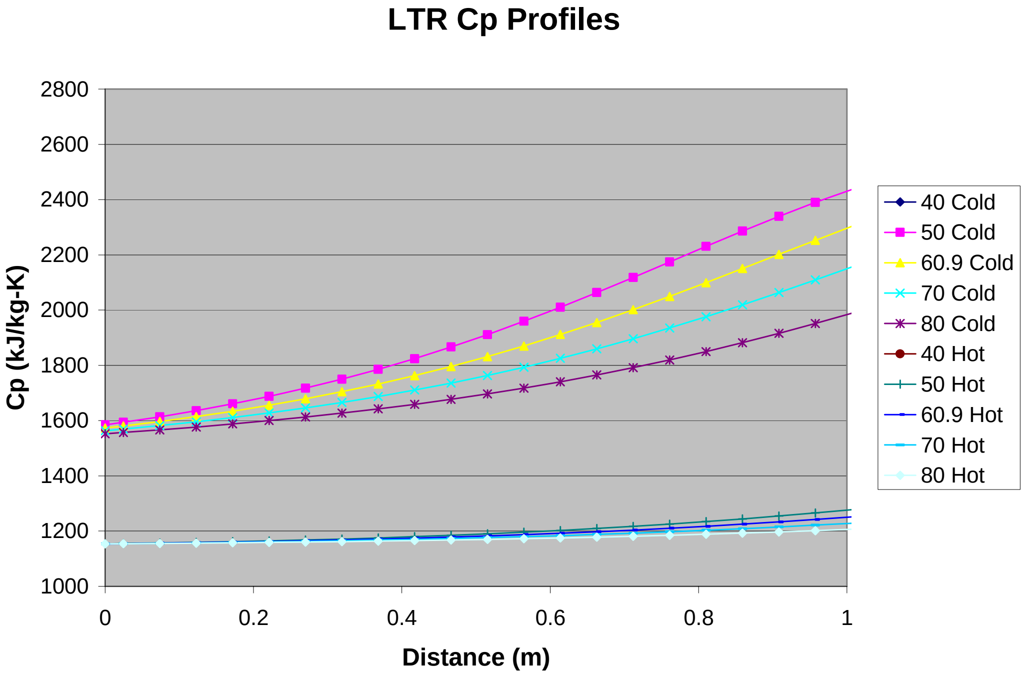


Figure 4-23: LTR Steady State Cps

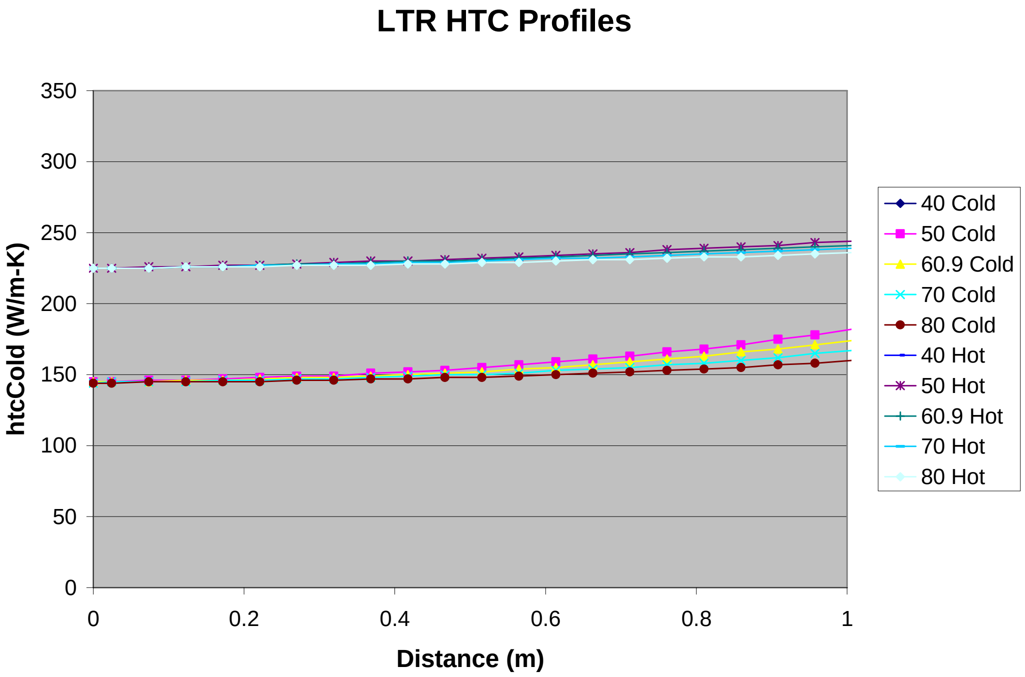


Figure 4-24: LTR 10% Mass Flow Rate HTRs

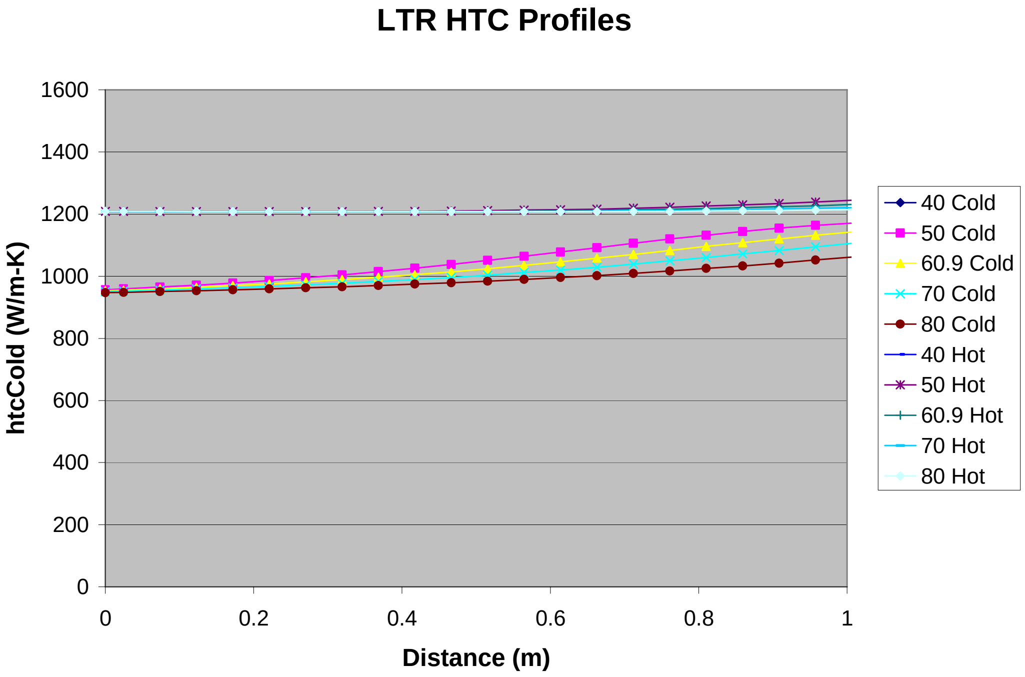


Figure 4-25: LTR 50% Mass Flow Rate HTRs

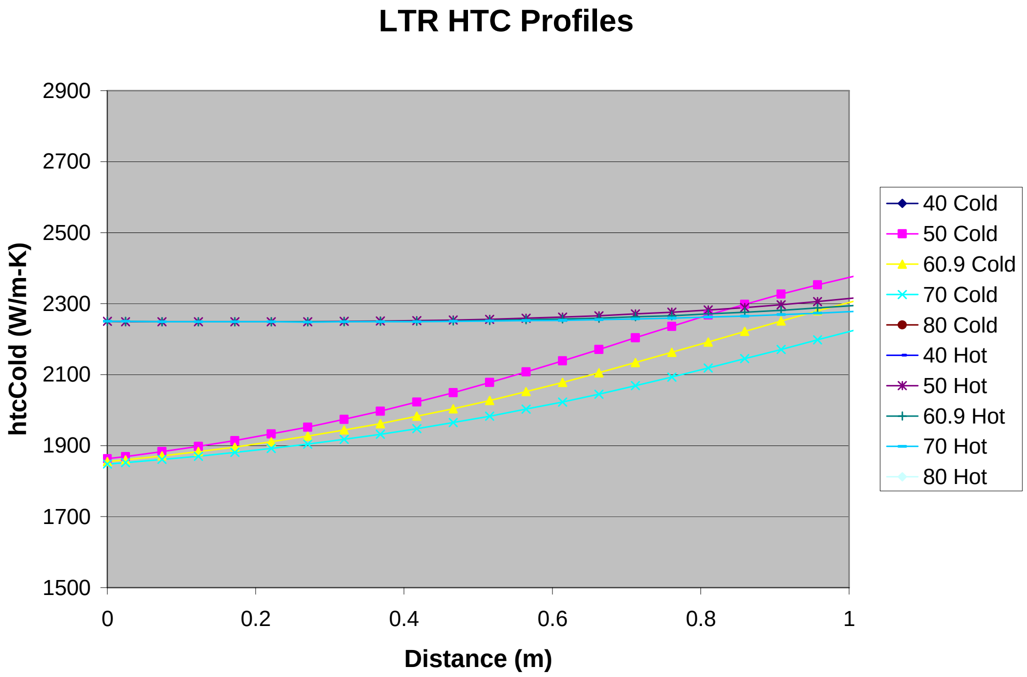


Figure 4-26: LTR 110% Mass Flow Rate HTRs

The cold side heat transfer coefficients clearly vary while the hot side heat transfer coefficients are much more linear but do change (at 60.9°C the HTC varies by 16% from inlet to outlet). The shapes of the HTR profiles are not qualitatively changed by changes in the mass flow rate but the magnitude varies with the flow rate.

The cold side isobaric specific heat clearly varies while the hot side Cp is much more linear but does change (at 60.9°C the Cp value varies by 37% from inlet to outlet). The Cps do not change significantly with changes in mass flow rate.

4.43 High Temperature Recuperator Performance

The high temperature recuperator has hot S-CO_2 cooled by cold S-CO_2 both at 100% of the full flow rate. Due to the similarity between the LTR and HTR and since the HTR is further from the critical point than the LTR only the normal operating mass flow rate was calculated. The inlet cold CO_2 temperature was varied between 140°C and 180°C (nominal is 159.114°C). Highlights of these runs are shown in *Figure 4-27: HTR Steady State HTCs* through *Figure 4-28: HTR Steady State Cps*.

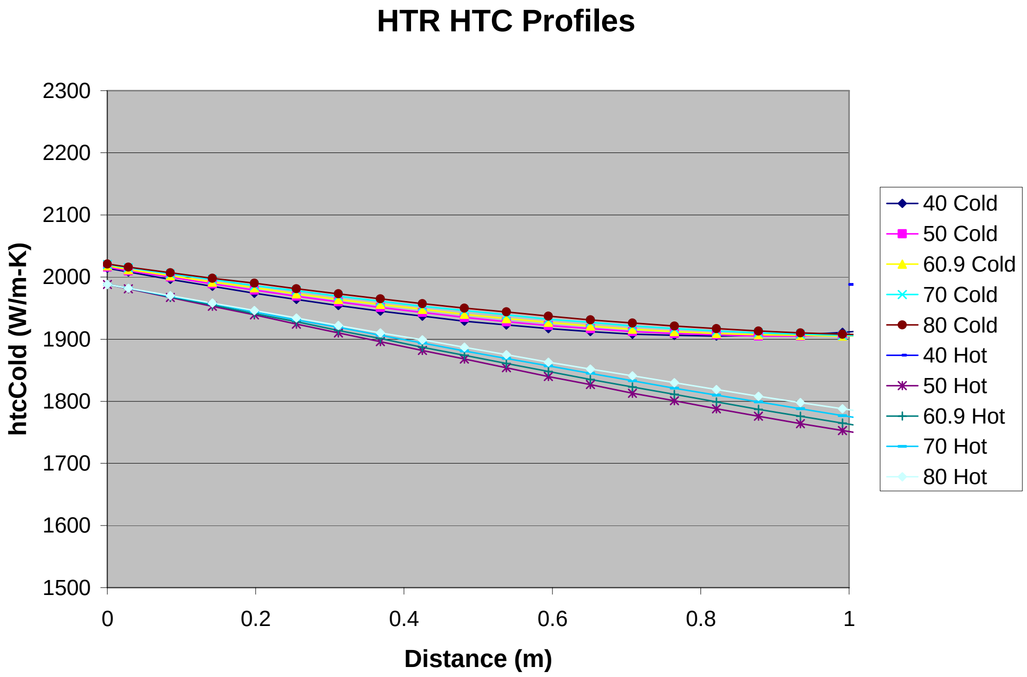


Figure 4-27: HTR Steady State HTCs

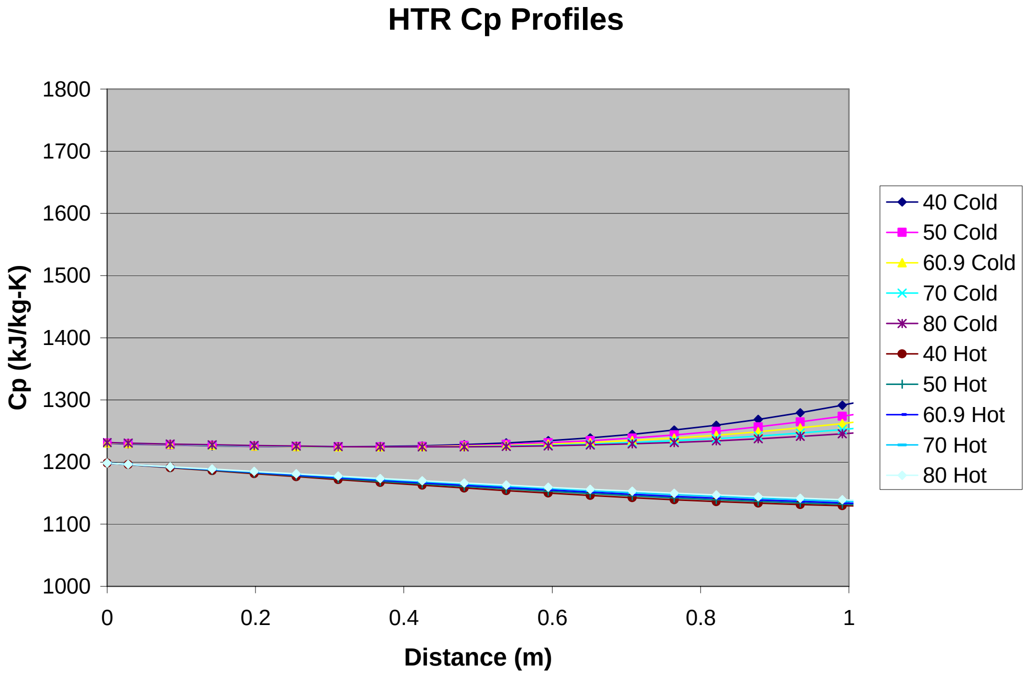


Figure 4-28: HTR Steady State Cps

Both the hot HTR and Cp profiles shown quasi-linear performance while the cold side shows some curvature at its inlet to the PCHE. In both cases the cold inlet temperature makes relatively little difference after the cold fluid has proceeded a significant distance into the PCHE.

4.44 Intermediate Heat Exchanger Performance

TO BE COMPLETED

Conclusion

Initial studies with the PCHEs suggest that only the isobaric specific heat of water in the precooler may be accurately treated as a constant. Standard assumptions like log-mean temperature difference do not hold due to the varying specific heats and heat transfer coefficients experience with CO_2. The low temperature and especially the high temperature recuperator show roughly linear Cp and HTC profiles but the precooler exhibits more complex behavior.

4.5 Precooler Thermal Inertia

This section will analyze and will attempt to characterize the thermal inertia present in the S-CO_2 recompression cycle precooler. The precooler design is preliminary but it presents the current best design.

4.51: Precooler Background

The precooler is a single module, straight channel, printed circuit heat exchanger (PCHE) with the geometry shown in *Table 4-3: Precooler Geometry*. The volumes within the precooler are shown in *Table 4-4: Precooler Volumes*.

Component	Distance (m)
Hot channel diameter	0.002
Cold channel diameter	0.002
Hot pitch divider	0.005
Cold pitch divider	0.005
PCHE height	18.52
PCHE width	0.6
PCHE Length	0.9664

Table 4-3: Precooler Geometry

Component	Volume (m^3)
CO_2	2.395
H_2O	2.395
Titanium	5.949

Table 4-4: Precooler Volumes

The precooler is constructed out of titanium with a thermal conductivity of 25 [W/m-K] and a surface roughness of $1E^{-5}$ [m]. In the recompression cycle with a 650°C reactor outlet temperature and 300 MWe design the precooler transfers 295 MW. The PCHE is cross flow with the inlet and outlet properties shown in *Table 4-5: Precooler Inlet and Outlet Properties*.

Fluid	In	Out
CO_2 Temperature (°C)	70.95	31.99
CO_2 Pressure (MPa)	7.7516	7.6899
H_2O Temperature (°C)	20.00	32.72
H_2O Pressure (MPa)	0.50132	0.39505

Table 4-5: Precooler Inlet and Outlet Properties

The steady state precooler internal profiles are shown in *Figure 4-29: Precooler Temperatures* through *Figure 4-35: Precooler Heat Transfer*. These figures are included for reference purposes only but the reader should note the CO_2 isobaric specific heat peak and how it ends up affecting the heat transfer.

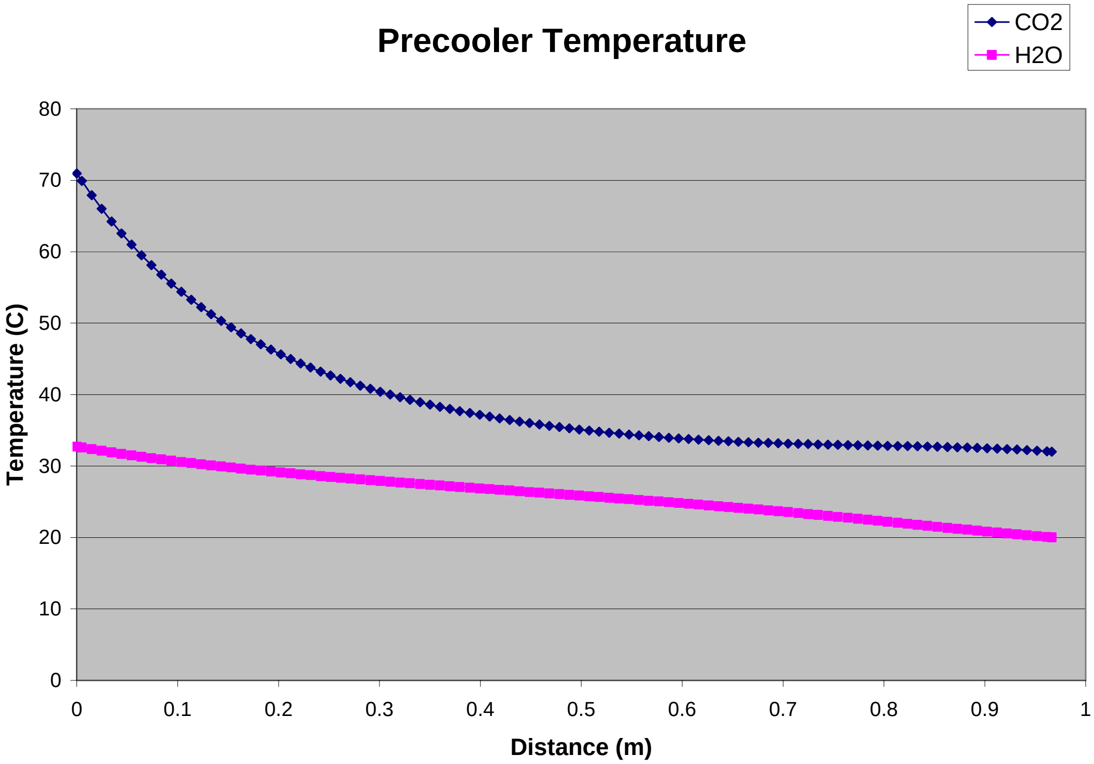


Figure 4-29: Precooler Temperatures

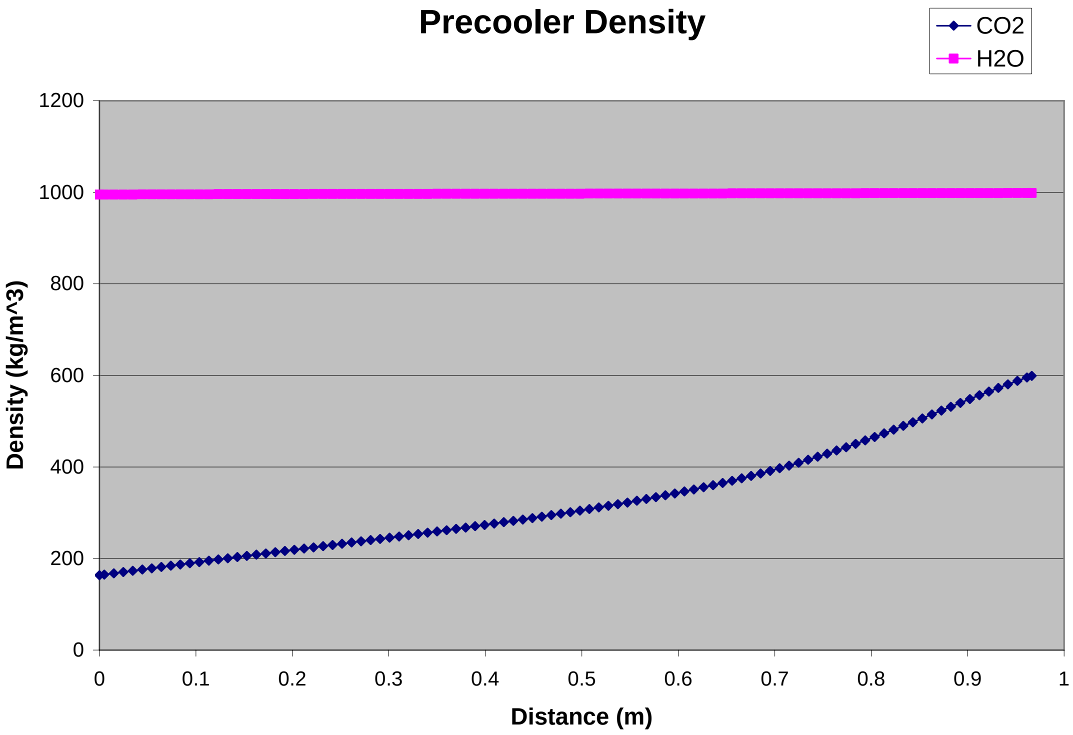


Figure 4-30: Precooler Density

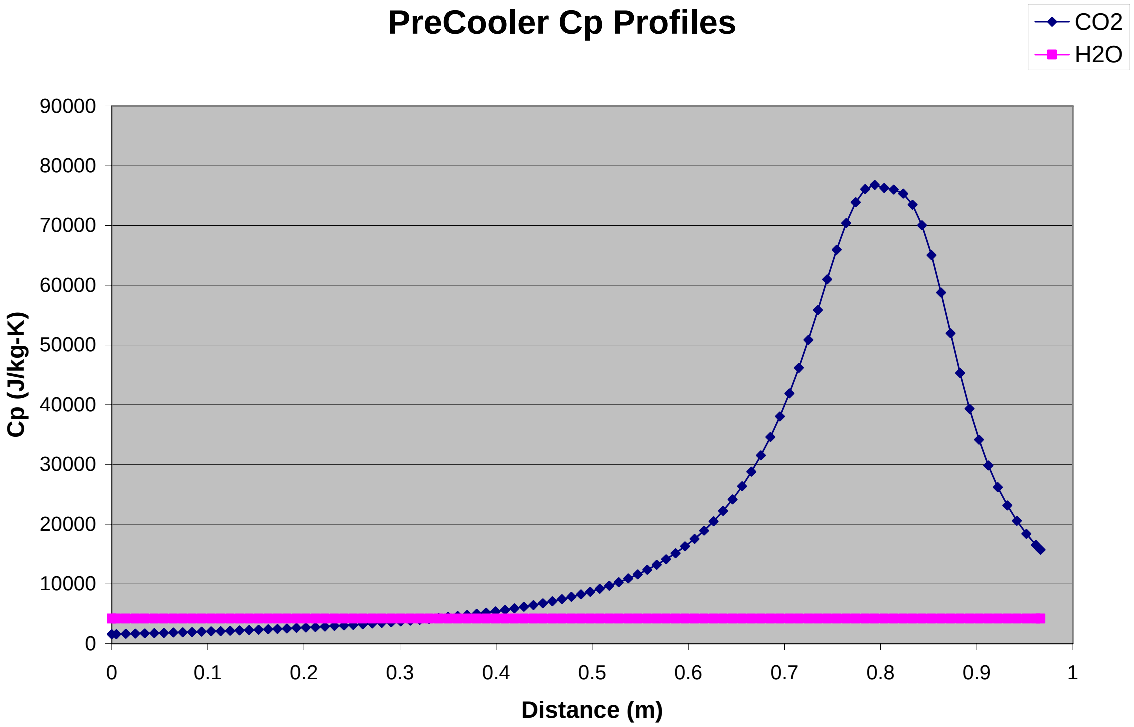


Figure 4-31: Precooler Isobaric Specific Heat

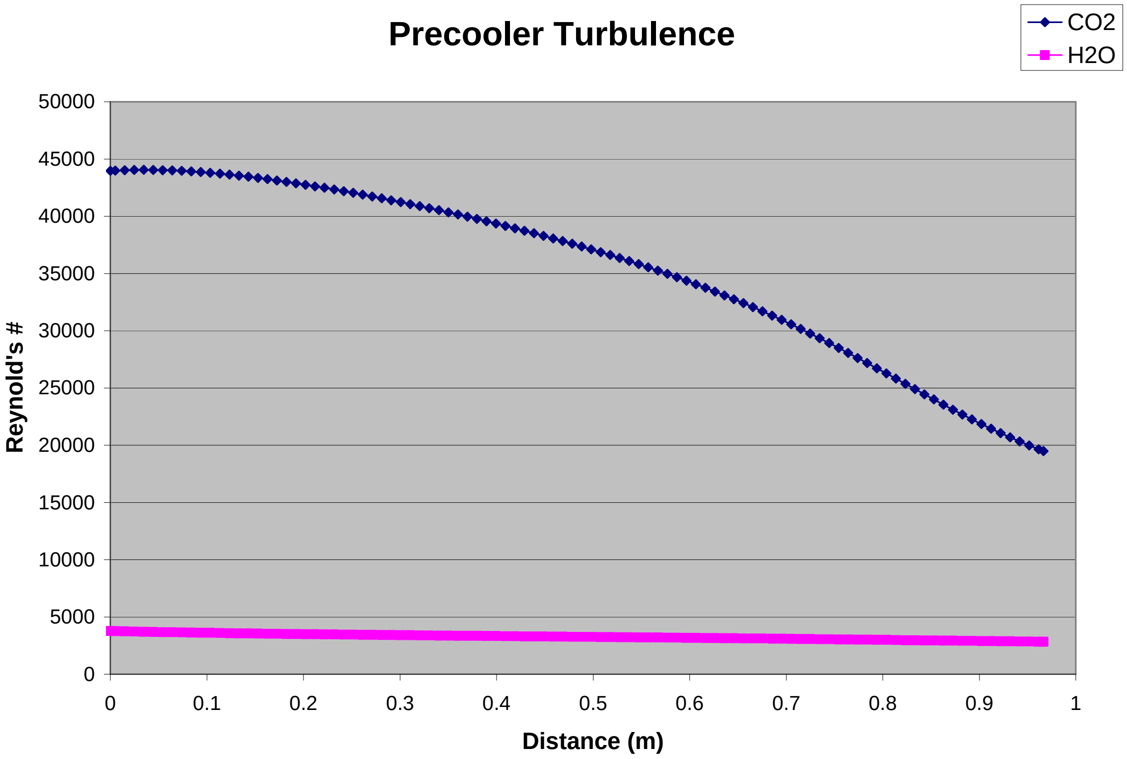


Figure 4-32: Precooler Reynolds #'s

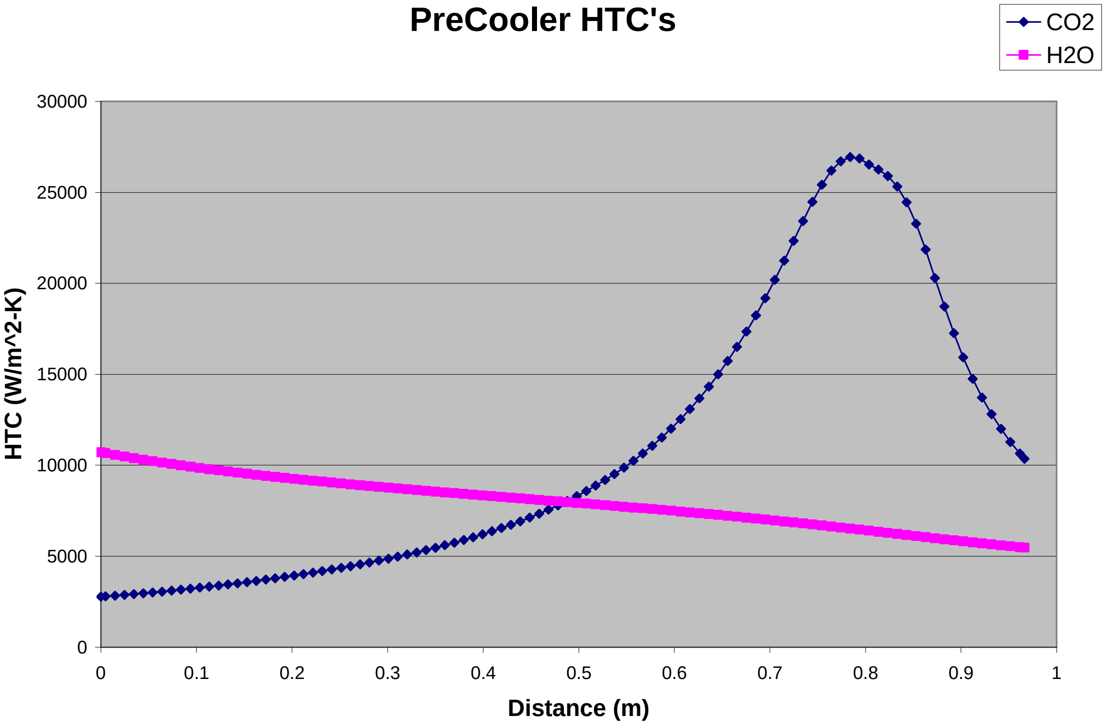


Figure 4-33: Precooler HTCs

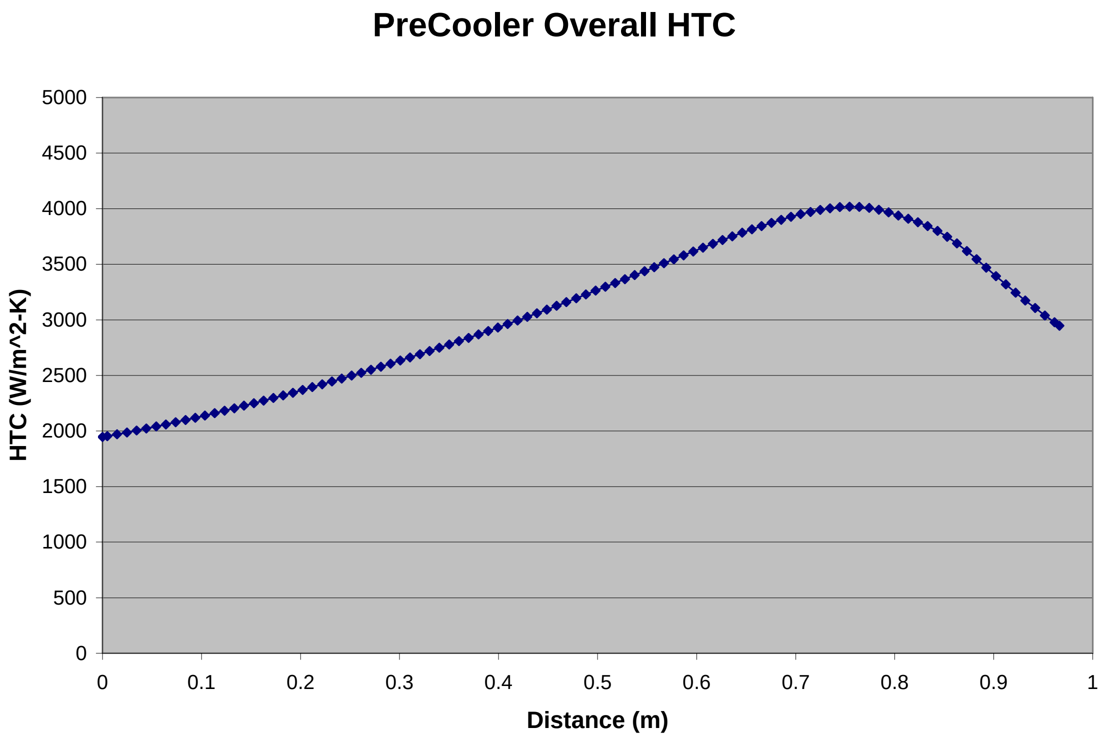


Figure 4-34: Precooler Overall HTC

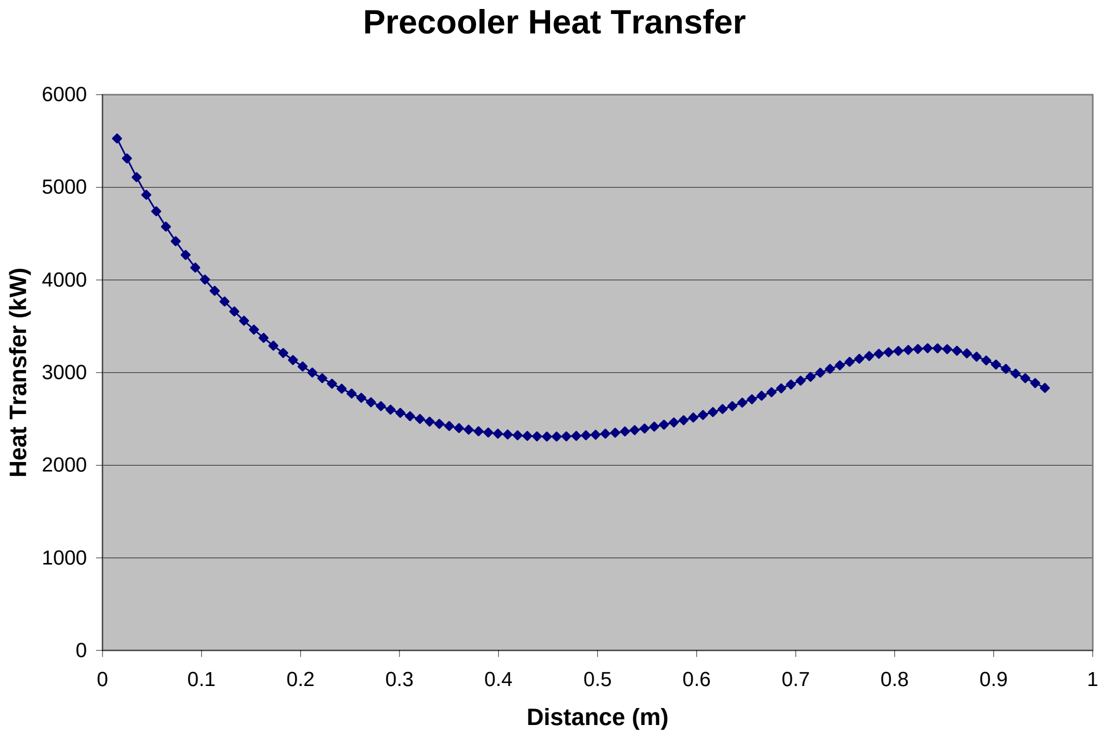


Figure 4-35: Precooler Heat Transfer

4.52: Precooler Thermal Inertia Analysis

The thermal inertia of the precooler is not simple to characterize due to the complexity of the component. However, several general observations may be readily drawn.

The Biot Number is a measure of the internal versus external resistance to heat transfer. Looking at this number from the perspective of the titanium block we can calculate this number for both fluids. This results in *Figure 4-36: Precooler Biot #*.

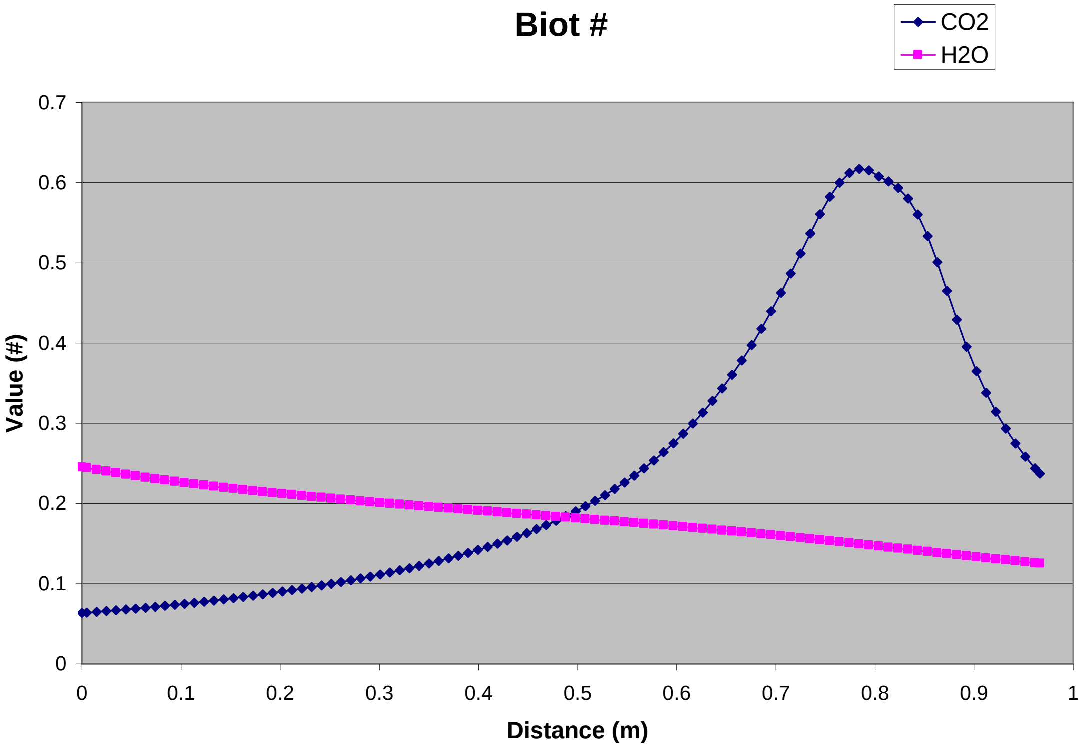


Figure 4-36: Precooler Biot #

These figures suggest that external (i.e. the fluid's) resistance is the dominant barrier to heat transfer but that the metal's resistance is somewhat comparable. The Biot # with CO_2 ranges from a system almost wholly dominated by the fluid resistance at the entrance to a system partially determined by the metal resistance near 0.8 meters. The Biot # with water is more constant and mainly dominated by fluid resistance.

One might also look at the steady state temperature distribution within the titanium. *Figure 4-37: Titanium Temperature Distribution* features the previously seen fluid temperatures and the titanium wall temperature on the hot and cold side and the midpoint temperature. Clearly, there is a much larger temperature difference between the fluids than within the titanium. However, the titanium temperature difference is neither negligible nor constant as seen in *Figure 4-38: Titanium Temperature Difference*.

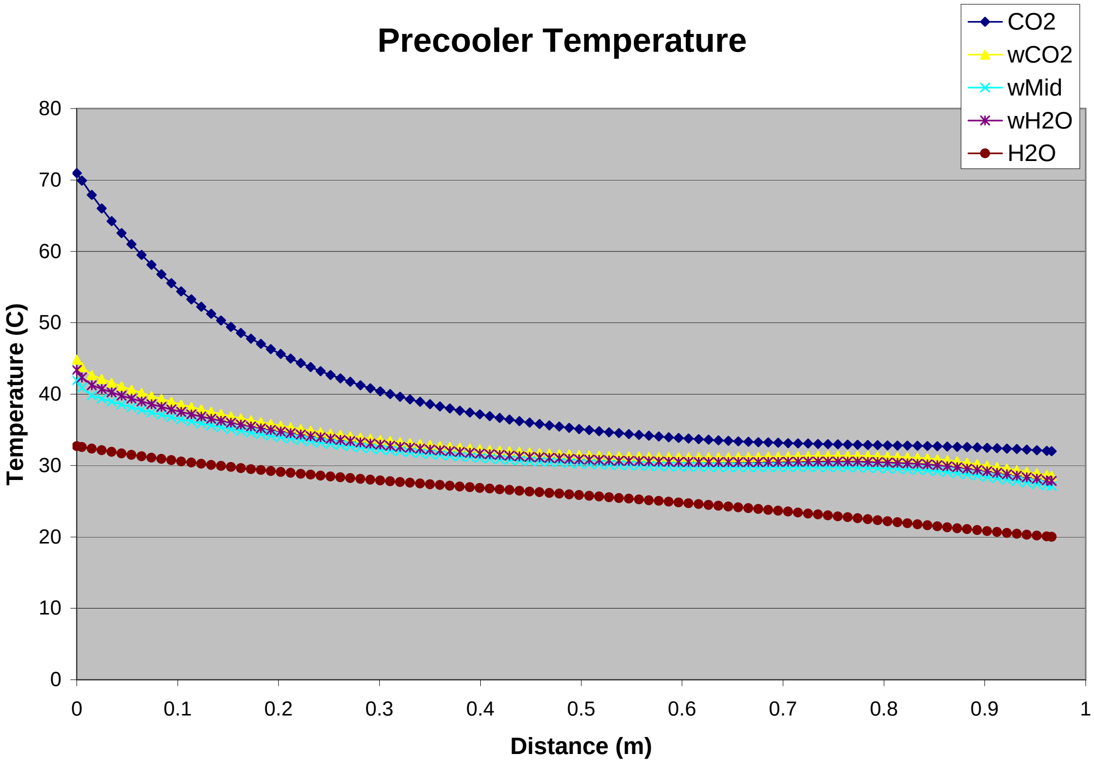


Figure 4-37: Titanium Temperature Distribution

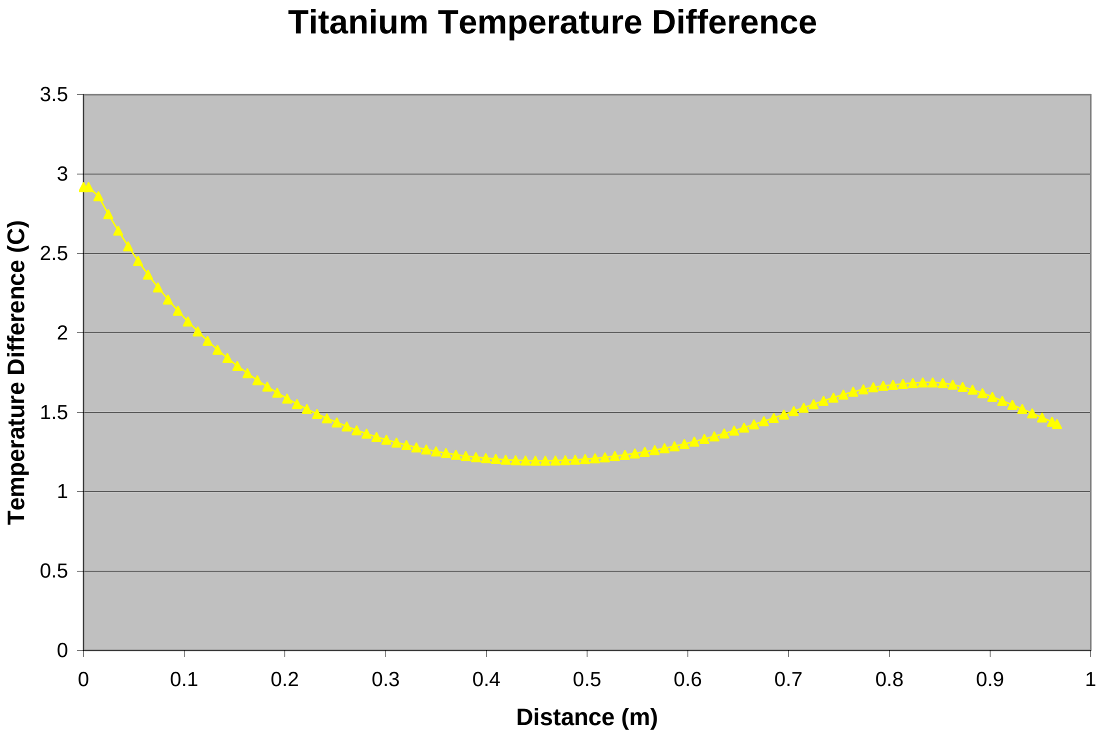


Figure 4-38: Titanium Temperature Difference

An interesting result comes from the comparing the heat capacity of the CO_2, water, and titanium as shown in ***Error! Reference source not found.***. This figure shows the amount of energy required to raise the temperature of the substance. The water has about 2/3 of the titanium's heat capacity but the CO_2 unexpectedly contains 42% more. Note that if one divides the titanium heat capacity by the total heat transferred it yields 48.7 K/sec.

Despite the higher density and greater volume in the titanium, the CO_2 heat capacity becomes so large (see ***Error! Reference source not found.***) compared to the titanium heat capacity of 522.3 [J/kg-K] that the super-critical fluid requires more energy to change temperature than the metal block surrounding it.

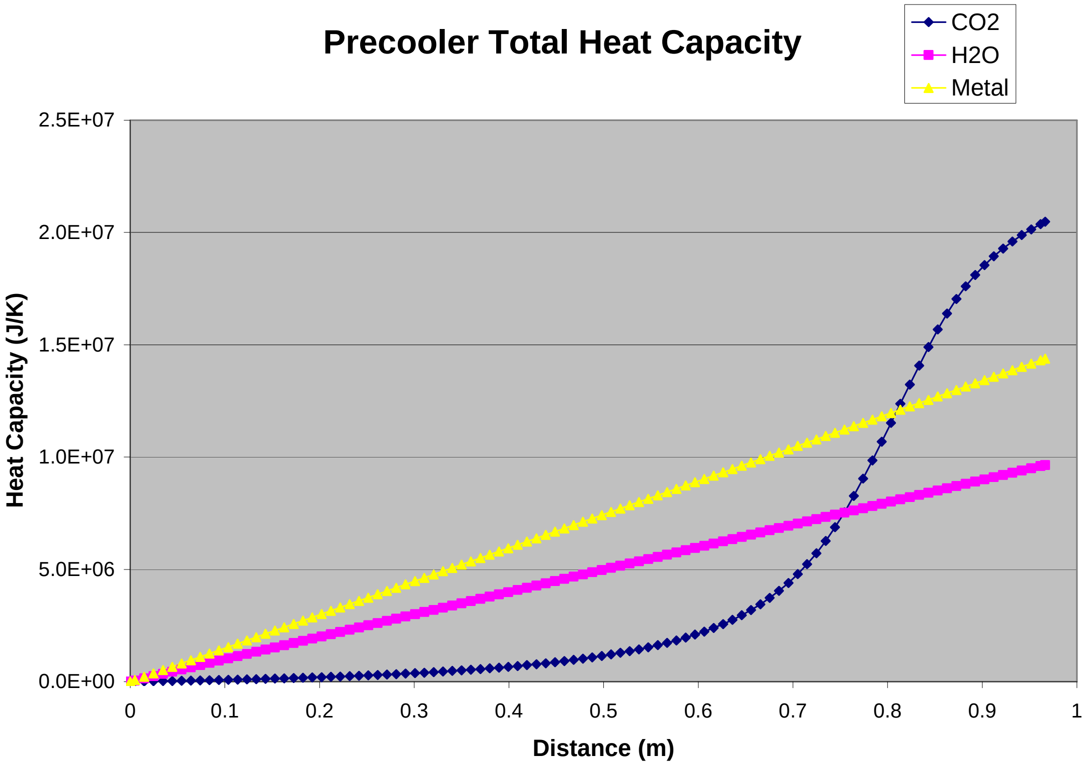


Figure 4-39: Precooler Total Heat Capacity

A significant caveat must be mentioned. This is the thermal inertia of the whole substance mass. If the inlet temperature of the CO_2 were to change by one degree it would have a much smaller effect than changing the whole fluid mass. Looking at ***Error! Reference source not found.*** in this light, one might conclude that this graph states that it requires a lot of energy to change the outlet temperature of the CO_2 significantly.

The reader may note that the Fourier number of the titanium plate between the fluid channels is 32.4*t.   This suggests that heat penetrates the titanium relatively quickly compared to its storage capacity.

4.6: PCHE Implicit Solution Method

This section will derive the system of equations used to solve a nodalized PCHE implicitly during a transient. The reader is assumed to have already read the preceding sections for background on computationally nodalizing a PCHE and the various performance models used. A diagram of the implicit computational model is shown in *Figure 4-38*. Incropera and Dewitt offer a good overview[39] of the implicit transient energy transfer process which will be used as a guideline here.

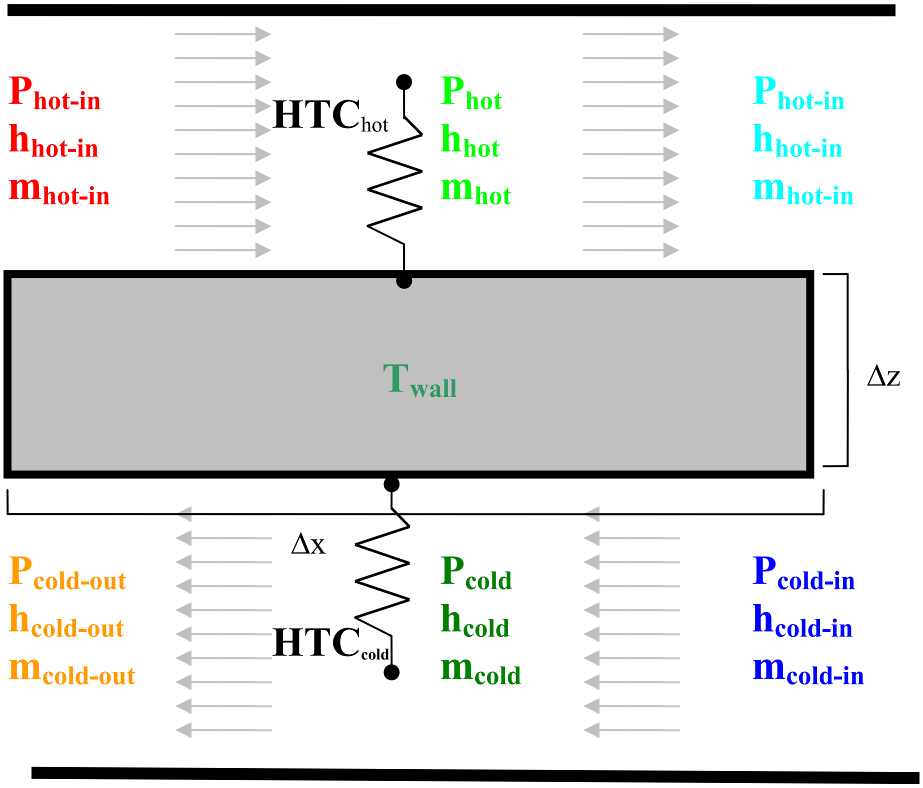


Figure 4-40: PCHE Zone Model

The assumptions of this model are the following:

- The fluids are radially uniform (i.e. temperature and pressure) and only change axially. Due to the small channel diameters (typically 0.002 m) and typically highly turbulent flows this is a reasonable assumption.
- The momentum equation may be treated quasi-statically. This analysis is not designed to capture rapid fluid momentum effects.
- The wall has no thermal resistance/is at a uniform temperature (in thickness – it varies axially) and provides no axial conduction. The thermal resistance of a thin PCHE metal divider is much less than the resistances found at the fluid boundaries therefore the temperature difference across a metal section will be relatively small. Furthermore, there is a large amount of heat being transferred

between the fluid streams therefore axial conduction should add a negligible amount of heat transfer.

- Within a zone one may perform all calculations at the average fluid properties based upon assumed linear enthalpy and pressure change. In practice, one may simply increase the number of computational nodes until this assumption is adequately satisfied.

Using these assumptions this chapter will derive a set of non-linear mass and energy conservation equations and will match fluid pressure drops to model a PCHE section.

Derivation

Conservation of mass for a control volume is:

$$\frac{dM}{dt} = \dot{m}_{in} - \dot{m}_{out}$$

The hot fluid with a constant control volume may be implicitly discretized as:

$$\frac{dM_{hot}}{dt} = \dot{m}_{hot-in} - \dot{m}_{hot-out} \rightarrow \frac{V_{hot}d(\rho_{hot})}{dt} = \dot{m}_{hot-in} - \dot{m}_{hot-out} \rightarrow \frac{V_{hot}(\rho_{hot}^{p+1} - \rho_{hot}^{p})}{\Delta t} = \dot{m}_{hot-in}^{p+1} - \dot{m}_{hot-out}^{p+1}$$

We will solve for the average mass flow rate for convenience:

$$\dot{m}_{hot}^{p+1} = \frac{\dot{m}_{hot-in}^{p+1} + \dot{m}_{hot-out}^{p+1}}{2} \rightarrow \dot{m}_{hot-out}^{p+1} = 2\dot{m}_{hot}^{p+1} - \dot{m}_{hot-in}^{p+1}$$

Substituting:

$$\frac{V_{hot}(\rho_{hot}^{p+1} - \rho_{hot}^{p})}{\Delta t} = \dot{m}_{hot-in}^{p+1} - \left(2\dot{m}_{hot}^{p+1} - \dot{m}_{hot-in}^{p+1}\right)$$

$$\left(2\dot{m}_{hot}^{p+1} - \dot{m}_{hot-in}^{p+1}\right) = \dot{m}_{hot-in}^{p+1} - \frac{V_{hot}(\rho_{hot}^{p+1} - \rho_{hot}^{p})}{\Delta t}$$

$$2\dot{m}_{hot}^{p+1} = 2\dot{m}_{hot-in}^{p+1} - \frac{V_{hot}(\rho_{hot}^{p+1} - \rho_{hot}^{p})}{\Delta t}$$

$$\dot{m}_{hot}^{p+1} = \dot{m}_{hot-in}^{p+1} - \frac{V_{hot}(\rho_{hot}^{p+1} - \rho_{hot}^{p})}{2\Delta t}$$

Equation 4-5: PCHE Hot Fluid Mass Conservation

Energy conservation for a control volume can be simply expressed as:

$$\frac{dE}{dt} = \dot{E}_{in} - \dot{E}_{out} \rightarrow \frac{d(Mu)}{dt} = \dot{E}_{in} - \dot{E}_{out}$$

When this is applied implicitly and discretely to the hot fluid:

$$\frac{M_{hot}^{p+1}U_{hot}^{p+1} - M_{hot}^{p}U_{hot}^{p}}{\Delta t} = \dot{m}_{hot-in}^{p+1}\, h_{hot-in}^{p+1} - \dot{m}_{hot-out}^{p+1}\, h_{hot-out}^{p+1} - \dot{Q}_{hot}^{p+1}$$

Note that the enthalpy terms capture the fluid flow work and that there is no shaft work in a PCHE. The hot fluid only looses heat from surface convection:

$$\dot{Q}^{p+1}_{hot} = HTC^{p+1}_{hot}A\left(T^{p+1}_{hot} - T^{p+1}_{wall}\right)$$

The fluid heat transfer coefficient and temperature is calculated from the average node enthalpy and pressure.

Assuming our control volume is constant:

$$\frac{V_{hot}\left(\rho^{p+1}_{hot}u^{p+1}_{hot} - \rho^{p}_{hot}u^{p}_{hot}\right)}{\Delta t} = \dot{m}^{p+1}_{hot-in}\, h^{p+1}_{hot-in} - \dot{m}^{p+1}_{hot-out}\, h^{p+1}_{hot-out} - HTC^{p+1}_{hot}A\left(T^{p+1}_{hot} - T^{p+1}_{wall}\right)$$

We will solve this equation for the average hot fluid enthalpy for later computational convenience:

$$h^{p+1}_{hot} = \frac{h^{p+1}_{hot-in} + h^{p+1}_{hot-out}}{2} \rightarrow h^{p+1}_{hot-out} = 2h^{p+1}_{hot} - h^{p+1}_{hot-in}$$

Substituting:

$$\frac{V_{hot}\left(\rho^{p+1}_{hot}u^{p+1}_{hot} - \rho^{p}_{hot}u^{p}_{hot}\right)}{\Delta t} = \dot{m}^{p+1}_{hot-in}\, h^{p+1}_{hot-in} - \dot{m}^{p+1}_{hot-out}\left(2h^{p+1}_{hot} - h^{p+1}_{hot-in}\right) - HTC^{p+1}_{hot}A\left(T^{p+1}_{hot} - T^{p+1}_{wall}\right)$$

$$\dot{m}^{p+1}_{hot-out}\left(2h^{p+1}_{hot} - h^{p+1}_{hot-in}\right) = \dot{m}^{p+1}_{hot-in}\, h^{p+1}_{hot-in} - HTC^{p+1}_{hot}A\left(T^{p+1}_{hot} - T^{p+1}_{wall}\right) - \frac{V_{hot}\left(\rho^{p+1}_{hot}u^{p+1}_{hot} - \rho^{p}_{hot}u^{p}_{hot}\right)}{\Delta t}$$

$$2h^{p+1}_{hot} = \frac{\dot{m}^{p+1}_{hot-in}\, h^{p+1}_{hot-in} - HTC^{p+1}_{hot}A\left(T^{p+1}_{hot} - T^{p+1}_{wall}\right) - \frac{V_{hot}\left(\rho^{p+1}_{hot}u^{p+1}_{hot} - \rho^{p}_{hot}u^{p}_{hot}\right)}{\Delta t}}{\dot{m}^{p+1}_{hot-out}} + h^{p+1}_{hot-in}$$

$$h^{p+1}_{hot} = \frac{\dot{m}^{p+1}_{hot-in}\, h^{p+1}_{hot-in} - HTC^{p+1}_{hot}A\left(T^{p+1}_{hot} - T^{p+1}_{wall}\right) - \frac{V_{hot}\left(\rho^{p+1}_{hot}u^{p+1}_{hot} - \rho^{p}_{hot}u^{p}_{hot}\right)}{\Delta t}}{2\,\dot{m}^{p+1}_{hot-out}} + \frac{h^{p+1}_{hot-in}}{2}$$

Equation 4-6: PCHE Hot Fluid Energy Conservation

The hot fluid pressure is simply the correlation predicted pressure drop for the quasi-static momentum case:

$$\Delta P = f\left(h, p\right)$$

Applying this implicitly:

$$P^{p+1}_{hot-out} = P^{p+1}_{hot-in} - \Delta P^{p+1}_{hot}$$

Once again we will solve for average pressure:

$$P^{p+1}_{hot} = \frac{P^{p+1}_{hot-in} + P^{p+1}_{hot-out}}{2}$$

Substituting:

$$P^{p+1}_{hot} = \frac{P^{p+1}_{hot-in} + \left(P^{p+1}_{hot-in} - \Delta P^{p+1}_{hot}\right)}{2}$$

Equation 4-7: PCHE Hot Fluid Pressure

Similarly, we will derive the same equations for the cold fluid. Care should be taken regarding sign conventions here. Recall that for computational efficiency all of the

PCHEs are solved from one side to the other and it was arbitrarily chosen to be the hot fluid inlet to outlet/the cold fluid outlet to inlet. Thus the cold fluid is computationally solved in reverse of the flow direction as suggested by *Figure 4-41*.

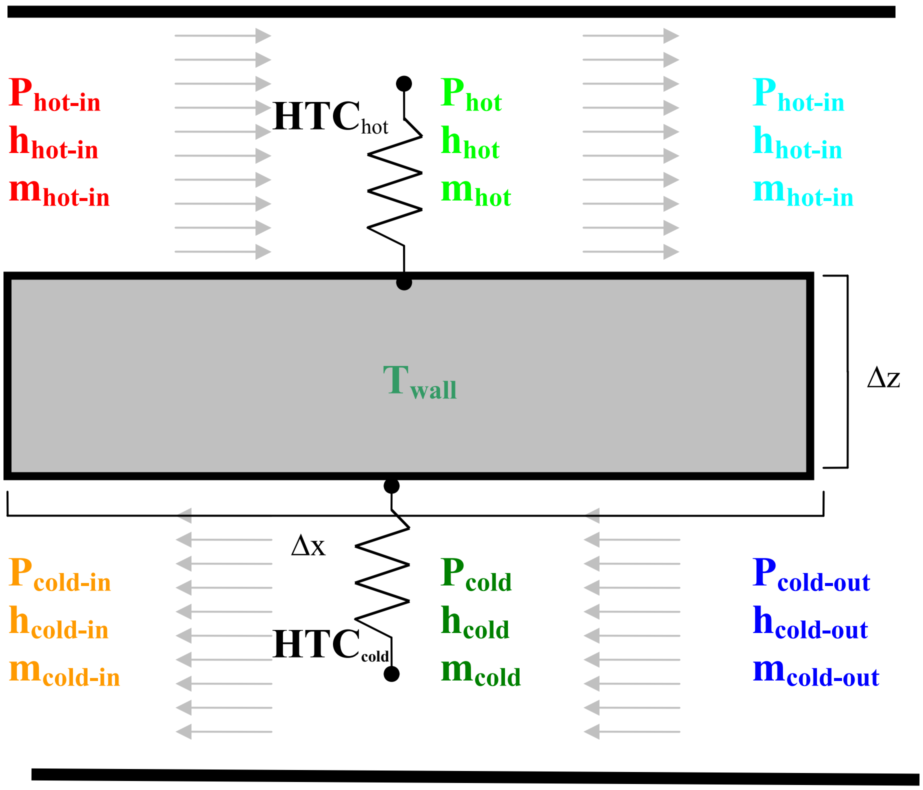


Figure 4-41: PCHE Zone Computational Model

For mass conservation of the cold fluid in a control volume with reversed flow:

$$\frac{dM_{cold}}{dt} = \dot{m}_{cold-out} - \dot{m}_{cold-in} \rightarrow \frac{V_{cold}\, d(\rho_{cold})}{dt} = \dot{m}_{cold-out} - \dot{m}_{cold-in} \rightarrow$$

$$\frac{V_{cold}(\rho_{cold}^{p+1} - \rho_{cold}^{p})}{\Delta t} = \dot{m}_{cold-out}^{p+1} - \dot{m}_{cold-in}^{p+1}$$

Again solving for the average value:

$$\dot{m}_{cold}^{p+1} = \frac{\dot{m}_{cold-in}^{p+1} + \dot{m}_{cold-out}^{p+1}}{2} \rightarrow \dot{m}_{cold-out}^{p+1} = 2\dot{m}_{cold}^{p+1} - \dot{m}_{cold-in}^{p+1}$$

Substituting:

$$\frac{V_{cold}(\rho_{cold}^{p+1} - \rho_{cold}^{p})}{\Delta t} = 2\dot{m}_{cold}^{p+1} - \dot{m}_{cold-in}^{p+1} - \dot{m}_{cold-in}^{p+1}$$

$$\frac{V_{cold}(\rho_{cold}^{p+1} - \rho_{cold}^{p})}{2\Delta t} = \dot{m}_{cold}^{p+1} - \dot{m}_{cold-in}^{p+1}$$

$$\dot{m}^{p+1}_{cold} = \dot{m}^{p+1}_{cold-in} + \frac{V_{cold}\,(\rho^{p+1}_{cold} - \rho^{p}_{cold})}{2\Delta t}$$

Equation 4-8: PCHE Computational Cold Fluid Mass Conservation

The cold fluid energy conservation equation is:

$$\frac{M^{p+1}_{cold}u^{p+1}_{cold} - M^{p}_{cold}u^{p}_{cold}}{\Delta t} = \dot{m}^{p+1}_{cold-out}\, h^{p+1}_{cold-out} + \dot{Q}^{p+1}_{cold} - \dot{m}^{p+1}_{cold-in}\, h^{p+1}_{cold-in}$$

This equation states that:

- as hotter and more fluid are found in the “*in*” location then more energy is required from the control volume
- as more heat flows from the structure to the cold fluid then the “*in*” location must increase in energy

This preserves the correct directionality for energy conservation.

Note that the enthalpy terms capture the fluid flow work and that there is no shaft work in a cold fluid of a PCHE. The cold fluid only looses heat from surface convection:

$$\dot{Q}^{p+1}_{cold} = HTC^{p+1}_{cold}A\left(T^{p+1}_{wall} - T^{p+1}_{cold}\right)$$

The fluid heat transfer coefficient and temperature is calculated from the average node enthalpy and pressure.

Once again we will solve for the average cold fluid enthalpy in the zone for computational convenience:

$$h^{p+1}_{cold} = \frac{h^{p+1}_{cold-in} + h^{p+1}_{cold-out}}{2} \rightarrow h^{p+1}_{cold-out} = 2h^{p+1}_{cold} - h^{p+1}_{cold-in}$$

Substituting:

$$\frac{V_{cold}\left(\rho^{p+1}_{cold}u^{p+1}_{cold} - \rho^{p}_{cold}u^{p}_{cold}\right)}{\Delta t} = \dot{m}^{p+1}_{cold-out}\left(2h^{p+1}_{cold} - h^{p+1}_{cold-in}\right) + HTC^{p+1}_{cold}A\left(T^{p+1}_{wall} - T^{p+1}_{cold}\right) - \dot{m}^{p+1}_{cold-in}\, h^{p+1}_{cold-in}$$

$$\dot{m}^{p+1}_{cold-out}\left(2h^{p+1}_{cold} - h^{p+1}_{cold-in}\right) = \dot{m}^{p+1}_{cold-in}\, h^{p+1}_{cold-in} - HTC^{p+1}_{cold}A\left(T^{p+1}_{wall} - T^{p+1}_{cold}\right) + \frac{V_{cold}\left(\rho^{p+1}_{cold}u^{p+1}_{cold} - \rho^{p}_{cold}u^{p}_{cold}\right)}{\Delta t}$$

$$2h^{p+1}_{cold} = \frac{\dot{m}^{p+1}_{cold-in}\, h^{p+1}_{cold-in} - HTC^{p+1}_{cold}A\left(T^{p+1}_{wall} - T^{p+1}_{cold}\right) + \dfrac{V_{cold}\left(\rho^{p+1}_{cold}u^{p+1}_{cold} - \rho^{p}_{cold}u^{p}_{cold}\right)}{\Delta t}}{\dot{m}^{p+1}_{cold-out}} + h^{p+1}_{cold-in}$$

$$h^{p+1}_{cold} = \frac{\dot{m}^{p+1}_{cold-in}\, h^{p+1}_{cold-in} - HTC^{p+1}_{cold}A\left(T^{p+1}_{wall} - T^{p+1}_{cold}\right) + \dfrac{V_{cold}\left(\rho^{p+1}_{cold}u^{p+1}_{cold} - \rho^{p}_{cold}u^{p}_{cold}\right)}{\Delta t}}{2\,\dot{m}^{p+1}_{cold-out}} + 2h^{p+1}_{cold-in}$$

Equation 4-9: PCHE Computational Cold Fluid Energy Conservation

The cold fluid pressure is simply the correlation predicted pressure drop for the quasi-static momentum case:

$$\Delta P = f(h, p)$$

Applying this implicitly (to a reverse flow control volume):

$$P_{cold-out}^{p+1} = P_{cold-in}^{p+1} + \Delta P_{cold}^{p+1}$$

Once again we will solve for average pressure:

$$P_{cold}^{p+1} = \frac{P_{cold-in}^{p+1} + P_{cold-out}^{p+1}}{2}$$

Substituting:

$$P_{cold}^{p+1} = \frac{P_{cold-in}^{p+1} + \left(P_{cold-in}^{p+1} + \Delta P_{cold}^{p+1}\right)}{2}$$

Equation 4-10: PCHE Cold Fluid Pressure

Finally, we need only solve for the metal wall energy conservation. Once again we implicitly discretize the energy conservation equation:

$$\frac{dE}{dt} = \dot{E}_{in} - \dot{E}_{out} \rightarrow \frac{d(Mu)}{dt} = \dot{E}_{in} - \dot{E}_{out} \rightarrow \frac{M_{wall}(u_{wall}^{p+1} - u_{wall}^{p})}{\Delta t} = \dot{E}_{in} - \dot{E}_{out}$$

The only energy transfer is from the fluid surface convection:

$$\frac{M_{wall}(u_{wall}^{p+1} - u_{wall}^{p})}{\Delta t} = \dot{Q}_{hot} - \dot{Q}_{cold}$$

$$\frac{M_{wall}(u_{wall}^{p+1} - u_{wall}^{p})}{\Delta t} = \dot{Q}_{hot} - \dot{Q}_{cold}$$

For a metal the specific heat may be assumed constant:

$$\frac{M_{wall}C_{wall}(T_{wall}^{p+1} - T_{wall}^{p})}{\Delta t} = HTC_{hot}^{p+1}A\left(T_{hot}^{p+1} - T_{wall}^{p+1}\right) - HTC_{cold}^{p+1}A\left(T_{wall}^{p+1} - T_{cold}^{p+1}\right)$$

Once again we will solve for an average value:

$$T_{wall}^{p+1} = \frac{T_{wall}^{p+1} + T_{wall}^{p+1}}{2} \rightarrow T_{wall}^{p+1} = 2T_{wall}^{p+1} - T_{wall}^{p+1}$$

Substituting:

$$\frac{M_{wall}C_{wall}(T_{wall}^{p+1} - T_{wall}^{p})}{\Delta t} = HTC_{hot}^{p+1}A\left(T_{hot}^{p+1} - T_{wall}^{p+1}\right) - HTC_{cold}^{p+1}A\left(T_{wall}^{p+1} - T_{cold}^{p+1}\right)$$

$$\frac{M_{wall}C_{wall}}{\Delta t}T_{wall}^{p+1} - \frac{M_{wall}C_{wall}}{\Delta t}T_{wall}^{p} = HTC_{hot}^{p+1}AT_{hot}^{p+1} - HTC_{hot}^{p+1}AT_{wall}^{p+1} - HTC_{cold}^{p+1}AT_{wall}^{p+1} + HTC_{cold}^{p+1}AT_{cold}^{p+1}$$

$$\frac{M_{wall}C_{wall}}{\Delta t}T_{wall}^{p+1} + HTC_{hot}^{p+1}AT_{wall}^{p+1} + HTC_{cold}^{p+1}AT_{wall}^{p+1} = HTC_{hot}^{p+1}AT_{hot}^{p+1} + HTC_{cold}^{p+1}AT_{cold}^{p+1} + \frac{M_{wall}C_{wall}}{\Delta t}T_{wall}^{p}$$

$$T_{wall}^{p+1}\left(\frac{M_{wall}C_{wall}}{\Delta t} + HTC_{hot}^{p+1}A + HTC_{cold}^{p+1}A\right) = HTC_{hot}^{p+1}AT_{hot}^{p+1} + HTC_{cold}^{p+1}AT_{cold}^{p+1} + \frac{M_{wall}C_{wall}}{\Delta t}T_{wall}^{p}$$

$$T_{wall}^{p+1} = \frac{HTC_{hot}^{p+1} AT_{hot}^{p+1} + HTC_{cold}^{p+1} AT_{cold}^{p+1} + \frac{M_{wall} C_{wall}}{\Delta t} T_{wall}^{p}}{HTC_{hot}^{p+1} A + HTC_{cold}^{p+1} A + \frac{M_{wall} C_{wall}}{\Delta t}}$$

Equation 4-11: PCHE Wall Energy Conservation

We now have seven equations and seven unknowns which must be simultaneously solved to arrive at a new time steps solution. Unfortunately, the common practice of putting these equations into a tridiagonal matrix and solving for the various values is difficult due to the nature of this calculation. Both fluid's properties (i.e. density and internal energy) and performance models (i.e. heat transfer coefficient and friction factor) are calculated based upon the node's average enthalpy and pressure. The equation of state for carbon dioxide is quite complex but even if one were able to express the various fluid properties succinctly enough to put into a matrix one would still have the problem of updating the performance models. Therefore, these equations will be solved by a general non-linear numerical solver.

At a new time step the solver will be supplied with the initial guesses of the seven variables to be solved for: $X_{guess} = (\dot{m}_{hot}^{p+1}, h_{hot}^{p+1}, P_{hot}^{p+1}, \dot{m}_{cold}^{p+1}, h_{cold}^{p+1}, P_{cold}^{p+1}, T_{wall}^{p+1})$ at the node's average values. The solver will then update the various fluid properties based upon each fluid's average enthalpy and pressure and then update the pressure drop and heat transfer models using these new values.

The newly calculated fluid properties and performance model results are then plugged into *Equation 4-6* through Equation 4-11 along with the known incoming fluid property values: $X_{in} = (\dot{m}_{hot-in}^{p+1}, h_{hot-in}^{p+1}, P_{hot-in}^{p+1}, \dot{m}_{cold-in}^{p+1}, h_{cold-in}^{p+1}, P_{cold-in}^{p+1})$, which are the outputs of previous calculations, and the previous time step's known values to predict the current time step's average fluid values: $X_{calculated} = (\dot{m}_{hot}^{p+1}, h_{hot}^{p+1}, P_{hot}^{p+1}, \dot{m}_{cold}^{p+1}, h_{cold}^{p+1}, P_{cold}^{p+1}, T_{wall}^{p+1})$. By iteratively refining the guessed values until they match the calculated values to tolerance the PCHE node may be solved at the current time step. Using the converged variables the code then calculates the fluid outlet values: $X_{out} = (\dot{m}_{hot-out}^{p+1}, h_{hot-out}^{p+1}, P_{hot-out}^{p+1}, \dot{m}_{cold-out}^{p+1}, h_{cold-out}^{p+1}, P_{cold-out}^{p+1})$ which become the next calculations inputs. Since each calculation only depends upon the previous calculation's results we maintain our implicit assumption.

4.7: PCHE Modeling Summary

Printed circuit heat exchangers are a key component for the highly recuperated S-CO_2 recompression cycle. Although there is considerable uncertainty about the selection of appropriate heat transfer and pressure drop correlations it is judged that current GAS-PASS/CO_2 correlations are conservative.

Due to the non-linear property changes of S-CO2 simple analytical correlations are not available to model heat exchanger performance. However, PCHEs can be efficiently computed by axially nodalizing a single channel, computing with enthalpy at the midpoint of a node, using past solutions as initial guesses, and avoiding excess computational loops when PCHEs are computed within a general non-linear solver. Efficient computation allows realistic transient simulation where a small non-linear set of equations must be solved within each node of every PCHE. Taken together, GAS-PASS/CO_2 offers a fast and accurate method to compute heat exchange and pressure drop in this cycle.

5 The GAS-PASS/CO_2 Code

5.1 Introduction

GAS-PASS/CO_2 has been significantly modified and improved as briefly documented in this chapter. The code is now fully functional and can successfully simulate the S-CO_2 recompression cycle. It is still a work in progress but future changes will stem from simulation results and new control capabilities instead of fundamental code development.

This chapter contains six sections:

1. The introduction
2. A brief description of the current code.
3. A review of the unchanged GAS-PASS solution process.
4. An explanation and analysis of the new component conservation equations.
5. A description of the techniques used to make the code robust.
6. A summary of the new code.

5.2 GAS-PASS/CO_2

GAS-PASS/CO_2 represents a comprehensive update to GAS-PASS/H. It retains the original code structure and solution process but has been rewritten on a line by line basis where original code was retained and the majority of the current code was written from scratch. The current code is 20,000 lines of Fortran 90[i] in 29 source files.

GAS-PASS/CO_2 now includes all real fluid properties, detailed turbomachinery performance interpolation with correction for fluid property effects, and detailed PCHE heat transfer and pressure drop calculations.

While tailored to the S-CO_2 cycle the code is modular and general, allowing the end user to change almost any part of the plant simply by modifying simple text input files. New capabilities (like modeling the physics of hydrogen electrolysis) will, of course, require modifying the source code but the end user could easily change the current code to something quite different without coding. For example, the current model could be converted to a helium plant, heated by lead bismuth, with a radial turbine, and axial-radial hybrid compressors without editing the source. The major limitation to creating new plant designs is creating and combining new component input files.

Due to the general nature of the fluid properties code any fluid that can be pre-computed and put in tabular format can be used. Due to the general nature of the turbomachinery

[i] Gas-Pass is programmed and developed with Compaq Visual Fortran 6.6.C. Older Compaq Visual Fortran compilers will not work with some new features.

methods, any design that offers sets of shafts speed curves can be modeled. Due to the general nature of the heat transfer code any type of PCHE can be modeled for any set of fluids (assuming the current heat transfer and pressure drop correlations may be used). These capabilities should place GAS-PASS/CO_2 in an excellent position for future use.

5.2.1 Input/Output

Gas-Pass takes 9 input files, 23 property tables, and produces 4 output files by default. The only command line argument GAS-PASS/CO_2 takes is the name of the input file which lists all of these other files. *Table 5-1* provides a brief description of the input and output files.

The input file is separated into four sections:

1. The first section contains the main GAS-PASS/CO_2 input file and the four outputs files to be generated.
2. The second section contains the geometry and characteristics. Each PCHE has a file.
3. The third section contains the performance curves for the turbomachinery. Each turbomachine has one file.
4. The final section contains the property table files for all three fluids: the cycle working fluid, the heat removal/cooling external fluid in the tertiary circuit, and the heat addition/heating external fluid in the primary circuit.

Input Order	**Purpose**	**Description**
1	IO	The main Gas-Pass input file.
2	IO	The main Gas-Pass output file containing the simulation results.
3	IO	An output file listing boundary conditions and solved for variables.
4	IO	An output file that creates a new main Gas-Pass input file with converged steady state solutions from the current simulation.
5	IO	A run log containing all screen output and warning messages.
6	PCHE	The precooler specifications.
7	PCHE	The low temperature recuperator specifications.
8	PCHE	The high temperature recuperator specifications.
9	PCHE	The intermediate heat exchanger specifications.
10	TURBO	The main compressor performance curves.
11	TURBO	The recompressing compressor performance curves.
12	TURBO	The turbine performance curves.
13	PROP	The working fluid enthalpy and pressure to isobaric specific heat table.
14	PROP	The working fluid enthalpy and pressure to density table.
15	PROP	The working fluid enthalpy and pressure to internal energy table.
16	PROP	The working fluid enthalpy and pressure to thermal conductivity table.
17	PROP	The working fluid enthalpy and pressure to temperature table.

18	PROP	The working fluid enthalpy and pressure to viscosity table.
19	PROP	The working fluid temperature and pressure to enthalpy table.
20	PROP	The working fluid enthalpy and pressure to entropy table.
21	PROP	The working fluid entropy and pressure to enthalpy table.
22	PROP	The cooling fluid enthalpy and pressure to isobaric specific heat table.
23	PROP	The cooling fluid enthalpy and pressure to density table.
24	PROP	The cooling fluid enthalpy and pressure to internal energy table.
25	PROP	The cooling fluid enthalpy and pressure to thermal conductivity table.
26	PROP	The cooling fluid enthalpy and pressure to temperature table.
27	PROP	The cooling fluid enthalpy and pressure to viscosity table.
28	PROP	The cooling fluid temperature and pressure to enthalpy table.
29	PROP	The heating fluid enthalpy and pressure to isobaric specific heat table.
30	PROP	The heating fluid enthalpy and pressure to density table.
31	PROP	The heating fluid enthalpy and pressure to internal energy table.
32	PROP	The heating fluid enthalpy and pressure to thermal conductivity table.
33	PROP	The heating fluid enthalpy and pressure to temperature table.
34	PROP	The heating fluid enthalpy and pressure to viscosity table.
35	PROP	The heating fluid temperature and pressure to enthalpy table.

Table 5-1: Gas-Pass Input and Output Files

Generally, the main Gas-Pass input file will be the only file that the user will want to edit and all the other files need to be set up only once. The reader should note that the fluid property tables are a repetitive set of tables for the various fluids except that the working fluid has two extra fluid tables which are used to perform turbomachinery performance calculations.

5.2.2 Source Code

The source code has changed significantly. It is grouped by function and includes the new sections of properties, turbomachinery, and PCHE simulation. Commonly used code has been grouped into libraries and the general input/output routines have been grouped as well. These source files are briefly explained in *Table 5-2*.

Name	**Folder**	**Function**
Main.f90		To control the flow of the program
Imsl.f90		To provide the exact interface expected by IMSL solver routines.
Memory.f90		To hold variables in memory that are not directly passed between subroutines. Includes property tables, turbomachinery curves, & PCHE zone data.
Common.f90	Library	To provide commonly used general routines
Engineering.f90	Library	To provide commonly used engineering routines.
Gp.f90	Library	GAS-PASS specific general routines steady state to

		variable vector passing.
Math.f90	Library	To provide commonly used mathematical routines.
System.f90	Library	To provide commonly used routines that interface with the operating system.
Files.f90	IO	To read in the names and locations of all the files used with GAS-PASS
Input.f90	IO	To read in and process main GAS-PASS input file.
ssOutPout.f90	IO	To output detailed variable list and to create a new GAS-PASS input file based upon converged results.
Props.f90	Properties	To handle all property related routines including able read in & interpolation (linear & log).
Turbodata.f90	Turbomap	To read in and process turbomachinery characteristics & performance curves.
Turbointerpoalte.f90	Turbomap	To interpolate along & between shaft speed curves.
Turbolibrary.f90	Turbomap	To provide commonly used turbomachinery routines.
Pche.f90	PCHE	To calculate steady state PCHE performance using Newton's method.
Pche_tr.f90	PCHE	To calculate transient PCHE performance using non-linear equation solver.
Pcheinitialize.f90	PCHE	To initialize a PCHE's zone data with good guesses.
Pcheinput.f90	PCHE	To read in & process a PCHEs characteristics.
Pchestruct.f90	PCHE	Provide custom memory structures for zone data and PCHE characteristics.
Pchethermo.f90	PCHE	To hold PCHE heat transfer and pressure drop correlations.
Zones.f90	PCHE	Calculate a zone's heat transfer, pressure drop, mass inventory, and energy inventory. Used in both steady state and transient calculations.
Components.f90	SS	Provide the solution equations and model calls for steady state components.
Fcn.f90	SS	Supervise the execution of one plant evaluation given variable inputs.
SS.f90	SS	Supervise calculations of steady state solution.
Components_tr.f90	TR	Provide the equations and model calls for transient components.
Fcn_tr.f90	TR	Supervise the execution of one transient plant evaluation given variable inputs.
TR.f90	TR	Supervise the calculations of the transient solution and outputting data.

Table 5-2: Gas-Pass Source Files

5.3 GAS-PASS Solution Process

GAS-PASS/CO_2 uses a general Newtonian root-finding algorithm (a modified Powell-Hybrid method[40]) designed for non-linear systems of equations to compute all variable values implicitly. The user specifies which inputs are variables, boundary conditions, and constants within the input deck. Once basic checks are preformed the solution process begins with a call to the numerical solver. Note that a discussion of the various non-linear solution techniques is included in *Appendix B:* Roots of Nonlinear Systems, but the general technique is unchanged from GAS-PASS.

The numerical solver then calls a custom made function, *Fcn(x)*, which contains all of the equations necessary to solve the system at a time step. *Fcn* is organized by code modules which correspond to components within the system. Each component will contain one or more equations in the form:

$F(i) = f_i(x)$

Which must equal 0 to satisfy the system. Typically these equations are conservation laws or the output variables of other subprograms which apply conservation laws. In any case, these equations take the *x* inputs supplied by the numerical solver and evaluate the given equation. The solver then compares the error of the solution to the change in the *x* inputs and modifies the variables accordingly until tolerance is reached. A flowchart of the overall code solution process is shown in *Figure 5-2*.

The numerical solution process is fully implicit with interactive refinement of the user specified variables at each time step until the system of equations is within tolerance of the system root as outlined below:

1. At time $t_i > t_{i-1}$
2. Take X_i inputs from X_{i-1} (temperature, work …)
3. Calculate Jacobian (see *Figure 5-1*)
4. Use Jacobian to guess new X_i
5. Evaluate $F(X_i)$
6. Repeat (4), (5) to find system solution

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}$$

Figure 5-1: Jacobian

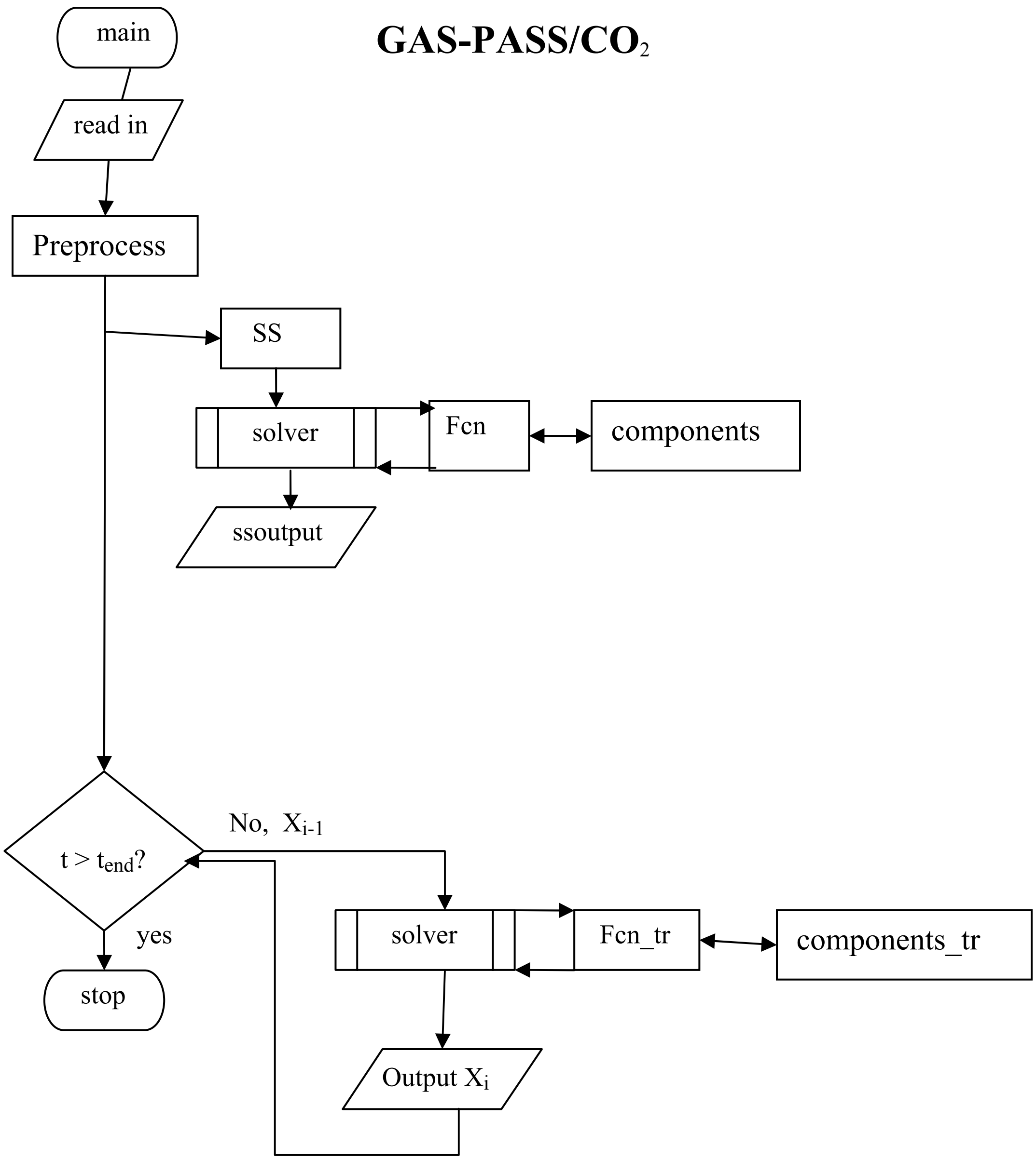


Figure 5-2: GAS-PASS/CO2 Flowchart

5.4 Component Conservation Equations

This section discusses the conservation equations Gas-Pass solves in the steady state and transient modes. The actual equations are lengthy and may be found in *Appendix A:* GAS-PASS/CO2 Equations. The section is in two parts: a general discussion of the equations and their derivation, and a discussion of avoiding mass equation linear dependence.

5.4.1 Equation Models and Assumptions

GAS-PASS/CO_2 equations are grouped by component and follow from simple conservation laws, or are set to match more complex modeling calculations. The equations assume lumped regions in space, perfect mixing, and that conditions at the outlet are the same as in the volume interior. All equations refer to the plant network layout shown in *Figure 5-3*.

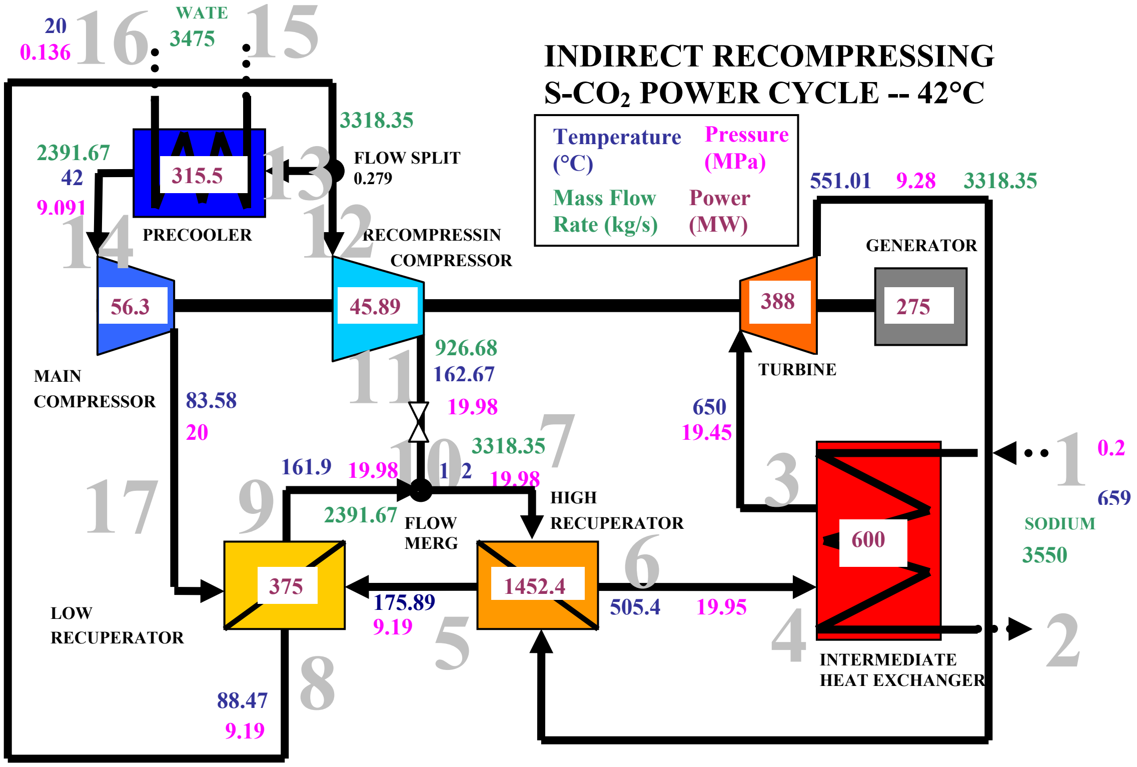


Figure 5-3: Network Plant Diagram for the S-CO_2 Recompression cycle[41]

The Gas-Pass conservation equations are formulated based upon treating a component as a black box. Gas-Pass provides a component with guesses of its inputs and outputs which the component module uses to do its calculations and the results are plugged into conservation equations. The numerical solver then takes care of changing the input and output guesses to balance the conservation equations. Generally one component's outputs become the next component's inputs.

Although there are a number of ways to derive the conservation equations it is only necessary that the derivation process be consistent. An easy way to determine the number of equations is to treat a component as a black box with a set of inputs that produce a set of outputs. If one assumes that the incoming fluid is determined then setting the number of equations can be boiled down to two simple ideas:

1. A fluid flow stream has three properties which completely define it: mass flow rate, specific enthalpy, and pressure.
2. What happens within a component is irrelevant except when energy is transferred:
    - If the energy transfer is *external* to the system it is a boundary condition.
    - If the energy is transferred to or from another component *within* the system then there must be a conservation of energy equation for this process.

In practice the first rule dictates that for each fluid coming out of a component there are three equations. The second rule is a little more involved, thus several examples will be given:

- A reactor has only three equations because the heat that is added is not from another component – it is a boundary condition.
- A heat exchanger has only six equations (three for each outgoing stream) because the heat transfer occurs within the component.
- A turbomachine has four equations because work is added or removed from the shaft, which affects other components in the system.
- A generator has one equation because there is no fluid flow but there is work added and removed from other system components via the shaft.

The fundamental conservation equations are quite simple as shown in *Equation 5-1* through *Equation 5-2*. *Equation 5-3* is included solely to indicate that the momentum contains a significant assumption.

The momentum equation is reduced to quasi-steady state pressure drop calculations. This allows a significant simplification of the solution with a relatively small loss of accuracy on flow dynamics since S-CO$_2$ is almost incompressible. The application of these equations in the steady state and in transient processes is elaborated upon in *Appendix A:* GAS-PASS/CO2 Equations.

Equation 5-1: Mass Conservation Equation

$$\frac{\partial M}{\partial t} = \dot{m}_{in} - \dot{m}_{out}$$

Equation 5-2: Energy Conservation

$$\frac{dE}{dt} = \dot{E}in - \dot{E}out$$

Equation 5-3: Quasi-Static Momentum Equation -- Pressure Drop

$$dP^{time} = \Delta P^{time}_{calculated}$$

5.4.2 Preventing Mass Equation Linear Dependence

Setting up a closed system of components (such that there is no mass (typically) leaving the system) requires care that the solution matrix does not become linearly dependent in the mass equations. When setting up component conservation equations the mass conservation equation is typically the most straightforward. However, if the system loop is closed then this system of equations will become linearly dependent as shown in the following examples.

5.4.2.1 Linear Mass Dependence Example – Steady State

Figure 5-4 shows a very simple closed system where component **A** feeds directly into component **B** which feeds directly back into component **A**. There is no mass storage in the pipes (the connectors between components), no mass can escape the system, and the volumes of the components do not change.

Figure 5-4: Component Mass Equations

The mass conservation equation for a control volume is:

$$\frac{dM}{dt} = \dot{m}_{in} - \dot{m}_{out} \rightarrow \dot{m}_{in} - \dot{m}_{out} = \dot{M}$$

Thus setting up a system of mass conservation equations we have:

$$1 * \dot{m}_{in}^{A} - 1 * \dot{m}_{out}^{A} = \dot{M}^{A}$$

$$1 * \dot{m}_{in}^{B} - 1 * \dot{m}_{out}^{B} = \dot{M}^{B}$$

If we setup this system for the steady state (no mass storage) then we have:

$$1 * \dot{m}_{in}^{A} - 1 * \dot{m}_{out}^{A} = 0$$

$$1 * \dot{m}_{in}^{B} - 1 * \dot{m}_{out}^{B} = 0$$

By definition we know that what goes into a pipe must come out therefore:

$$\dot{m}^{A}_{out} = \dot{m}^{B}_{in}$$
$$\dot{m}^{B}_{out} = \dot{m}^{A}_{in}$$

Thus if we substitute the **A** variables for **B** variables our system becomes:

$$1 * \dot{m}^{A}_{in} - 1 * \dot{m}^{A}_{out} = 0$$
$$1 * \dot{m}^{A}_{out} - 1 * \dot{m}^{A}_{in} = 0$$

Which in matrix form is:

$\dot{m}^{A}_{in}$	$\dot{m}^{A}_{out}$	$\dot{M}$
1	-1	0
-1	1	0

Alternatively both equations can be rearranged to yield:

$$\dot{m}^{A}_{in} = \dot{m}^{A}_{out}$$
$$\dot{m}^{A}_{in} = \dot{m}^{A}_{out}$$

From either perspective it is apparent that we have two equations but only one unknown which is an over-determined or linearly-dependent system of equations. Therefore, any answer will satisfy these equations and our simulation will appear to work quite robustly regardless of the behavior of the closed loop mass equations i.e. we are not simulating a real/physical system.

5.4.2.2 Linear Mass Dependence Example – Transient

It is worth showing that this problem applies to the transient case as well. Once again we will use *Figure 5-4* which shows a very simple closed system where component **A** feeds directly into component **B** which feeds directly back into component **A**.

The transient mass equations are:

$$1 * \dot{m}^{A}_{in} - 1 * \dot{m}^{A}_{out} = \dot{M}^{A}$$
$$1 * \dot{m}^{B}_{in} - 1 * \dot{m}^{B}_{out} = \dot{M}^{B}$$

Even though there can be storage in a component we know that, by definition, what goes into a pipe must come out therefore:

$$\dot{m}^{A}_{out} = \dot{m}^{B}_{in}$$
$$\dot{m}^{B}_{out} = \dot{m}^{A}_{in}$$

Thus if we substitute the **A** variables for **B** variables our system becomes:

$$1*\dot{m}_{in}^{A}-1*\dot{m}_{out}^{A}=\dot{M}^{A}$$

$$1*\dot{m}_{out}^{A}-1*\dot{m}_{in}^{A}=\dot{M}^{B}$$

Which in matrix form is:

$\dot{m}_{in}^{A}$	$\dot{m}_{out}^{A}$	$\dot{M}$
1	-1	$1\ \dot{M}^{A}$
-1	1	$1\ \dot{M}^{B}$

Alternatively both equations can be substituted for each other:

$1*\dot{m}_{in}^{A}=\dot{M}^{A}+1*\dot{m}_{out}^{A}$ -> substituting for $\dot{m}_{in}^{A}$ in the **B** equation:

$$1*\dot{m}_{out}^{A}-\left(\dot{M}^{A}+1*\dot{m}_{out}^{A}\right)=\dot{M}^{B} \text{ ->}$$

$$-\dot{M}^{A}=\dot{M}^{B}$$

From either perspective this result makes common sense -- if component **A** stores more mass then it has to come from component **B** and vice-versa. However, this still means we have two identical equations. If we substitute only the **A** values into the **B** equations we arrive at:

$$1*\dot{m}_{in}^{A}-1*\dot{m}_{out}^{A}=\dot{M}^{A}$$

$$1*\dot{m}_{out}^{A}-1*\dot{m}_{in}^{A}=-\dot{M}^{A}$$

Or:

$\dot{m}_{in}^{A}$	$\dot{m}_{out}^{A}$	$\dot{M}$
1	-1	$\dot{M}^{A}$
-1	1	$-\dot{M}^{A}$

Therefore we still have an over-determined system with two identical equations where any $\dot{m}_{in}^{A}$ will work as long as $\dot{m}_{out}^{A}$ is $\dot{M}^{A}$ smaller. The situation is the same as in the steady state example.

5.4.2.3 Avoiding Linear Mass Dependence

We have now seen that for a two component closed loop, setting up mass conservation in both components creates a linearly dependent system. What is true for two components is true for any number, therefore this counterintuitive result should be clearly stated:

> ***in a closed loop system setting up mass conservation equations for each component leads to simulation failure***.

Fundamentally, what has been lacking thus far is the amount of fluid in the system. The equations so far have described rates of change without mentioning the amount, therefore they are conceptually as well as mathematically incomplete.

To create linear independence and fully describe the system we can simply remove one component's mass conservation equation and replace it with a system fluid storage/inventory equation. This will be made clear with an example.

If we again reference the system in *Figure 5-4* we can setup the mass equations as follows:

$$1*\dot{m}_{in}^{A}-1*\dot{m}_{out}^{A}=\dot{M}^{A}$$

$$\sum_{i=1}^{n}M^{i}=M^{Total}$$

The second equation simply states that the total fluid in the system is equal to the sum of the fluid masses in individual components. This equation becomes slightly more sophisticated if we recognize that the mass in each component can change with time but the system inventory does not:

$$\sum_{i=1}^{n}M(t)^{i}=M^{Total}$$

The amount of fluid in a component may be calculated from (assuming a fixed control volume):

$$M(t)=\rho(t)*Vol$$

Therefore:

$$\sum_{i=1}^{n}M(t)^{i}=M^{Total}->\sum_{i=1}^{n}\rho(t)^{i}Vol^{i}=M^{Total}$$

Similarly we may express the rate of mass storage change in a component:

$$\frac{dM}{dt}=\dot{M}=\frac{d\rho}{dt}*Vol$$

Putting this together we now have:

$$1*\dot{m}_{in}^{A}-1*\dot{m}_{out}^{A}=\frac{d\rho^{A}}{dt}*Vol^{A}$$

$$\sum_{i=1}^{n} \rho(t)^{i} Vol^{i} = M^{Total}$$

Since there are only two components **A** and **B**:

$$1*\dot{m}_{in}^{A}-1*\dot{m}_{out}^{A}=\frac{d\rho^{A}}{dt}*Vol^{A}$$

$$\rho(t)^{A}Vol^{A}+\rho(t)^{B}Vol^{B}=M^{Total}$$

By now it should be clear that these conservation equations are not linearly dependent. However, it bears further clarification that this still allows us to properly simulate a real system. If one takes a time derivative of the second equation:

$$\frac{d\rho^{A}}{dt}Vol^{A}+\frac{d\rho^{B}}{dt}Vol^{B}=0\rightarrow-\frac{d\rho^{A}}{dt}Vol^{A}=\frac{d\rho^{B}}{dt}Vol^{B}\rightarrow-\dot{M}^{A}=\dot{M}^{B}$$

We find the same result that we previously achieved with the mass conservation equations. Therefore, the conservation of system inventory equation we have added is simply the *integral of the mass flow rate equations with time*. Conceptually this means that the first equation contains time varying mass information and the second equation contains time integrated (not invariant) mass information. This suggests that these equations completely describe system mass while managing to avoid linear dependence.

Looking at the new equations from a physical perspective may make this clear.

$$1*\dot{m}_{in}^{A}-1*\dot{m}_{out}^{A}=\dot{M}^{A}$$

$$\sum_{i=1}^{n} M^{i} = M^{Total}$$

Looking at *Figure 5-4* we can readily deduce that if we know the mass flow rates coming in and out of component **A** then we know every mass flow rate in the system (since the pipes do not have storage in this example). The first equation contains these terms and it also describes how the amount of fluid stored in component **A** is changing which is by definition equal to the how the amount of fluid stored in component **B** is changing. Thus this first equation describes everything about mass in the system except:

- the amount of mass stored in component **A**
- the amount of mass stored in component **B**

Looking at the second equation it should be readily apparent that this equation describes the amount of mass stored in the components, therefore we have complete system mass information in these two equations.

5.4.2.4 Implementing the Solution

In practice one must calculate the amount of fluid mass stored in the system before the mass flow rate equations can be applied. In the steady state this is easily accomplished by simply setting the total mass equal to the calculated sum of the individual masses the first time the code is called and holding it constant thereafter. In pseudo-code:

```
In all but one component:
        F(1) = Min - Mout
        Mass(i) = Rho(i) * Vol(i)

In one component:
        Mass(i) = Rho(i) * Vol(i)
        CalcMass=SUM(Mass(*))

        If(FirstTime) Then
                TotalMass = CalcMass
        Endif

        F(1) = (TotMass - CalcMass)
```

In the transient we can use the already-calculated total fluid mass, thus the pseudo code would be:

```
In all but one component:
                Mass(i) = Rho(i) * Vol(i)
                F (1)=(Mass(i)- MassOld(i))/dT – (Min-Mout)

In one component:
                Mass(i) = Rho(i) * Vol(i)
                CalcMass=SUM(Mass(*))
                F(1) = (TotMass - CalcMass)
```

When computing the total fluid mass, care should be taken not to add in components that will have a non-varying fluid mass. If a component doesn't store mass during the transient then:

$$\frac{dM^i}{dt} = 0 \rightarrow \dot{m}_{in}^{i} = \dot{m}_{out}^{i}$$

This does not mean that the component cannot have a fluid mass; it just means that the stored mass cannot change thus:

$$M^i(t) = M^i$$

However, if the component's fluid mass is:

$$M(t) = \rho(t) * Vol;$$

then this dictates that the fluid density in that component cannot change, thus its temperature and pressure cannot change with time. Since any real simulation will have changing temperatures and pressures, in practice this means that if a mass is computed for components that do not have transient mass storage then this mass cannot be added to the total fluid inventory without violating mass conservation. An excellent way to avoid this issue altogether is to set the volume of components to 0 that will not have transient mass storage.

Currently, GAS-PASS/CO_2 only provides volumes for transient mass storage in the PCHEs and turbine, which contain the vast majority of the compressible inventory in the cycle.

5.5 Gas-Pass Solution Techniques

Converging the S-CO_2 recompression cycle initially proved quite difficult. This was expected given the strong non-linearity's in the S-CO_2 properties (see *Section 2.1*) near the critical point where parts of the cycle operate. The reader has probably noted numerous "tricks" and techniques to promote system linearity throughout this report. This section is the outgrowth of what happens when those tricks and techniques fail.

GAS-PASS/CO_2 is quite robust compared to previous versions of GAS-PASS/H. Previously, the failure of almost any process during simulation would lead to simulation failure. Numerous attempts have been made to allow GAS-PASS/CO_2 to continue with a simulation, for as long as is reasonable, even if computation becomes difficult.

The major techniques developed are: progressive solution tolerance, time step failure, Jacobian scaling & numerical turbomachinery performance extrapolation, and user controlled dynamic time stepping. This section will briefly analyze each of these techniques.

5.5.1 Progressive Solution Tolerance

The user sets the numerical solver steady state and transient solution tolerance (roughly 10^{-9} with the current solver) via the input file. If this tolerance is not met and no extrapolation errors have been detected then GAS-PASS/CO_2 will inform the user it is having trouble and attempts to converge with a looser tolerance. This process is repeated until the solution converges or until a very large tolerance is not met and the calculation fails.

In general, this can allow a simulation to run past one or two difficult steps. The S-CO_2 cycle can have unexpected difficult regions such as when the pseudo-critical peak is encountered in the main compressor due to a changing pressure or temperature. By allowing the code to run through one or two difficult time steps a simulation can get past things like the pseudo-critical peak without failure.

A more common occurrence is for this technique to only delay failure by a few time steps. If the code needs to converge to a looser tolerance because it can't converge the transient it is asked to do, then no amount of tolerance relaxation will prevent failure. If this is the case, then once the code stops converging tightly it will rapidly run away from a realistic solution and fail within a few steps. The only advantage of this technique in this case is that it gives the user a heads up that he should carefully check his run since he is asking GAS-PASS/CO_2 to simulate something very difficult.

5.5.2 Time Step Failure

If GAS-PASS/CO_2 cannot converge a time step then it will report the error and simply move on to the next time step and try again. It is not uncommon for GAS-PASS/CO_2 to fail at a single time step but recover and successfully run afterwards. This occurs in practice when a boundary condition changes sharply or the system hits a difficult point like a pseudo-critical property peak. By allowing GAS-PASS/CO_2 to fail but recover the user can gain a lot more information about the simulation and the problem encountered than by simply letting the code exit.

5.5.3 Jacobian Guessing

When using a Newtonian numerical solver with a non-linear system it is frequently necessary, for numerical purposes, to provide system values well outside of operating regions. A Newtonian solver works well in a linear region, which in benign systems (i.e. when simulating an ideal gas plant) can be quite large, but in more non-linear systems (like a S-CO_2 plant) the linear region, always present near the root, can be quite small.

Therefore, in non-linear systems it is necessary either to assure that one stays within the linear region or that the code can handle some operations in the non-linear region. While the former goal is desirable it is probably not possible for realistic operating conditions or transients and may be quite difficult numerically. However, the second goal can be accomplished in several ways.

When simulating a plant several variables are likely to go well outside of expected bounds including fluid properties and turbomachinery performance curves when numerical perturbations of the system, during root finding, are performed. Obviously, it is desirable to have a wide range on the property tables and the performance curves but these are limited by fundamental physics like freezing and disassociation or stall and surge. Even if one makes the fluid property tables and turbomachinery performance curves as large as is physically possible it is entirely plausible that a Newtonian numerical solver will request values well outside of these bounds. This is especially likely to occur during the solver's first solution after a Jacobian has been calculated.

5.5.3.1 Jacobian Slope Estimation

Figure 5-5 shows a two dimensional example of guessing the system root using the Jacobian. If the dotted blue line in the figure represents the function we are modeling and we are trying to arrive at the root then the difference in calculating the slope from the non-linear point *A* or the more linear point *B* is large.

Since we are using a Newtonian solver our Jacobian is a one dimensional derivative, thus we are just taking a slope from the initial guess shown by the black lines connected to *A* and *B*. Unfortunately, if the function is non-linear then this slope can easily predict a root a long way from the true value, as happens with *A*. Correspondingly, point *B*, which is relatively close to point *A* on the *X* axis, predicts a value much closer to the true root.

In practice this means that the guess from point *A* will probably put the system outside of realistic performance bounds and generate some type of error while point *B* will converge nicely. This can be quite frustrating to the user since point *B* may be only slightly different from point *A*.

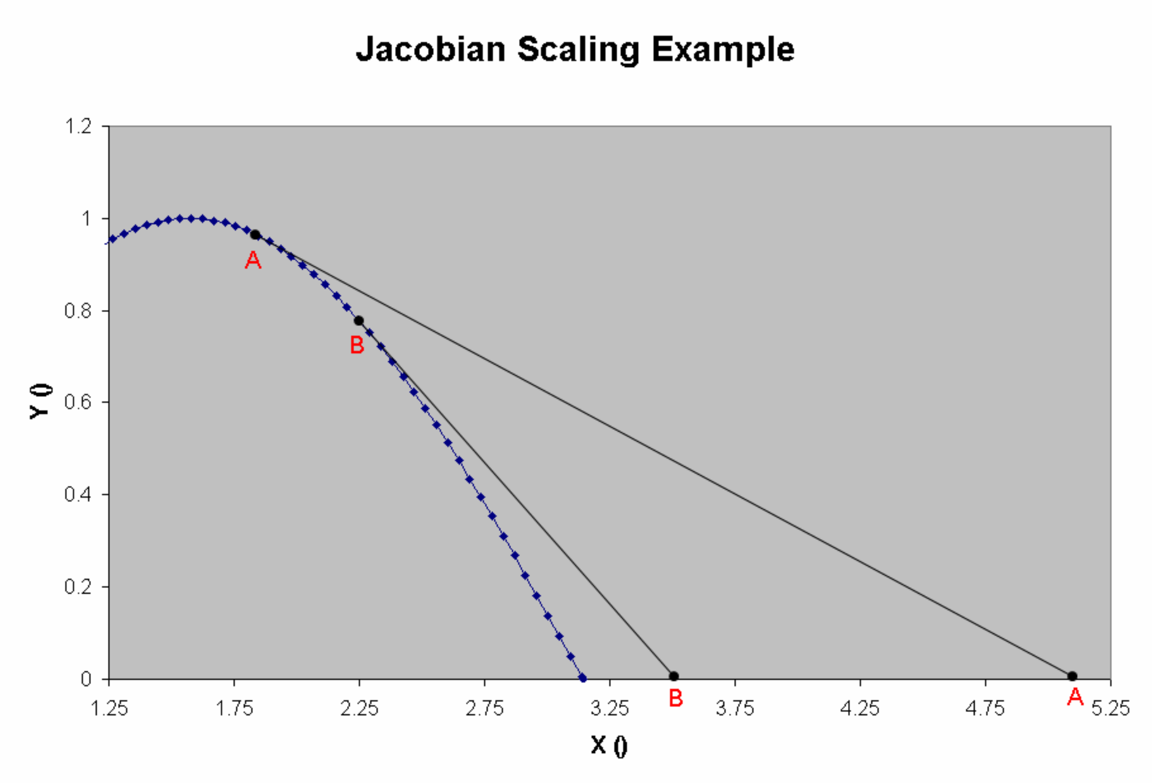


Figure 5-5: Jacobian Scaling Example

5.5.3.2 Jacobian Scaling

One method to minimize this problem is to scale the Jacobian. A scaling factor may be applied to the (whole) Jacobian to prevent the solver from taking unrealistically large steps and may simply be thought of as decreasing the predicted distance to the root. Note that when multiplying the Jacobian by this factor a *larger* scaling factor decreases the distance to the root. The scaling factor should not be less than one since good numerical solvers will automatically converge on the root without external assistance.

GAS-PASS/CO_2 handles this Jacobian scaling in two manners:

1. The user can specify a default Jacobian scaling factor for both the steady state and the transient in the input file. A large scaling factor is appropriate for a highly non-linear system and a value of one is appropriate for a more linear problem. A

larger factor will slow the solution process (more steps are required to find the root) but may be necessary if the solver is taking many unrealistically large steps.

GAS-PASS/CO_2 normally runs with a value of 10 for the steady state (since the user may be making some big guesses about the root location and the steady state is a small percentage of runtime) and a value of 1 during the transient since GAS-PASS/CO_2 can scale automatically.

2. GAS-PASS/CO_2 automatically increases the Jacobian scaling factor if property errors are detected. In practice, the first indication of an unrealistically large step will be the solver asking for a fluid property that is out of table bounds. When this happens an error flag is generated, the solver stops computation and increases the Jacobian scaling factor, and then the solution process begins again. This recovering and rescaling happens repeatedly until the system converges or the Jacobian becomes unrealistically large and the time step fails.

   If the system converges then a true root has been found, all is well, and the code moves on to the next time step. If the system does not converge then GAS-PASS/CO_2 will give up on this time step and try the next in hopes that changing boundary conditions will allow a future solution (it occurs in practice). All of these actions are printed to the screen and written to the log file to warn the user and allow appropriate action.

Jacobian scaling is one of the most important solution-enabling techniques used in GAS-PASS/CO_2.

5.5.3.3 Numerical Extrapolation

Unfortunately, Jacobian scaling may not prevent all out of bound errors. An example of where this might occur is in the performance curve of the axial main compressor used in the S-CO_2 recompression cycle. The efficiency curve of this compressor at normal shaft speed is shown in *Figure 5-6*.

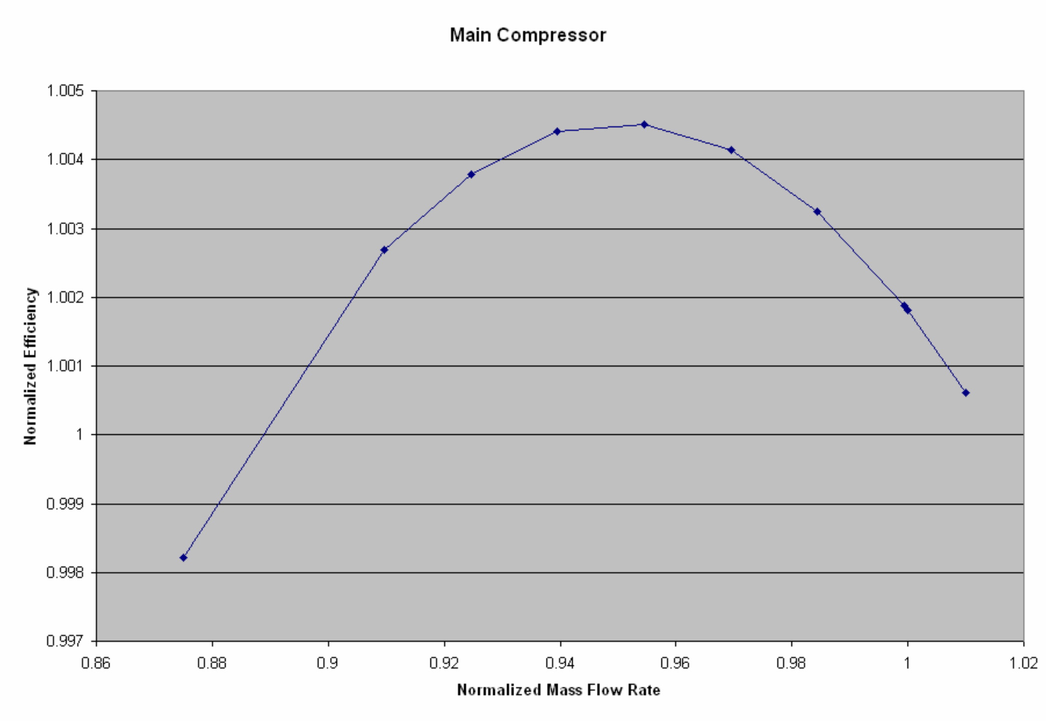


Figure 5-6: Axial Main Compressor Efficiency

This curve shows the expected physical performance of an axial compressor, but note that the operating range is quite small. The detailed compressor design code used to create this curve was only able to converge data points slightly less than 1% higher than the design mass flow rate.

An obvious solution to this problem is to extrapolate the performance curves past their expected region; however, one must be careful how this is implemented. Not only will an axial turbomachine stall and choke in a narrow range but simple linear extrapolation is insufficient. *Figure 5-7* shows the same performance curve with point **A** linearly extrapolated (from the last two data points) to 10% above design flow rate. This figure clearly shows that even relatively small mass flow rate changes can cause negative efficiency values from such tight compressor curves.

One solution to this problem is to create non-physical but numerically significant extrapolation. This numerical extrapolation would only be used to tell the solver that it is overshooting the root while simultaneously assuring that nonsensical answers (i.e. a negative efficiency) do not produce mathematical errors. An example of what this might look like is shown in *Figure 5-8*.

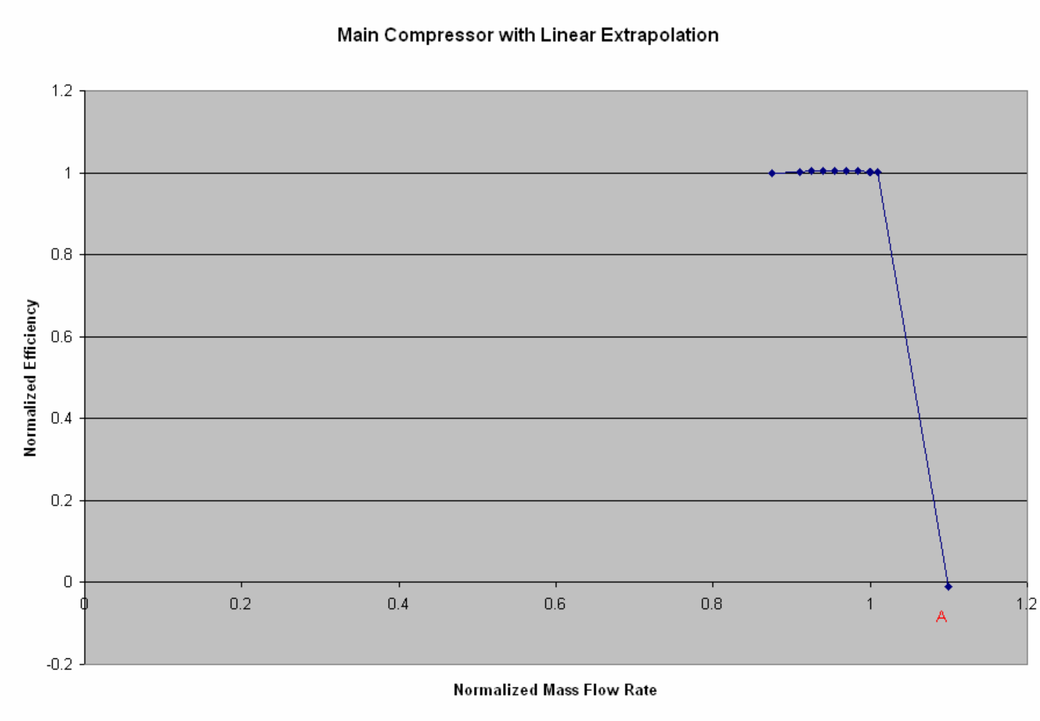


Figure 5-7: Axial Main Compressor Efficiency with Linear Extrapolation

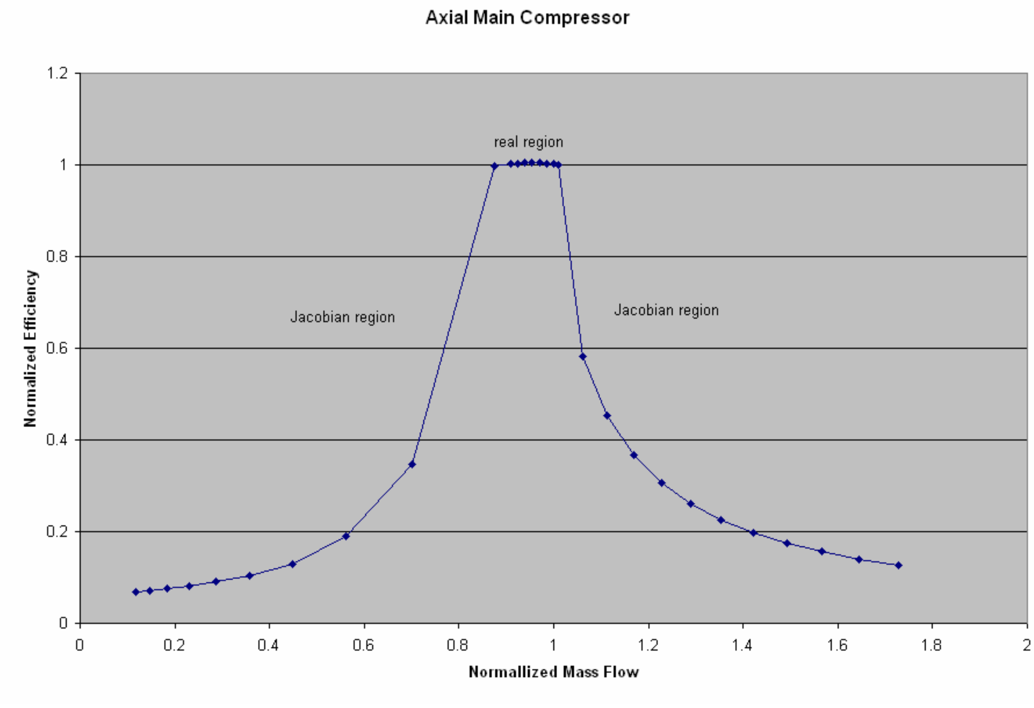


Figure 5-8: Axial Main Compressor Efficiency with Added Extensions

Figure 5-8 shows the narrow real operating region with much larger but non-physical efficiency curves attached to either side. These curves show a significant drop in initial efficiency that asymptotically decreases as the mass flow rates become ever further from the operating region. These curves were devised to show this behavior for several reasons:

- Everywhere outside of the real region will predict a very low efficiency that will inform the Jacobian it is not near the root (i.e. solution equations will be non-zero to a significant degree).
- The extrapolated curves are smoothly sloping with a derivative that informs the Jacobian that to achieve higher efficiencies it must move back towards the real region.
- The curves cover a very wide mass flow rate without introducing unrealistic values (i.e. an negative or much greater than 100% efficiency).

The rational for allowing these non-physical responses is that the converged code will never operate here. As shown in *Figure 5-5* the Jacobian is simply making predictions of the root and testing to see if it is accurate. When it tests these extrapolated data points it will discover that it has moved too far from the root and move back towards its original point – it will also keep the code from mathematically failing. GAS-PASS/CO_2 includes a check to make sure that no converged operating conditions (i.e. a time step's root) ever include non-physical extrapolation.

The simple but general way used in GAS-PASS/CO_2 to create the extrapolated curves shown in *Figure 5-8* is given below:

```
If the known axial operating region extends from $m_1 \ldots m_n$ then let $X_m = m_n - m_1$

If $m_{desired} > m_n$ then
        $Z = \frac{m_{desired} - m_n}{X_m} + 1$   which by definition is larger than 1.
Else If $m_{desired} < m_1$ then
        $Z = \frac{m_1 - m_{desired}}{X_m} + 1$   which by definition is larger than 1.
Endif

$f = 0.8 * MIN(\eta_1 ... \eta_n)$   thus giving a significantly smaller efficiency than all known real points
$\eta_{extrapolated} = f * \frac{1}{Z}$   we now have the efficiency shown in the figure
```

Note that this process may prevent GAS-PASS/CO_2 from failing due to extrapolation during root finding – it will not prevent the solution from moving past the turbomachinery performance map bounds. For example, note that in *Figure 5-6* there is very little mass flow rate margin above the design operating point of 1. Any simulation that increases the mass flow rate will probably exceed this map's boundaries.

Wider turbomachinery performance maps can be accommodated by using different turbomachines, such as using radial compressors. This issue is discussed in *Section 7.1* for the S-CO_2 recompression cycle.

5.5.4 User Controlled Dynamic Time stepping

User controlled dynamic time stepping enables the user to avoid simulation failure in practice. In an ideal world a simulation would occur at the smallest time step necessary at any point in the simulation, and code would run fast enough that this would not matter (as occurred with GAS-PASS/H simulating ideal gases). For GAS-PASS/CO_2 it is frequently necessary to have a small time step during rapid boundary condition changes, but this simulation is unnecessarily detailed during the vast majority of the simulation when boundary conditions vary slowly or not at all.

GAS-PASS/CO_2 allows the user to avoid this problem by controlling the time step size. It has been used quite successfully to simulate sharp perturbations in boundary conditions. For example, the transient simulated in *Section 6.2* has a very rapid change for ¼ of a second of a 1000 second simulation. Before the perturbation the simulation is effectively in a steady state and any-sized time step may be used. During and shortly after the perturbation very small time steps are required to capture the mass storage and major energy storage effects. Gradually, the mass storage effects decrease and the energy storage becomes less important. The user can therefore use progressively larger time steps until at large timescales the system once again reaches a steady state.

The time step instructions used for the simulation shown in *Section 6.2* were:

```
1000.0          :total problem runtime  [s]
0.01            :default time step size [s]
4               :# of time step instructions
0.0 5.0  1.0 1.0  1.25 1.0  1001.0 1000.0
```

The last line is in standard GAS-PASS/CO_2 boundary condition format. It tells the code to take 0.05 second time steps initially, linearly decrease the step size until it is taking 0.01 second time steps at the perturbation, hold that time step size until the perturbation is over, then linearly increase the time step size until at 1001 seconds 10 second time steps are being taken.

The net results of these instructions are to run this simulation with 638 time steps instead of 10^6 representing a, roughly, 157 fold speedup for the end user. The user will observe that the code takes longer to run the first 13 seconds of the simulation than it does the last 987 seconds.

While this is not strictly a solution technique, the great speedup it allows encourages the user to take appropriately small time steps during changing conditions and to run the simulation until it reaches a new steady state. This makes GAS-PASS/CO_2 more likely to run a successful simulation, and it is therefore included in this section on solution techniques.

5.5.5 Solution Techniques Conclusion

GAS-PASS/CO_2 has benefited from the difficult process of developing and initially converging the S-CO_2 recompression cycle with a progressively more robust solution process. The code may still fail if asked to simulate an unrealistic transient, but the code will now adapt to and survive many difficult transients without changing the fundamental solution process.

5.6 GAS-PASS/CO_2 Summary

GAS-PASS/CO_2 is a complete code, able to successfully simulate the S-CO_2 recompression cycle. It has the capability to simulate real fluid properties, turbomachinery performance maps, and PCHE heat exchange and pressure drop. The coding and input/output process have been significantly improved, making it much easier for the user and the developer to work with the code, and the solution process has been made much more robust, allowing greater success with more difficult cycles.

GAS-PASS/CO_2 will continue to be improved but only in response to future simulation needs and control system design requirements. Code testing has shown that relatively non-linear systems can be successfully simulated with Newtonian solvers given careful consideration of the system being modeled.

6 S-CO_2 Cycle Dynamic Performance

This chapter analyzes the general dynamic performance of the S-CO_2 recompression cycle focusing on understanding the gross behavior of the cycle, in the absence of detailed control actions which are to be introduced in subsequent work.

All of the transients shown in this chapter begin with a steady state system that is perturbed by a relatively rapid change in a boundary condition for a relatively short time. At the end of this perturbation the boundary condition is held at its new value with all other boundary conditions being held constant through the simulation. Therefore, there is no control system and all of the plant effects are due to inherent feedback from the system design.

The plant modeled in these simulations is for a 42°C main compressor inlet, 650 °C turbine inlet, uses radial compressors and an axial turbine, and functions at a fixed shaft speed with variable generator demand. The system flow split is controlled solely by the compressors balancing outlet pressures.

All four PCHEs have mass and energy storage as does the turbine. The mass and energy storage terms in this cycle are initial estimates of what is appropriate for these simulations. They will be refined in future work and the sensitivity of this plant to changing them will be characterized as well.

All other components use quasi-static mass and energy equations. Due to the relatively incompressible behavior of the fluid in the high pressure, cold side of the cycle there is relatively little change in mass storage and transients occur very rapidly compared to the rest of the cycle. Including these effects would be more useful in a code which carefully analyzed fluid momentum on short timescales and is neither necessary nor desirable here.

While the turbomachines are allowed to move within their predicted flow map as mass flow rates change to balance pressures, they are prevented from extrapolating beyond predicted mass flow rate bounds.

The chapter is subdivided into five sections:

1. Section one looks at the effect of variable fluid properties on the compressors.
2. Section two analyzes a decreased reactor power transient.
3. Section three analyzes an increased reactor power transient.
4. Section four analyzes an increased cooling water mass flow rate transient.
5. Section five summarizes this chapter.

The analysis in this chapter generally features two timescales – an immediate timescale during and shortly after the perturbed boundary condition and a long-term timescale showing the plant arriving at the new steady state value appropriate to the perturbed boundary condition. Note that plots show only every 10th data point for clarity.

6.1 Compressor Property Effects

Using parallel compressors with S-CO_2 presents unique behavior due to fluid property effects. Gas-Pass currently handles off-normal property effects by adjusting the pressure rise by the density ratio as shown in *Equation 6-1*. This equation states that the compressor outlet pressure is equal to the calculated pressure rise (based upon mass flow rate and shaft speed) at design conditions multiplied by the current fluid density divided by the design condition (steady state) density. As noted earlier, radial compressor characteristics are used throughout.

Equation 6-1: Compressor Off-Normal Property Correction

$$\Delta P = \Delta P_0 \frac{\rho}{\rho_0}$$

While simple in theory, in practice this effect will produce unique behavior due to the different density changes experienced by the compressors. The non-linear behavior of S-CO_2 fluid properties guarantees that if components begin with different fluid properties then a change in fluid properties will affect the components differently. This will be elucidated in the following example.

6.1.1 Increasing Heat Rejection

Figure 6-1 shows the compressor outlet pressures during an increasing heat rejection transient. This simulation forced the precooler water mass flow rate to linearly increase by 50% from 1 to 10 seconds – otherwise boundary conditions are constant.
This figure shows the main compressor pressure gently increasing then gradually decreasing. The recompressing compressor steeply linearly drops in pressure until the water mass flow rate stops changing at which point it linearly drops less steeply.

Figure 6-2 shows the compressor pressures during an identical transient except that fluid property variation is not taken into account. In this figure the recompressing compressor behaves nearly identically while the main compressor outlet pressure drops steeply, nearly in line with the recompressing compressor.

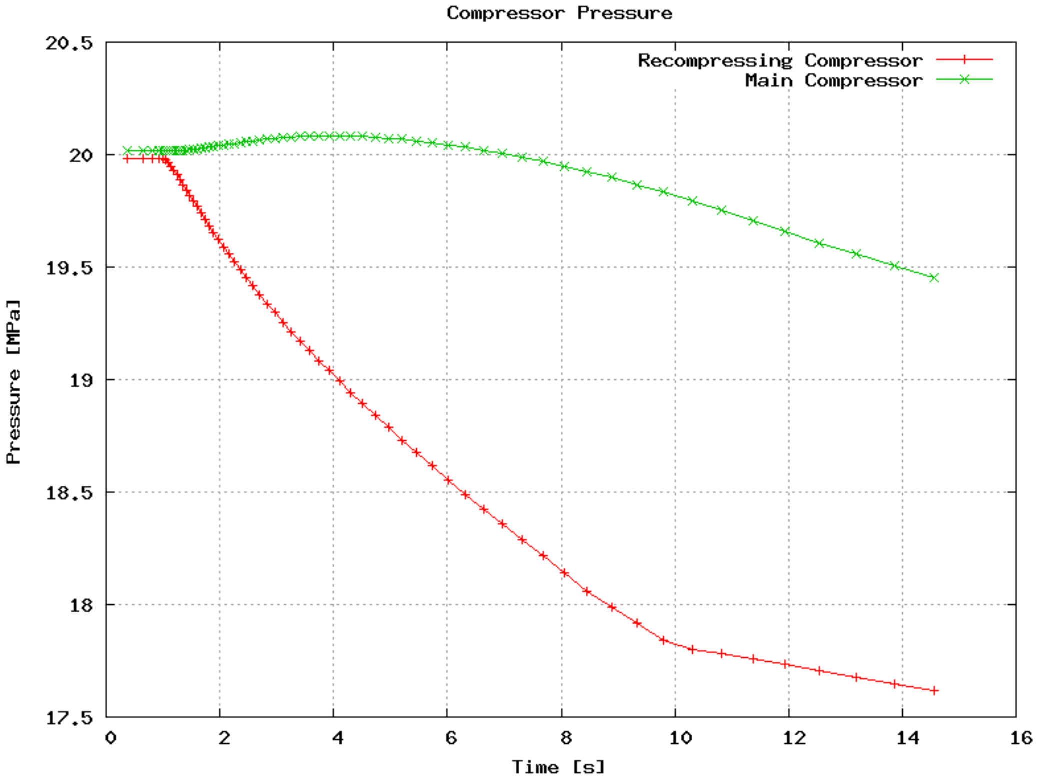


Figure 6-1: 150% Mpre - Compressor Pressures

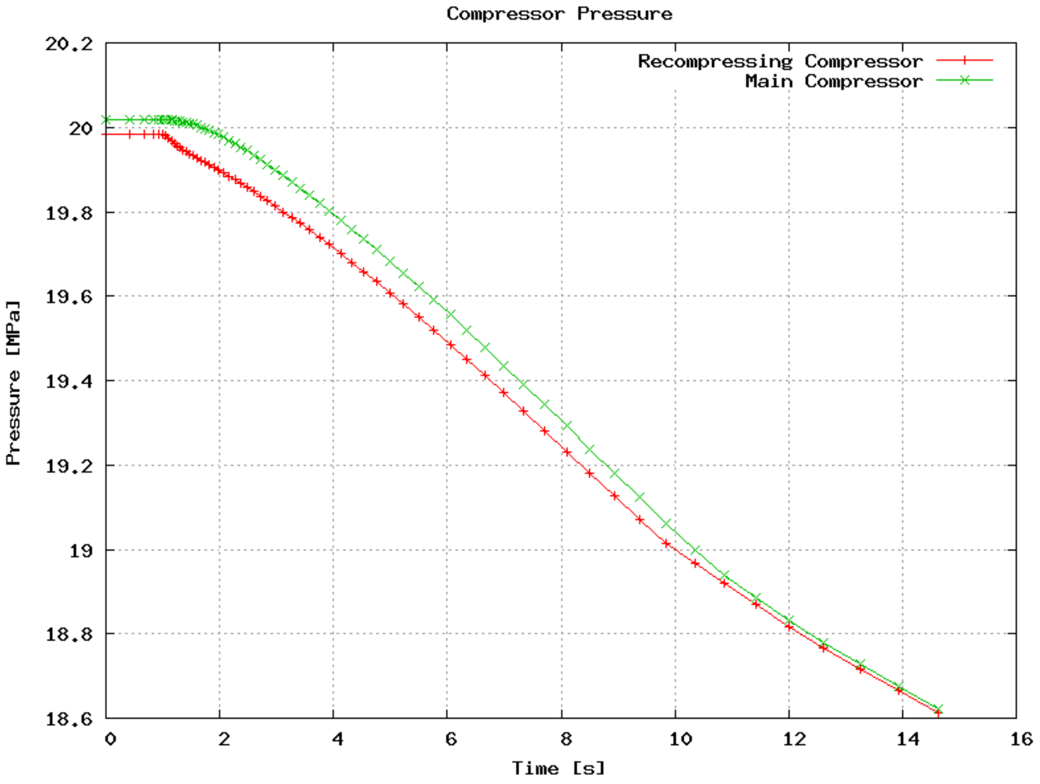


Figure 6-2: 150% Mpre - Compressor Pressures without Property Variation

The difference between the figures stems from the main compressor experiencing an increase in fluid density as the increasing water flow rate over-cools the S-CO_2 in the precooler. The net result is that accounting for fluid property variation produces an almost 2 MPa pressure difference in the compressor outlet pressure to almost zero in the no variation case. Note that this simulation was performed to illustrate fluid property variation effects. In a realistic case the mass flow rates of each compressor would be allowed to float to match outlet pressures. One might conclude that changes in heat rejection will be felt more strongly and more quickly in the main compressor than in the recompressing compressor and vice versa.

6.2 Decreasing Reactor Power Transient

This section analyzes the effect of decreasing reactor power by 25% over one quarter of a second. This rapid transient clearly shows the effect of decreasing power perturbations on mass and energy storage and compressor performance over the short and long-term. The transient occurs at one second when the reactor power is linearly decreased to 75% of its original value by 1.25 seconds.

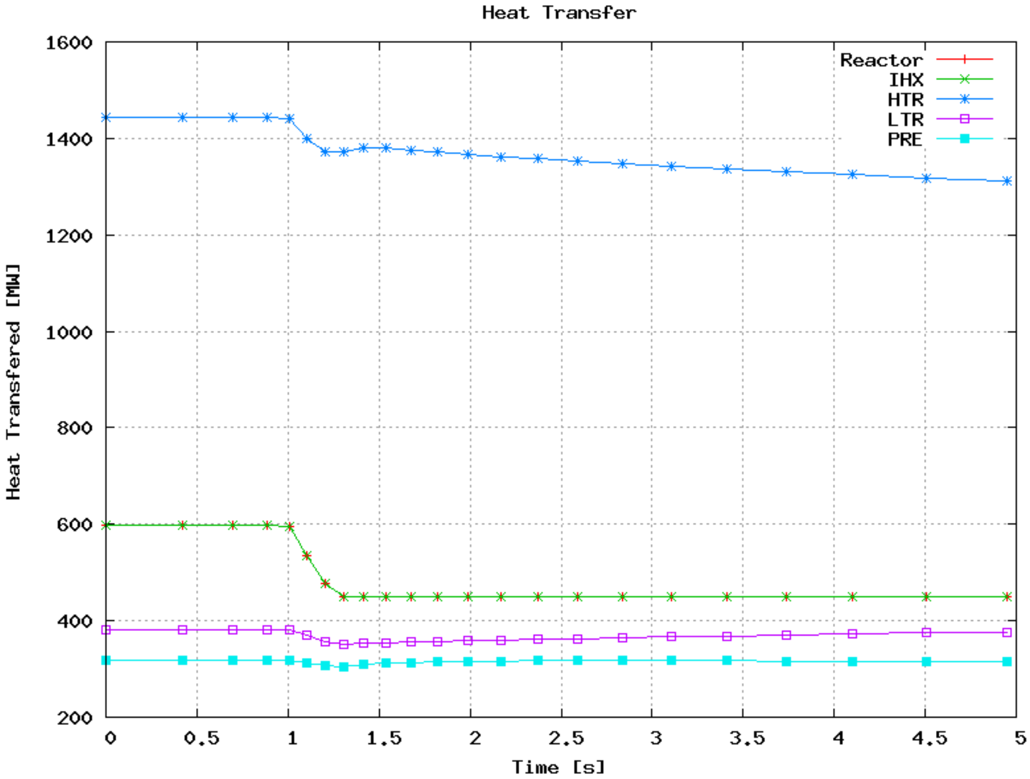


Figure 6-3: -Qrx Immediate Transient - Heat Transfer

Figure 6-3 shows heat transfer rates within the plant over a short timeframe around the perturbation. The IHX/reactor lines exactly match and show the amount of heat added to the cycle, and this transient's changing boundary condition. At the heat transfer perturbation the HTR line shows a somewhat rapid drop in heat exchange that slightly overshoots then gradually and linearly drops once the reactor stops changing power level. The LTR shows similar behavior but with a much smaller change and no overshoot. The PRE shows very little decrease in heat transfer. These relative effects in the PCHEs are expected because only the heat addition is being affected, therefore the hot temperatures in the cycle will change the most while the lower temperatures will not change significantly, as shown in *Figure 6-4*.

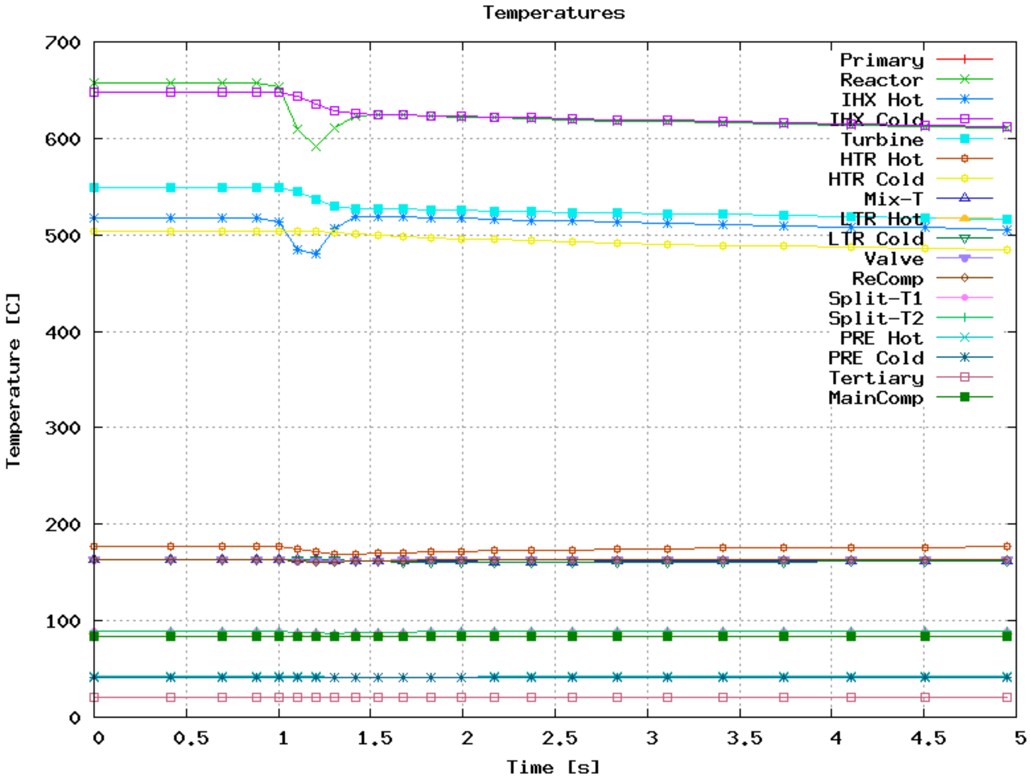


Figure 6-4: -Qrx Immediate Transient - Temperatures

Figure 6-4 shows the effects of the decreasing reactor power on the cycle temperatures. The hot side shows a relatively rapid drop during the perturbation and then a slow linear drop afterwards. The cold side shows a lesser drop as the fluids become colder, with the precooler CO_2 outlet temperature almost unchanged on this timescale. The IHX primary side temperatures show unique behavior during the perturbation with very sharp drops in temperature during the transient (since mass flow rate is held constant) that increase gradually at the end of the power perturbation. This is an artifact of the unrealistically fast drop in IHX heat transfer, which is forced to drop before the S-CO_2 fluid side has a chance to respond.

Figure 6-5 shows the immediate effect of the decreasing reactor power on mass flow rates. The mass flow rates are generally constant except for a short decrease in flow rates during the perturbation. This effect comes from the PCHEs, primarily the HTR, as shown in *Figure 6-6*, which temporarily decreases mass flow rate by absorbing fluid. It is mitigated by the turbine which rejects fluid during the perturbation as shown in *Figure 6-7*.

A change in temperatures will affect the density of the fluid in the components differently due to S-CO_2's non-linear property effects. For a decrease of reactor power this results primarily in the turbine shedding stored fluid and the HTR (mainly the cold side) absorbing it. For a short period of time this results in an increase in generator and turbine power as the turbine sees an increased mass flow rate.

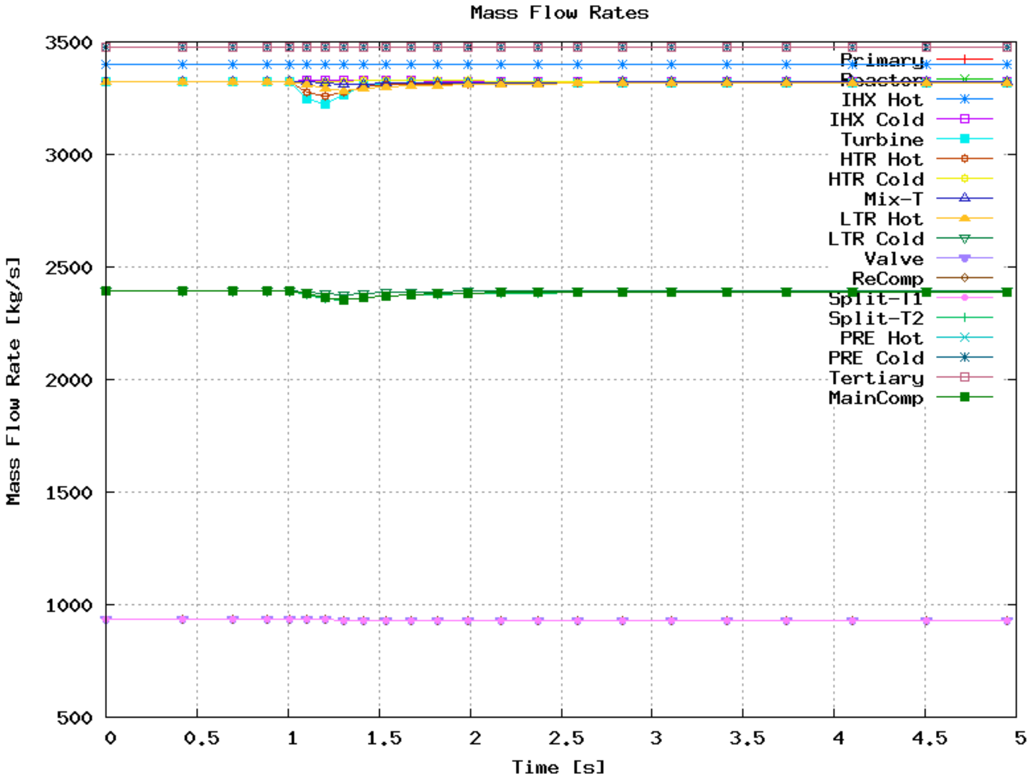


Figure 6-5: -Qrx Immediate Transient – Mass Flow Rates

The long-term effect of a decrease in reactor power must be a decrease in efficiency as shown in *Figure 6-8*. This figure shows that the efficiency of the cycle gradually settles to roughly 38% from its original 46% by around 200 seconds. Note that there is an initially a sudden, short spike in efficiency, to a very high value, as reactor power drops before the system has a chance to cool.

The long-term effect of decreasing reactor power is to lower system temperatures and pressures as shown in *Figure 6-9* and *Figure 6-10*. As in the short term results, the temperature and pressure drops are concentrated on the hot side of the cycle with the precooler changing little. It is worth noting that the vast majority of the transient temperature and pressure behavior occurs within the first 200 seconds. There is also a small decrease in overall mass flow rate, with a small decrease in main compressor and a slight increase in recompressor mass flow rates.

In summary, a decrease in reactor power produces a short drop in mass flow rates due to shifting mass storage and a long term decrease in temperatures and pressures primarily affecting the hot end of the cycle.

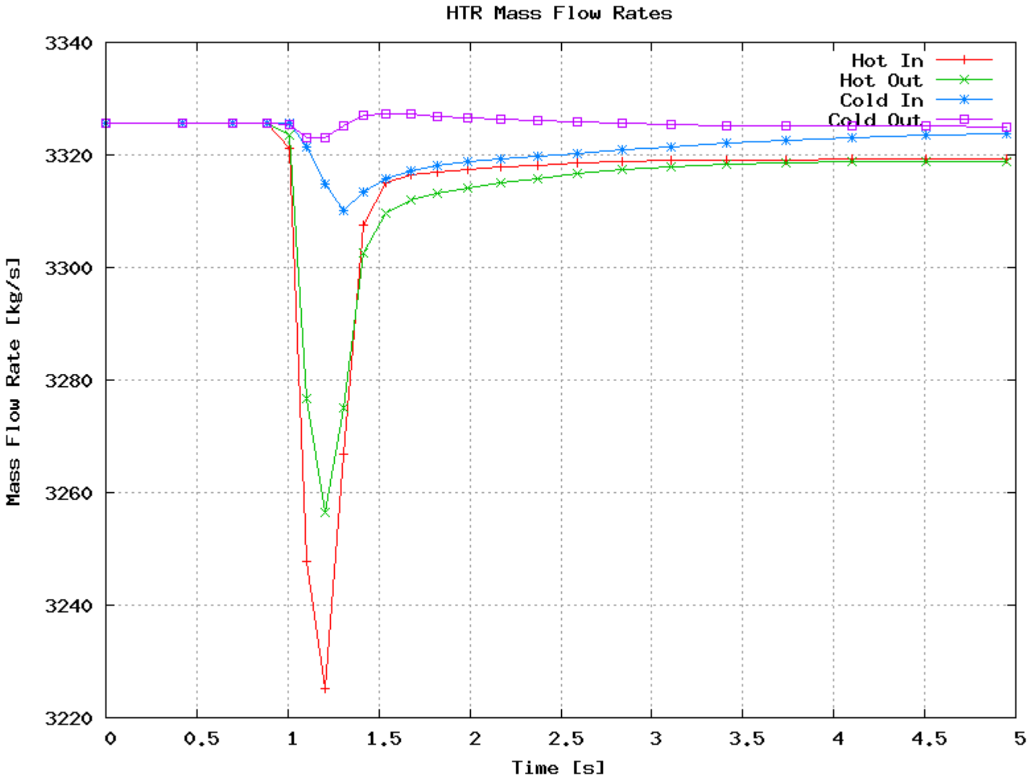


Figure 6-6: -Qrx Immediate Transient - HTR Mass Flow Rates

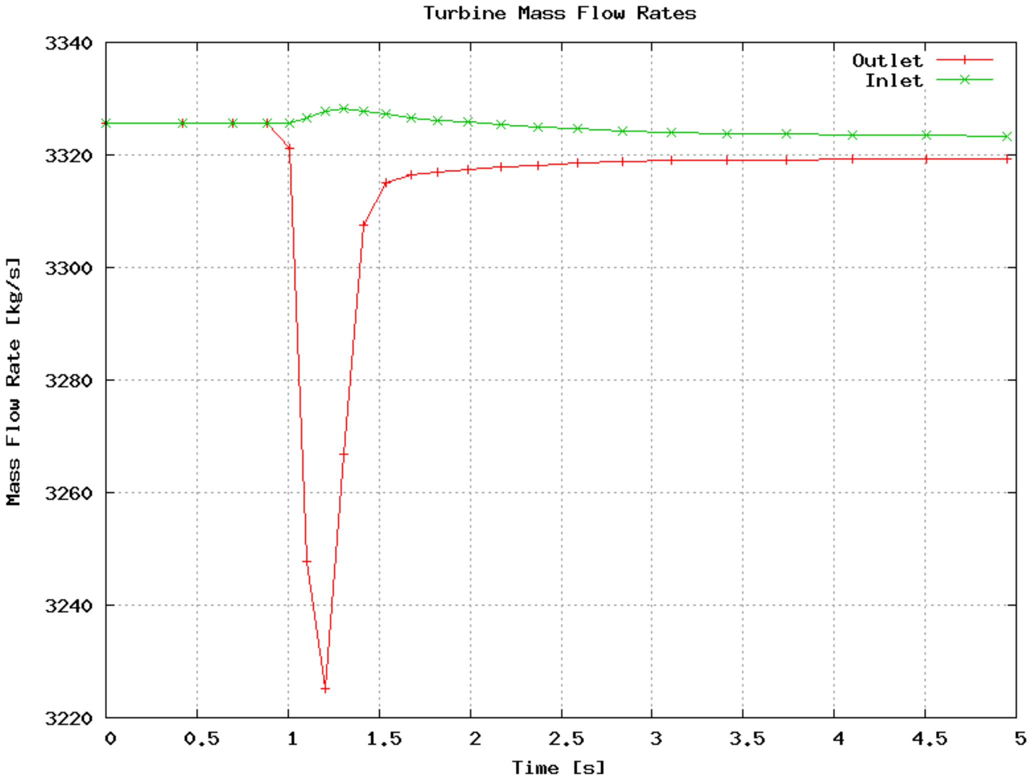


Figure 6-7: -Qrx Immediate Transient - Turbine Mass Flow Rates

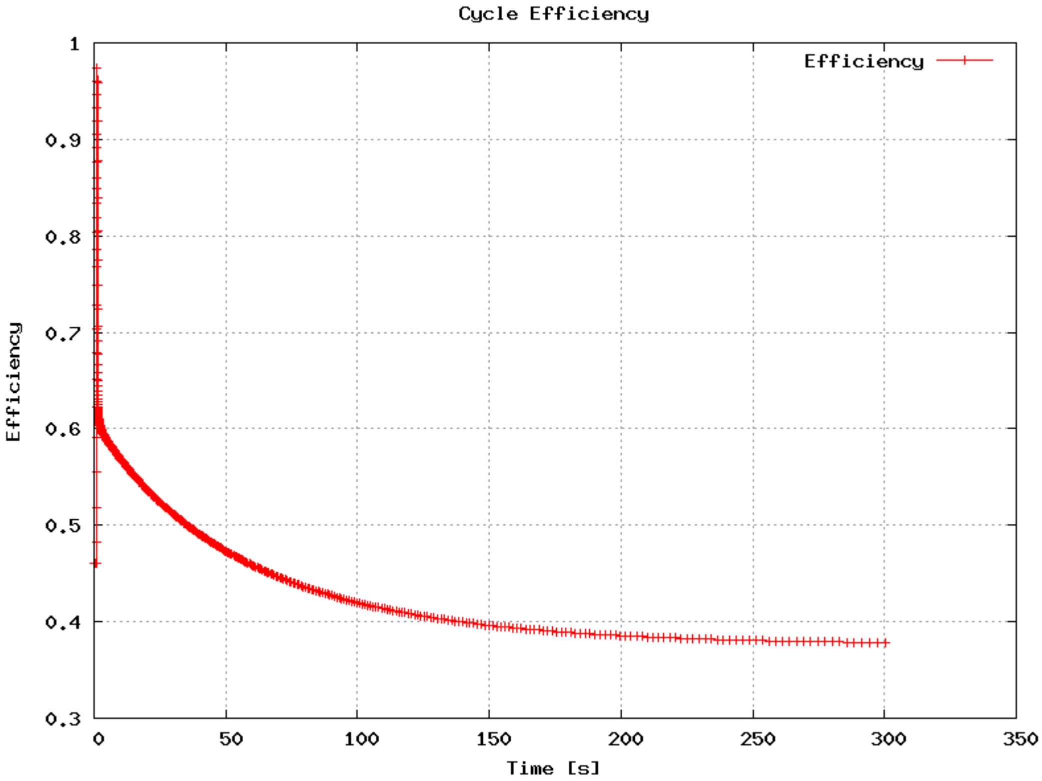


Figure 6-8: -Qrx Long-Term - Efficiency

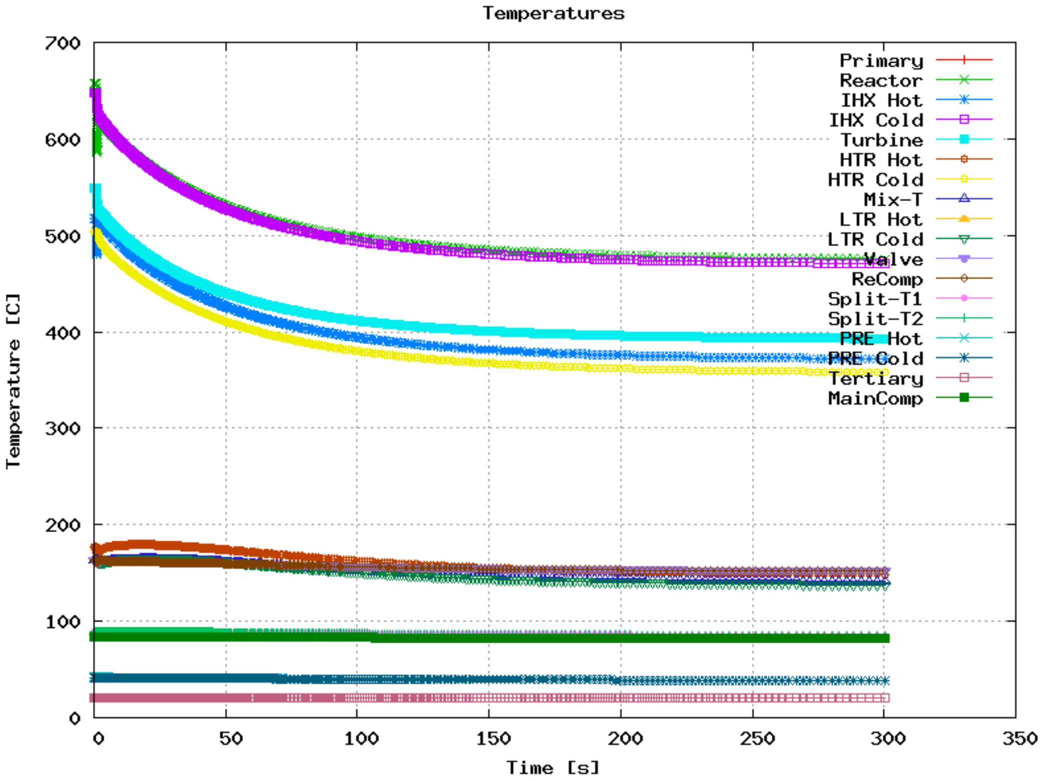


Figure 6-9: -Qrx Long Term – Temperatures

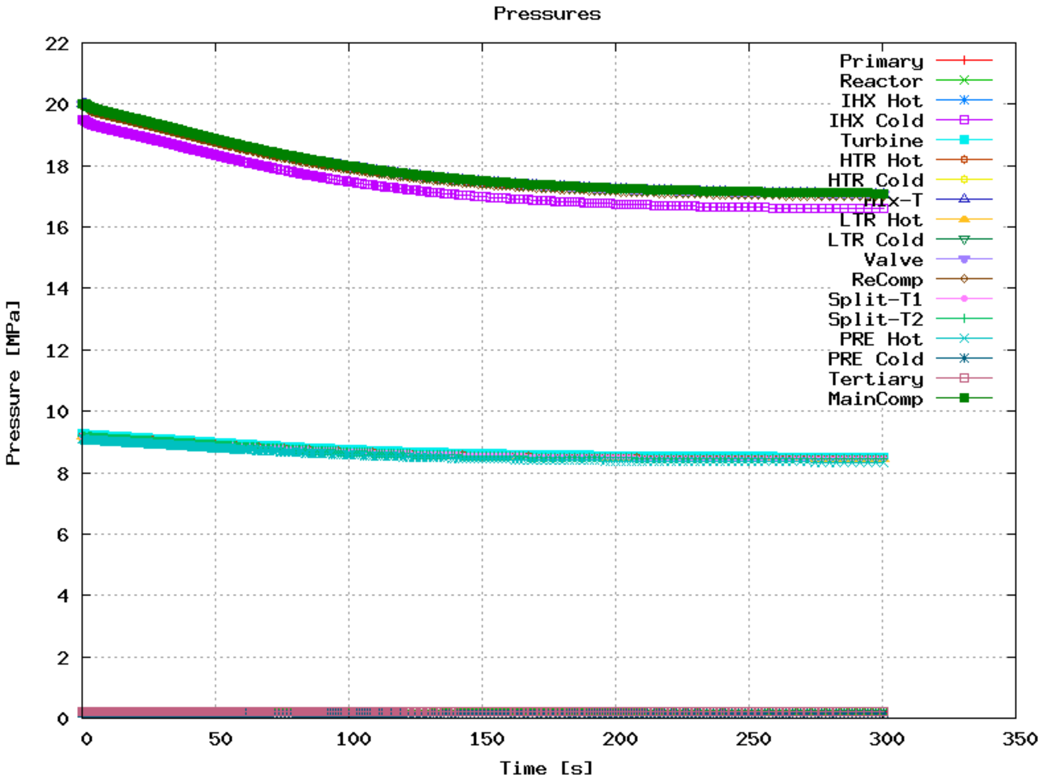


Figure 6-10: -Qrx Long Term – Pressures

6.3 Increasing Reactor Power Transient

This section analyzes the effect of increasing reactor power by 10% over one quarter of a second. This transient shows nearly the opposite performance effects of those previously seen in the decreasing power perturbations, therefore fewer plots will be offered. It is included both as a check on the simulation code and to provide a guideline for overpower transients.

The transient occurs at one second when the reactor power is linearly increased to 110% of its original value by 1.25 seconds. The short term effect on heat transfer is shown in *Figure 6-11*. This figure shows a linear increase in IHX power, a smooth and relatively large increase in HTR power, a smaller increase in LTR power, and a very small increase in precooler power. As expected, the effect of decreasing heat addition at the hot end of the cycle concentrates its affects on the hot end, with very little change produced in the cold region.

The short term mass flow rates show the inverse of the behavior seen in the decreasing power perturbation. This is clearly seen by comparing the HTR mass flow rate in this case, *Figure 6-12*, to the previous short term HTR mass flow rate in the decreasing power case, *Figure 6-6*. The net effect of the mass flow rates is to temporarily decrease the fluid supplied to the turbine which therefore lowers turbine work as shown *Figure 6-13*.

Figure 6-13 shows a rapid drop in turbine work which results in a rapid decrease in generator work since shaft speed is fixed. There is relatively little change in the compressor work requirements due to constant shaft speed and little change in compressor inlet temperatures as shown in *Figure 6-14*.

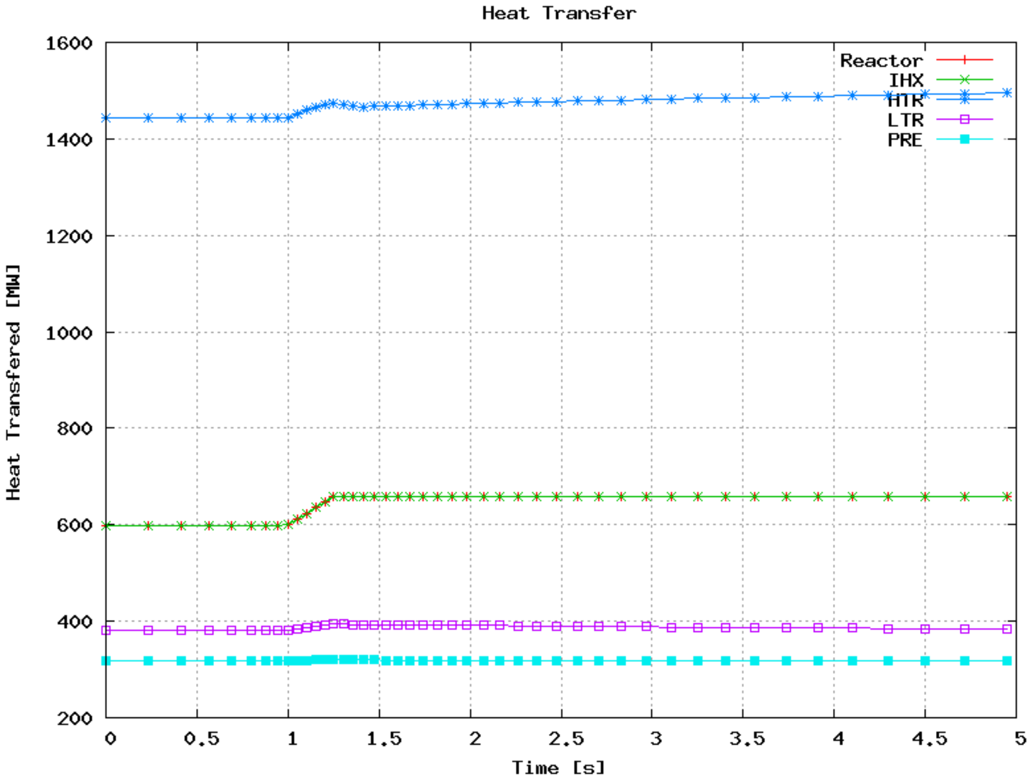


Figure 6-11: +Qrx Immediate Transient - Heat Transfer

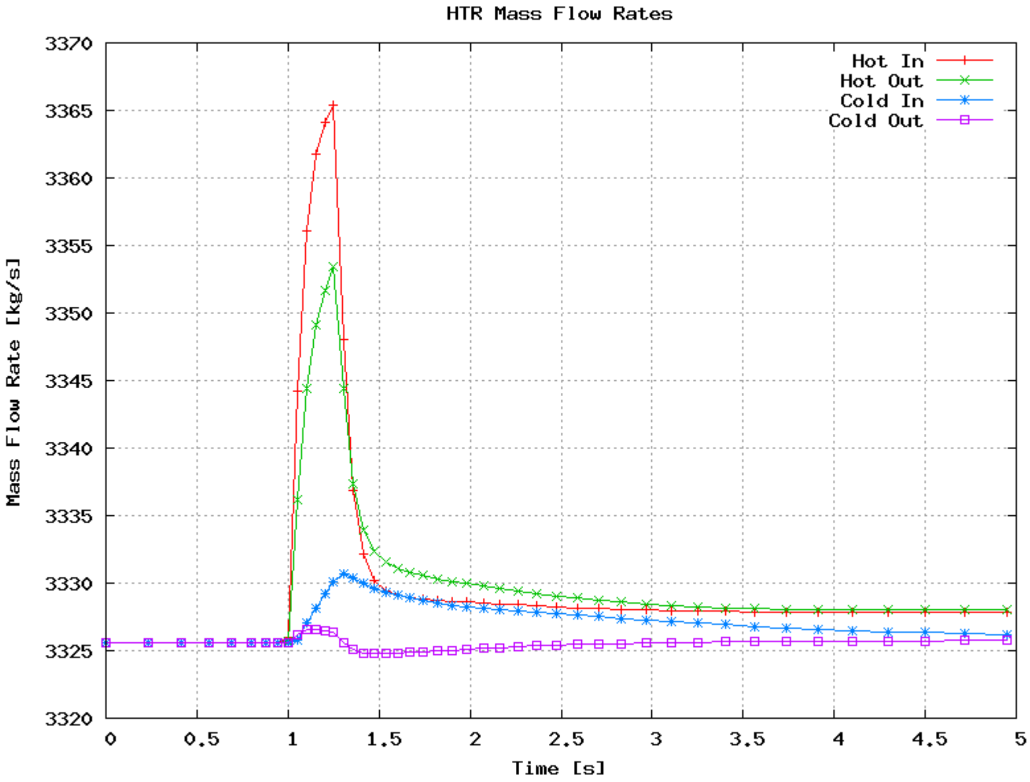


Figure 6-12: +Qrx Immediate Transient – HTR Mass Flow Rates

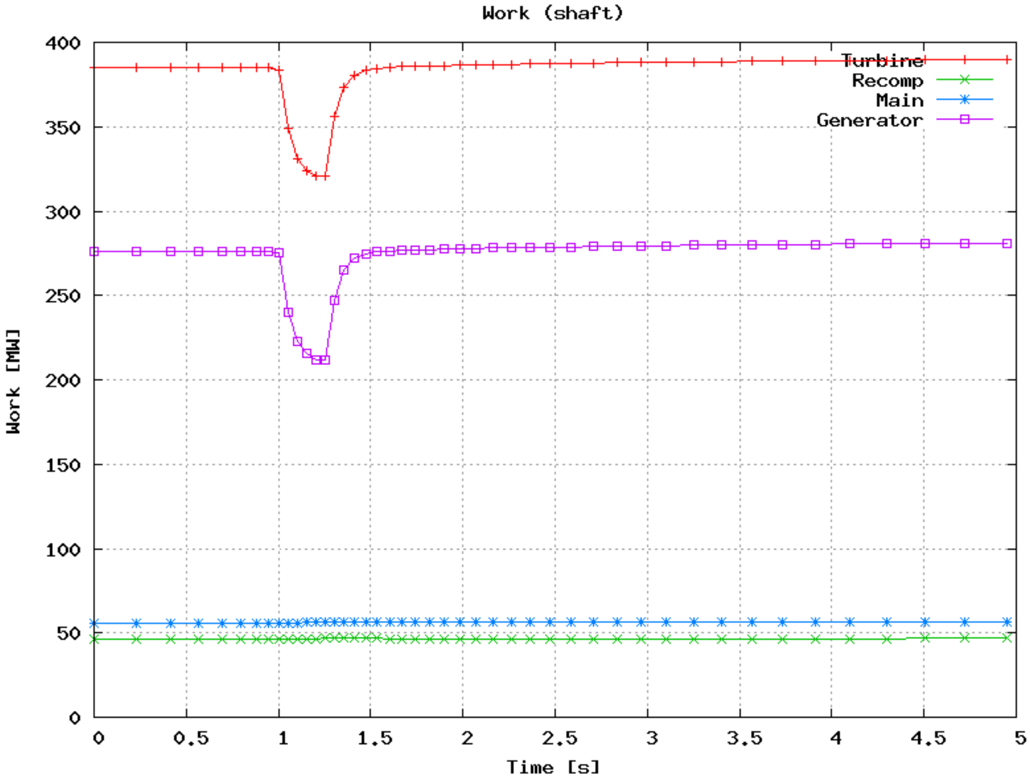


Figure 6-13: +Qrx Immediate Transient – Work

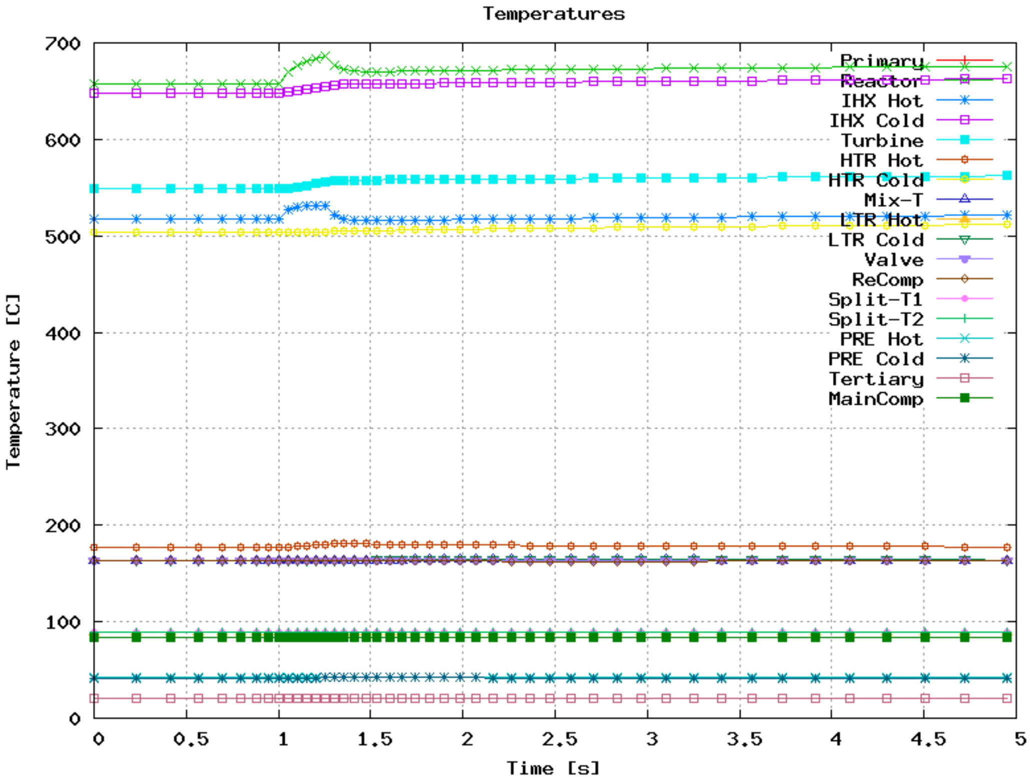


Figure 6-14: +Qrx Immediate Transient - Temperatures

The long-term effect of increasing reactor power is to increase temperatures and pressures. Their effects on heat transfer and work are shown in *Figure 6-15* and *Figure 6-16* respectively.

These figures show relatively little change in the cold end of the cycle either through heat transfer or compressor work, but a relatively large increase in the hot end of the cycle. The generator work moves from its design value of 276 MW and 46.1% efficiency to 320 MW and 48.6% efficiency. Note that these efficiency values do not include pipe pressure drop effects.

In summary, an increase in reactor power produces a short term drop in turbine power due to shifting mass storage and a long term increase in temperatures and pressures primarily affecting the hot end of the cycle. Generator power increase is solely due to higher cycle efficiency, a consequence of higher turbine inlet temperature.

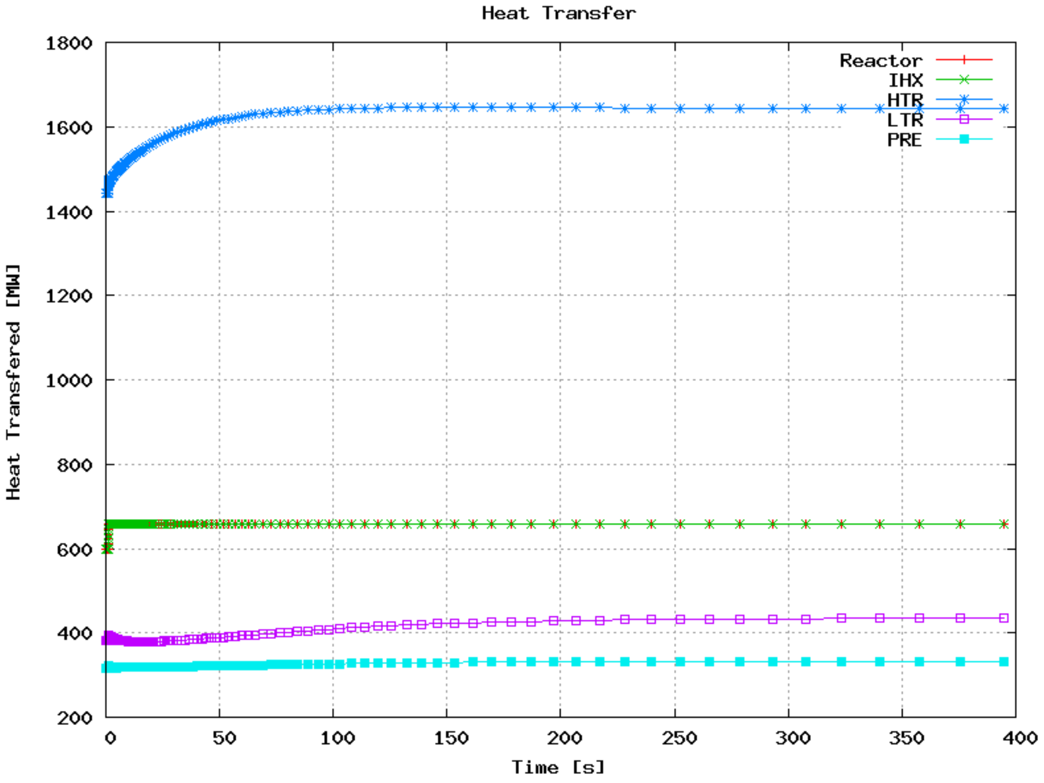


Figure 6-15: +Qrx Long-Term – Heat Transfer

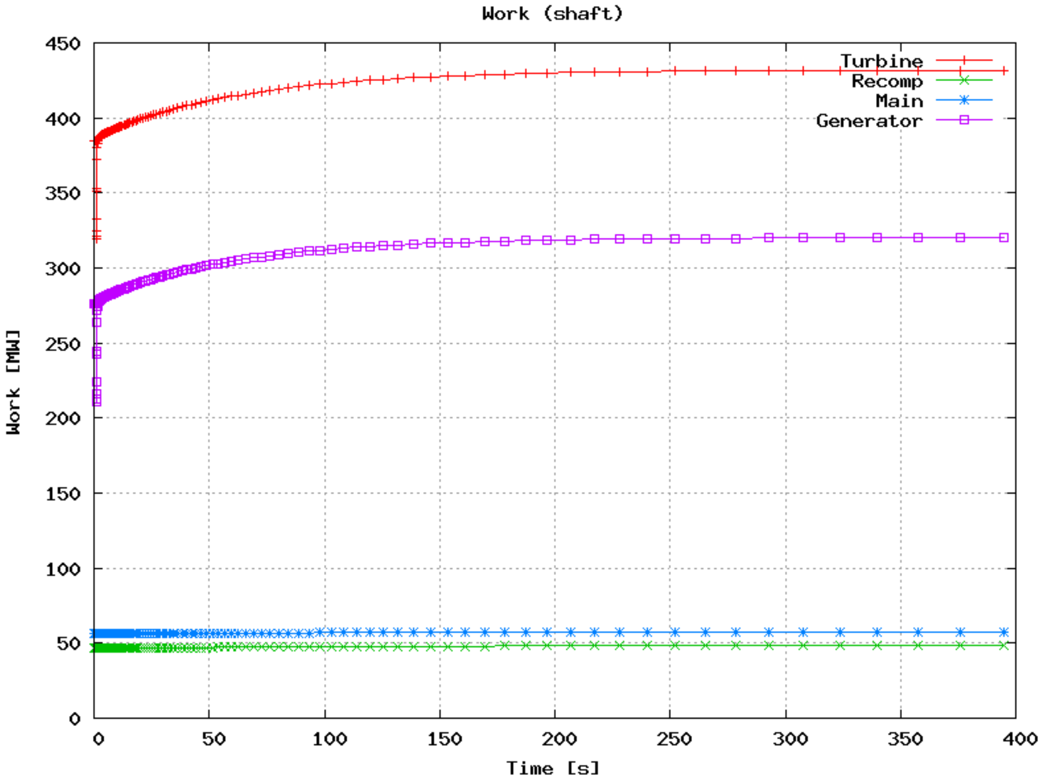


Figure 6-16: +Qrx Long-Term - Work

6.4 Increasing Heat Rejection Transient

This section analyzes the effect of increasing cooling water mass flow rate by 50% over nine seconds. Where the previously analyzed transients have occurred at the high temperature end of the cycle, this transient will show behavior at the lowest temperature of the cycle. Furthermore, where the recompressing compressor previously experienced the bulk of fluid property changes, now the main compressor will experience the bulk of property changes as the S-CO_2 cools towards the critical point.

The transient begins at 1 second when the water mass flow rate is linearly increased until 10 seconds where it is at a 50% higher flow rate as shown in *Figure 6-17*. This figure shows that the overall mass flow rates are nearly constant but the main compressor mass flow rate increases significantly while the recompressing compressor mass flow rate decreases.

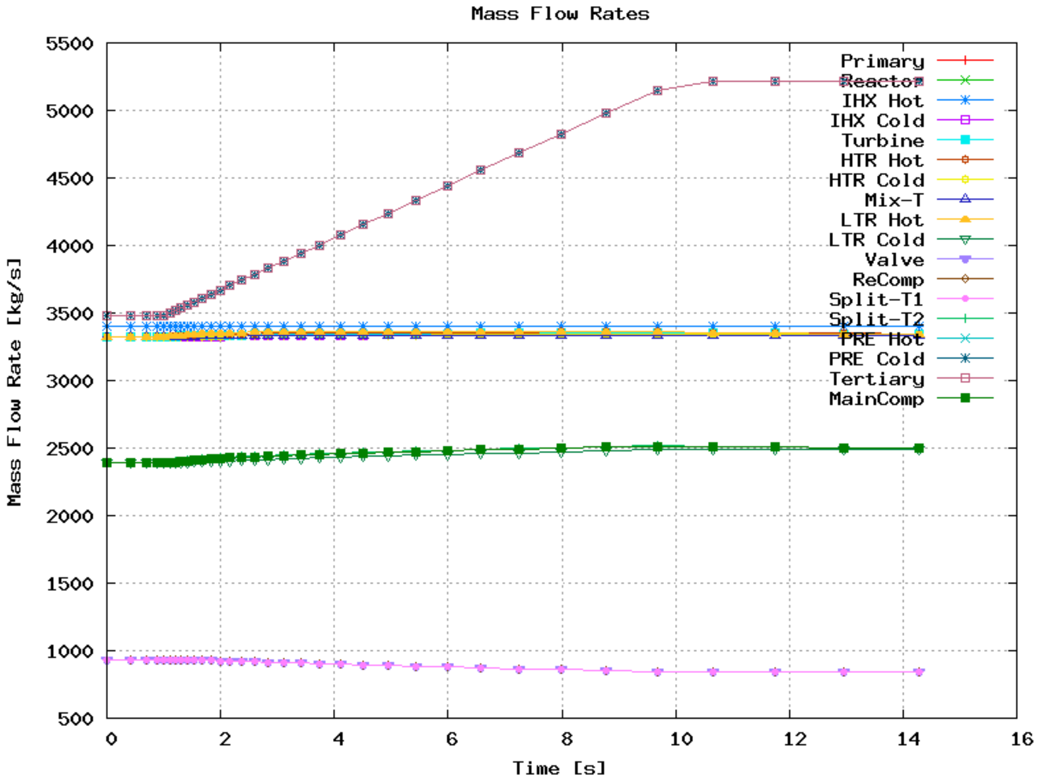


Figure 6-17: +Mpre Short Term - Mass Flow Rates

Since the main compressor receives denser fluid (which increases pressure rise) the compressors have to rebalance to match system pressures by decreasing main compressor pressure rise and increasing recompressing compressor pressure rise. The shape of the radial compressor pressure ratio performance curves, in *Figure 3-3* and *Figure 3-4*, shows that, starting at design values, increasing mass flow rate decreases pressure rise, while decreasing mass flow rate produces a small increase in pressure rise. Therefore, shifting mass flow to the main compressor from the recompressing compressor balances system pressure.

The long-term effect of removing more heat from the system is to decrease system pressures, shown in *Figure 6-18*, decrease the cold part of the cycle temperatures, shown in *Figure 6-19*, and to slightly decrease generator work.

The net work generated decreases because the losses in turbine work outweighs the decrease in compressor work, as seen in *Figure 6-16*. As more heat is removed from the system the turbine inlet temperature decreases and mass flow rate decreases slightly. These effects decrease turbine work slightly more than the compressor work drops with increasing fluid density.

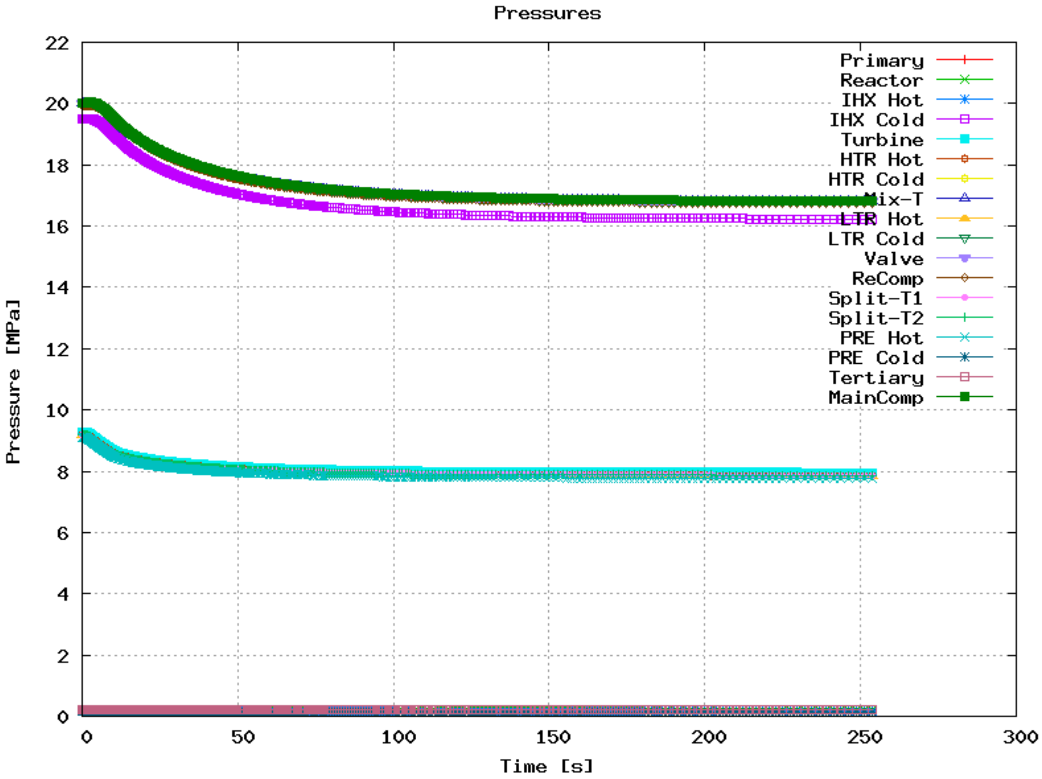


Figure 6-18: +Mpre Long-Term - Pressures

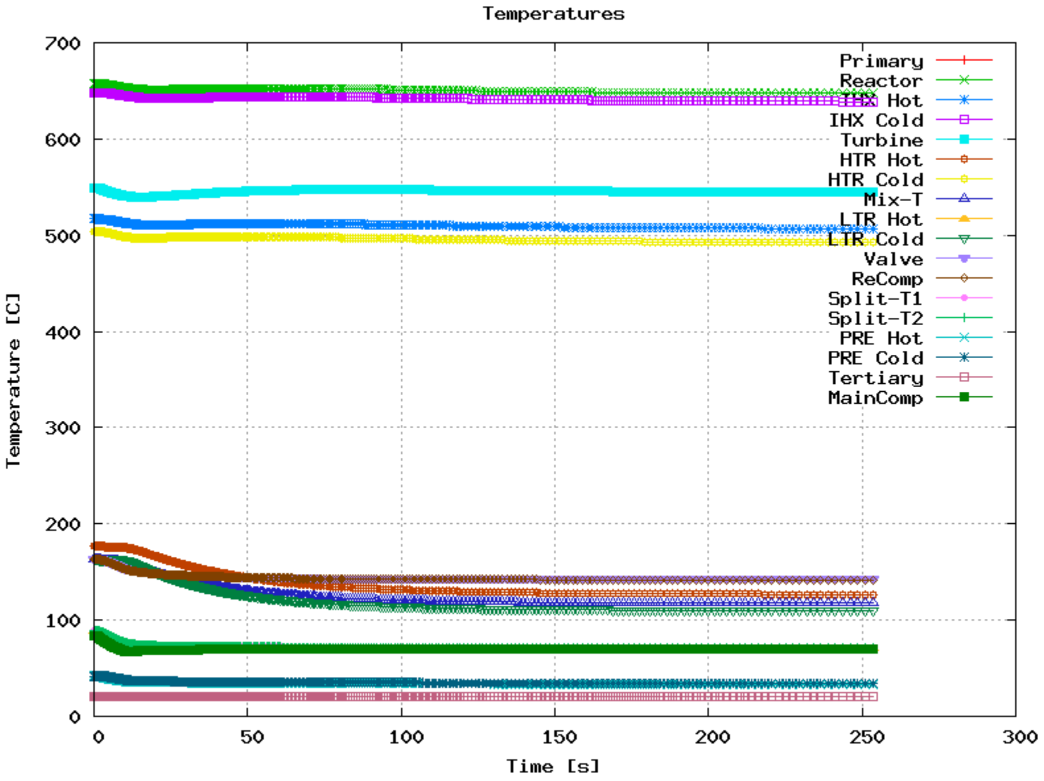


Figure 6-19: +Mpre Long-Term – Temperatures

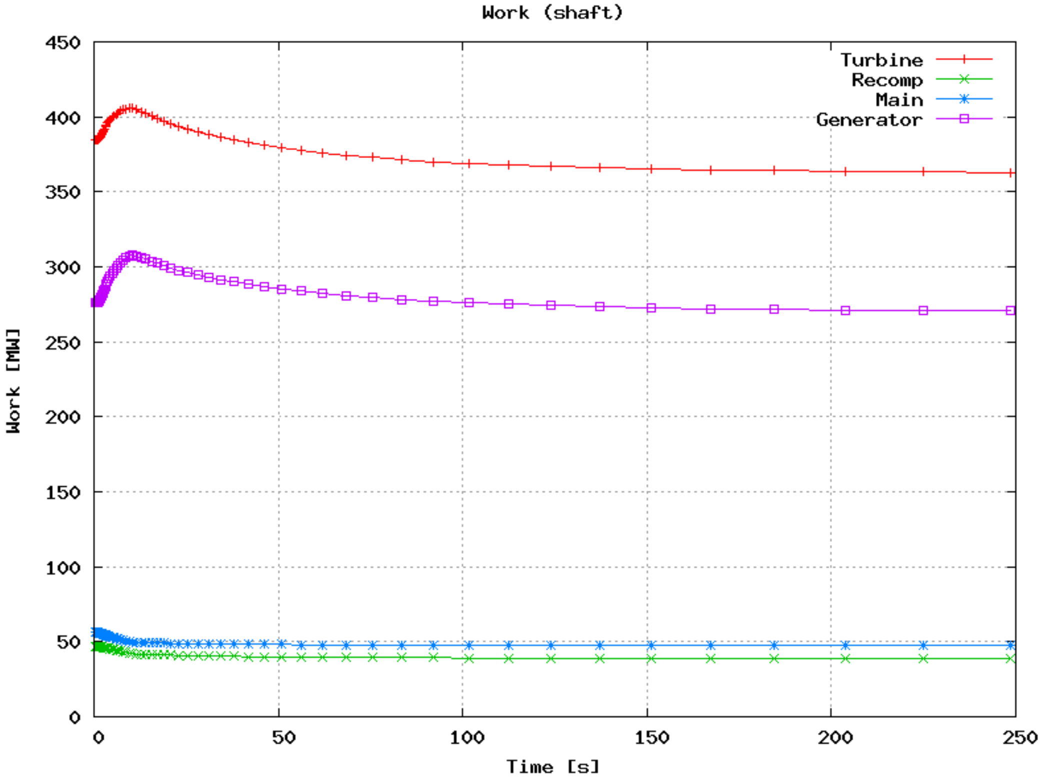


Figure 6-20: +Mpre Long-Term - Work

6.5 Transient Summary

GAS-PASS/CO_2 now has the ability to simulate transients within the S-CO_2 recompression cycle. Significant model work remains, especially regarding control, but initial results confirm expectations and suggest that there is significant potential for the cycle to perform adequately and be controlled appropriately.

In particular, the non-linear S-CO_2 fluid properties produce the unique effect of trading mass flow between the compressors. This presents both a challenge and an opportunity from the control perspective. The challenge stems from keeping both of the compressors within their operating range as they trade mass flow rates – this will be particularly important during inventory control. The opportunity comes from being able to preferentially affect the compressors. When decreasing reactor power, the recompressing compressor will take more mass flow rate. By increasing precooler cooling the main compressor will take more flow rate. This presents two knobs that can be used to control compressor flow rates individually even while they are on a single shaft. It remains to be seen how well these knobs can be used during actual transients.

7 Summary, Conclusions, & Future Work

7.1 Summary

This report has documented the modifications and current status of a workable version of GAS-PASS/CO_2 1.0, as of 9/15/06. Initial transient results appear to validate both the system model and the approach of using general non-linear Newtonian solvers. The code is in an excellent position to accomplish its design objective of rapid transient and control studies.

Four major accomplishments were essential, to meet this goal:

(1) Generating robust root finding algorithms to generate NIST RefProp CO_2 property data to allow the creation of accurate and flexible property data. Tabular indexing using logarithms with S-CO_2 allowed rapid and feasible implementation during transient simulations.

(2) Creating robust, general, and rapid turbomachinery performance estimation based upon shaft speed curves. Using pressure ratio instead of enthalpy change curves and switching to radial compressor curves has greatly improved the ability to simulate transients.

(3) Calculating heat exchanger performance with variable property fluids. Using enthalpy instead of temperature, proper nodal averaging, remembering node solutions between steps, and avoiding unnecessary solution loops have allowed reasonable implementation in a transient code.

(4) Determining how to induce convergence of the "Newtonian" solver scheme inherited from the ideal-gas GAS-PASS/H version with the highly non-linear S-CO_2 recompression cycle. Intelligent Jacobian scaling, turbomachinery performance extrapolation, allowing reduced convergence criteria when necessary, using variable time steps, and allowing failed time steps have created a much more robust code appropriate for use with this cycle.

Finally, early convergence difficulties were traced to the fact that the S-CO_2 PCS is in effect two cycles: a highly compressible hot end coupled to a near-incompressible cold end. These cycles require quite different modeling.

It should be noted that the main text has focused on best-compromise successful methodology. Only in a few instances has attention been paid to approaches which failed to work (i.e. converge) or which succeeded only through excessive time expenditure, and were accordingly rejected.

The code development process was iterative in the sense that the need for fixes was identified by running a variety of uncontrolled quasi-static and uncontrolled transient simulations. One of the biggest changes during this process was replacing the axial turbomachinery performance maps with radial performance maps. During simulation of an increase in precooler water mass flow rate (see *Section 6.4*) the axial compressor mass flow rates ran beyond the known/allowable operating region.

The design point of the axial compressors, in particular the main compressor, are quite close to their maximum flow rate at design shaft speed as shown in *Figure 7-1*. Therefore, a small increase in mass flow rate caused these compressors to move beyond currently predicted operating ranges. Without a control system the increase in water flow transient could not be successfully run using axial compressors. A design decision was made to avoid this very tight mass flow rate constraint by using radial compressors.

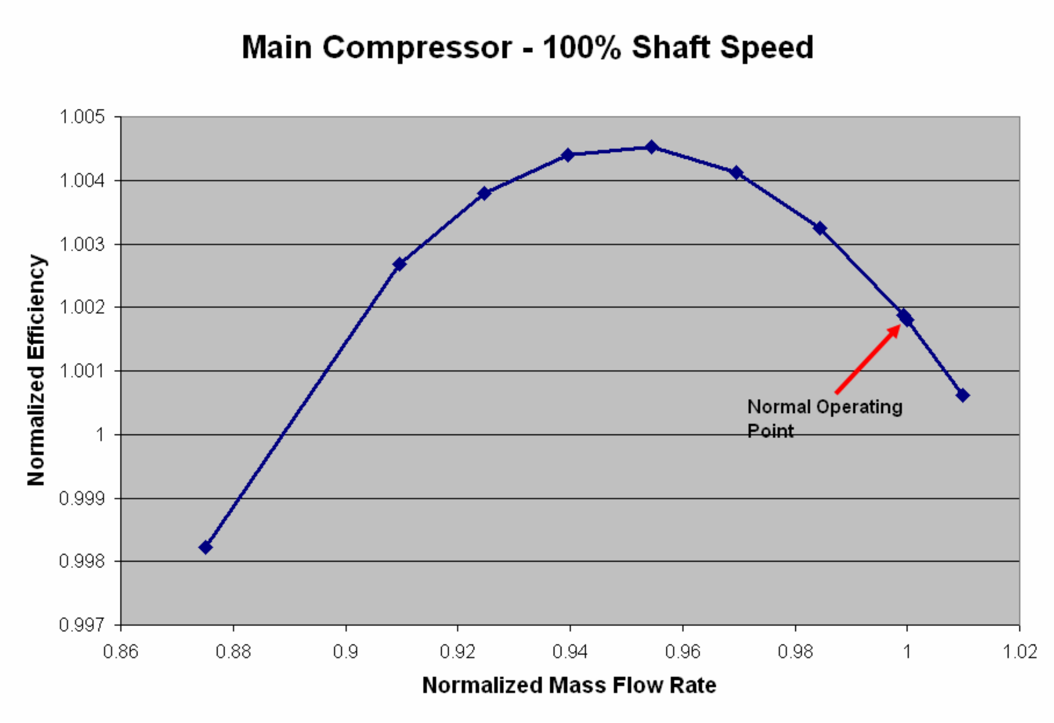


Figure 7-1: Axial Main Compressor 100% Shaft Speed Efficiency

Code version 1.0 has successfully run the transient tests presented in *Chapter 6*. Note that some of them are far more extreme than one would expect in practice: e.g. 10%/sec power ramps as opposed to 5%/min as previously specified in the simulation objectives.

7.2 Past Goals

While considerable transient analysis and control studies remain, GAS-PASS/CO_2 1.0 has met, or rejected after investigation, all but one of the previously submitted future work objectives.

The development needs outlined in the following report have been met or investigated and rejected except for control model development:

> M.J. Driscoll, P. Hejzlar, N. Carstens and Y. Wang, "Simulation of Supercritical CO_2 Brayton Cycle Plants," MIT-GFR-017, Sept. 2004

Numerics

1. Various solvers were investigated (see *Appendix B:* Roots of Nonlinear Systems) and numerous changes were made in the code implementation to allow robust solutions.
2. Current steady state equations are no longer a simple subset of the transient equations, therefore separate coding is warranted.
3. GAS-PASS/CO_2 has been rewritten virtually line by line and the code is virtually completely reconfigurable by input now. Valves are simple components in the input deck.

Models

1. Control system models are currently being investigated.
2. Heat capacities are separate for the structure and fluid in the largest system components, the PCHEs. Structure in other components is still assumed to be isothermal with the coolant but the structure mass can be adjusted to account for this.
3. Multiple stage turbomachinery analysis has been rejected as unnecessarily detailed for this type of code analysis. Considerable uncertainty exists even with the 3D computational design codes and attempting to incorporate this type of analysis into GAS-PASS/CO_2 is not warranted.
4. S-CO_2 properties have been added.
5. All properties are state dependent.
6. Gravity head is not warranted in a quasi-static momentum equation. Once again, this level of detail is not appropriate for GAS-PASS/CO_2 simulations.
7. Pipe friction loss coefficients depend upon Reynolds number.
8. Pressure loss and heat transfer depend upon flow regime.

Input/Output

1. Equation consistency checks have been added.
2. Numerous additional printouts of code models and performance have been created.
3. Plotting has been separated from GAS-PASS/CO_2. Scripts have been written to allow rapid evaluation of many dozens of plots. There is little immediate incentive for combining this within the Fortran coding due to the additional complexity of dealing with graphics when this need is currently met.

The future work objectives from the following report have been completely met.

> N.A. Carstens, P. Hejzlar and M.J. Driscoll, "Description of Supercritical CO_2 Systems Control Model," MIT-GFR-027, Sept. 2005.

1. The code has been updated to completely remove all ideal gas assumptions.
2. We have incorporated and significantly updated and improved turbomachinery off-normal performance mapping.
3. Gas-Pass has been run and debugged using carbon dioxide.
4. An intermediate heat exchanger has been added.
5. More realistic structure temperatures and heat storage have been added.
6. More sophisticated recuperator and precooler models have been developed and incorporated.
7. Sparse Jacobian speedup was investigated and abandoned due to the low yield and high complexity of its implementation.
8. Variable time steps are available for use by the user and, when properly used, can greatly speed up completion with no loss in resolution or accuracy.

7.3 Future Work

Table 7-1 (an updated version of *Table 1-1*) lists the full spectrum of code exercises which could be contemplated in the next phase of work in this area. Model development will continue but only in response to deficiencies identified during transient and control analysis.

Table 7-1: Topics for Future S-CO$_2$ Brayton Indirect Cycle Control Evaluations

1. Move current 42°C GAS-PASS/CO$_2$ model to 32°C model.
2. Control strategies: Bypass/Inventory/Temperature control; including location of bypass valves, adjustment of main/recompression compressor flow split
3. Quasi-steady-state operation between 50% and 100% power, and power transition between states: 10% steps and 5%/min. ramps; effect of variation in cooling water temperature
4. Normal Cycle startup and shutdown; evaluation of PCS use for reactor decay heat removal
5. Rapid Transients
   (a) reactor scram
   (b) primary system loss of flow, loss of coolant
   (c) loss of load
   (d) loss of heat sink
   (e) power cycle blow down
   (f) turbomachinery failure
6. Determine whether control schemes can be devised which will permit use of axial compressors as they are currently envisioned.
7. Selected benchmarking against other codes, such as RELAP

Subtasks (3), (5), (7) and parts of (5) (e.g. uncontrolled transients) can be accomplished in fairly short order using the current version, 1.0, of GAS-PASS/CO$_2$. The remainder, which are of greater interest, and hence higher priority, but which will take longer to implement, require building-in control valves and feedback programming.

Appendix A: GAS-PASS/CO_2 Equations

This appendix documents the conservation equations which are the basis for GAS-PASS/CO_2 1.0.

The sections which follow describe in some detail the roster of differential equations, and their difference representations, which go to make up the GAS-PASS/CO_2 1.0 program version as presently (Sept. 2006) constituted.

This level of detail is of interest mainly to those who may become involved in validating, rewriting or extending code capabilities. Accordingly it has been relegated to an appendix, and only higher level functional descriptions presented in the main text of this report.

A.1 Steady State Equations

In this section we give the set of steady state equations solved when component modules are assembled to represent the plant system shown in *Figure 5-3*. Each of the components in this figure is represented by a module that contains the component equations as shown below. The steady state equations simulate the system with all time-changing elements removed.

All of the equations are ultimately derived from simple conservation laws, however they are expressed in two fashions. About half of the equations are in the standard conservation format which the reader will expect. The other half of the equations ask Gas-Pass to match a variable to a calculated value. This calculated value will be based upon a more detailed (external) calculation which Gas-Pass takes as truth assuming that it satisfies the conservation laws.

The reader will note that several equations are trivial (i.e. $P_{in}=P_{out}$) and thus it might be possible to specify these directly in the code. This temptation (which would speed the calculation) should be avoided to assure numerical consistency. If variables are modified during a calculation then it is vital that other components have not used these variables already. With the general and somewhat complicated nature of the GAS-PASS/CO_2 input deck there is no easy way to assure that this is true, therefore a general rule was made: *variables which are used in other components can only be modified by the numerical solver*.

All of the equations are expressed as they are numerically solved with the right hand side expected to equal 0 and all values are normalized (typically the only term in the denominator) to keep them on the same scale. It is worth noting that equation normalization is a standard practice in the numerical simulation community[42] and is required to successfully simulate this cycle with this solution method. While separately normalization of each equation will change the variable *relative* slopes (or vector magnitude) towards the root, it will not change vector/slope's direction. Furthermore, as long as a root is found (all conservation equations are simultaneously satisfied), then the simulation process to arrive at the root is largely irrelevant. When one tries to converge GAS-PASS/CO_2 without equation normalization, the Jacobian condition number (a measure of how easy a problem is to solve digitally) becomes much smaller than machine precision, 10^{-19}, for the S-CO_2 recompression cycle, therefore the simulation fails. The current transient Jacobian condition number including normalization is 10^{-11}.

Not that inlet values are symbolized by a $_\mathbf{i}$ on the right hand side and outlet values are symbolized by a $_\mathbf{o}$ on the right hand side. The first time a type of equation is shown it will include a brief derivation. Future equations of the same type assume a similar derivation.

There are three fluids in the indirect cycle; the hot fluid symbolized by $^{\mathbf{hot}}$ on the right-hand side which is the sole source of heat addition, the working fluid symbolized by $^{\mathbf{fluid}}$ on the right-hand side, which moves inside the cycle and performs all the work within the system, and the cold fluid symbolized by $^{\mathbf{cold}}$ on the right-hand side, which removes all heat from the cycle.

7.3.1.1 A.1.1 Intermediate Heat Exchanger

The intermediate heat exchanger (IHX) is a Heatric™ printed circuit heat exchanger (PCHE) with liquid sodium on the hot side and CO_2 on the cold. The IHX is the only site of heat addition in this cycle. This PCHE is solved via the detailed 1D modeling code which Gas-Pass treats as a black box producing correctly calculated values.

The calculated values take advantage of the general nature of the Gas-Pass numerical solver by solving for the hot fluid outlet and cold fluid ***inlet*** of this counterflow heat exchanger. If the code were to solve for both fluid outlets we would waste an order of magnitude in calculation time due to the nature of the calculation.

To solve a counterflow heat exchanger numerically one must guess at the outlet values of one stream (say the cold fluid) then progressively march through the heat exchanger using the true hot fluid conditions and guessed cold fluid conditions. Once the other side of the PHCE is reached the calculated cold fluid enthalpy and pressure (from the guess) will not match the true incoming conditions. Thus we must use a loop to correct the fluid pressure guess and a second loop to correct the fluid enthalpy guess. Therefore there are two outer loops to calculate a counterflow heat exchanger's fluid outlets.

Gas-Pass avoids this problem by solving from one side of the PCHE (the hot fluid inlet and cold fluid outlet) only. The Gas-Pass numerical solver will then take care of matching the fluid inlet and outlet values of the various components in the plant. In general this will mean that Gas-Pass will solve for the PCHE cold fluid inlet conditions twice (also from the previous component) and the cold fluid outlet conditions not at all. However, to match the cold fluid inlet values will require varying the cold fluid outlet until the system becomes consistent. All of the subsequent PCHEs are solved in a similar manner. More detail regarding the PCHE solver code may be found in *Chapter 4*.

$$\begin{aligned} h_o^{hot} &= h_{o-calc}^{hot} \\ 0 &= h_o^{hot} - h_{o-calc}^{hot} \\ F(1) &= \frac{h_o^{hot} - h_{o-calc}^{hot}}{h_o^{hot}} \end{aligned} \tag{1}$$

Note that the *F* vector represents the error in each conservation equation. When *F(i)=0* then the conservation equation is solved exactly. When all of the F vectors are simultaneously zero (to user specified tolerance) then the system is at the root – everything is conserved and the solution is achieved.

$$F(2) = \frac{P_o^{hot} - P_{o-calc}^{hot}}{P_o^{hot}} \tag{2}$$

$$F(3) = \frac{\dot{m}_o^{hot} - \dot{m}_o^{hot}}{\dot{m}_o^{hot}} \tag{3}$$

$$F(4) = \frac{h_i^{fluid} - h_{i-calc}^{fluid}}{h_i^{fluid}} \tag{4}$$

$$F(5) = \frac{P_i^{fluid} - P_{i-calc}^{fluid}}{P_i^{fluid}} \tag{5}$$

$$F(6) = \frac{\dot{m}_i^{fluid} - \dot{m}_{i-calc}^{fluid}}{\dot{m}_i^{fluid}} \tag{6}$$

7.3.1.2 A.1.2 Turbine

The turbine is the sole source of work in this cycle. The reader should note that all the turbomachines are on the same shaft, thus turbomachine work is transferred directly. These initial equations will include a brief derivation for the reader's benefit.

The turbine mass conservation equation is not used, to prevent the creation of a singular solution matrix with redundant mass variables (see *Section 5.4.2*). To avoid this problem but still allow the system to be solved, an inventory control equation will be added at the end of this section.

The energy conservation equation is:

$$\dot{E}_{in} = \dot{E}_{out} \tag*{7}$$

$$\dot{m}_i^{fluid} * h_i^{fluid} = \dot{W} + \dot{m}_o^{fluid} * h_o^{fluid}$$

$$0 = \dot{m}_i^{fluid} * h_i^{fluid} - \dot{m}_o^{fluid} * h_o^{fluid} - \dot{W}$$

$$F(7) = \frac{\dot{m}_i^{fluid} * h_i^{fluid} - \dot{m}_o^{fluid} * h_o^{fluid} - \dot{W}}{\dot{m}_i^{fluid} * h_i^{fluid}}$$

The pressure drop equation is:

$$P_{in} - P_{out} = \Delta P \tag*{8}$$

$$P_i^{fluid} - P_{o-Calc}^{fluid} = P_i^{fluid} - P_o^{fluid}$$

$$0 = P_o^{fluid} - P_{o-Calc}^{fluid}$$

$$F(8) = \frac{P_o^{fluid} - P_{o-Calc}^{fluid}}{P_o^{fluid}}$$

Where $P_{o-Calc}^{fluid} = \frac{P_{in}^{fluid}}{PR}$ and pressure ratio, $PR\left(\dot{m}_{in}, \omega\right)$, is interpolated with mass flow rate and shaft speed from the turbine's performance curves.

Similarly, the outlet enthalpy equation is:

$$H_{in} - H_{out} = \Delta H \tag*{9}$$

$$H_i^{fluid} - H_{o-Calc}^{fluid} = H_i^{fluid} - H_o^{fluid}$$

$$0 = H_o^{fluid} - H_{o-Calc}^{fluid}$$

$$F(9) = \frac{H_o^{fluid} - H_{o-Calc}^{fluid}}{H_o^{fluid}}$$

Where $H_{o-Calc}^{fluid} = H_i^{fluid} - \eta * \left(H_i^{fluid} - H_{o-Ideal}^{fluid}\right)$, with efficiency, $\eta\left(\dot{m}_{in}, \omega\right)$, interpolated with mass flow rate and shaft speed from the turbine's performance curves and $H_{o-Ideal}^{fluid}$ is the enthalpy of isentropic expansion to P_o^{fluid}.

7.3.1.3 A.1.3 High Temperature Recuperator

The high temperature recuperator (HTR) is a Heatric™ printed circuit heat exchanger whose hot fluid outputs and cold fluid inputs are calculated from the detailed 1D nodalized PCHE code.

$$F(10) = \frac{h_o^{fluid} - h_{o-calc}^{fluid}}{h_o^{fluid}} \tag*{10}$$

$$F(11) = \frac{P_o^{fluid} - P_{o-calc}^{fluid}}{P_o^{fluid}} \tag{11}$$

$$F(12) = \frac{\dot{m}_o^{fluid} - \dot{m}_{o-calc}^{fluid}}{\dot{m}_o^{fluid}} \tag{12}$$

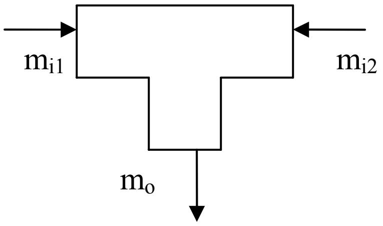
$$F(13) = \frac{h_i^{fluid} - h_{i-calc}^{fluid}}{h_i^{fluid}} \tag{13}$$

$$F(14) = \frac{P_i^{fluid} - P_{i-calc}^{fluid}}{P_i^{fluid}} \tag{14}$$

$$F(15) = \frac{\dot{m}_i^{fluid} - \dot{m}_{i-calc}^{fluid}}{\dot{m}_i^{fluid}} \tag{15}$$

7.3.1.4 A.1.4 Mixing Junction

The mass flow rates at a mixing-T junction, assuming incompressible flow, are related by



Mixing-T Junction Diagram

When applied to the steady state case one has:

$$F(16) = \frac{\dot{m}_{i1}^{fluid} + \dot{m}_{i2}^{fluid} - \dot{m}_o^{fluid}}{\dot{m}_{i1}^{fluid} + \dot{m}_{i2}^{fluid}} \tag{16}$$

The energy conservation equation is:

$$F(17) = \frac{\dot{m}_{i1}^{fluid} * h_{i1}^{fluid} + \dot{m}_{i2}^{fluid} * h_{i2}^{fluid} - \dot{m}_o^{fluid} * h_o^{fluid}}{\dot{m}_{i1}^{fluid} * h_{i1}^{fluid} + \dot{m}_{i2}^{fluid} * h_{i2}^{fluid}} \tag{17}$$

The mixing-T incoming stream1 fluid pressure assumes no pressure drop thus:

$$F(18) = \frac{P_{i1}^{fluid} - P_{o}^{fluid}}{P_{i1}^{fluid}} \tag{18}$$

The mixing-T incoming stream2 fluid pressure assumes no pressure drop thus:

$$F(19) = \frac{P_{i2}^{fluid} - P_{o}^{fluid}}{P_{i2}^{fluid}} \tag{19}$$

Note that this provides one more equation than our general derivation warrants due to the second pressure equation. This formulation is used because it is the best alternative available, considering that:

- One could combine both pressure equations into one but it is undesirable to use a numerical solver to solve for two pressures simultaneously. Furthermore, one would need to prevent the pressure differences between the streams from self canceling, thus some type of sign correction is necessary, which will confuse any solver.
- A second option would match both incoming fluid streams and then put that result directly into the outgoing pressure memory. Unfortunately, this may cause problems because the numerical solver is determining the outgoing pressure, thus it may try to change it before or after the memory substitution. It is somewhat dangerous to directly change the variables the solver is solving for.

This extra pressure equation is easily handled by introducing a valve before one of the mixing-T incoming streams. This valve will be used to match the desired mixing-T incoming stream pressure with the outlet of the previous component (in our case the recompressing compressor) by changing the pressure drop coefficient, *K*. Therefore, the outgoing fluid pressure of the valve is never solved for (*K* is solved in its place) and we are keeping a rigorous treatment of our solution variables.

7.3.1.5 A.1.5 Low Temperature Recuperator

The low temperature recuperator (LTR) is a Heatric™ printed circuit heat exchanger whose hot fluid outputs and cold fluid inputs are calculated from the detailed 1D nodalized PCHE code (see *Chapter 4*).

$$F(20) = \frac{h_{o}^{fluid} - h_{o-calc}^{fluid}}{h_{o}^{fluid}} \tag{20}$$

$$F(21) = \frac{P_{o}^{fluid} - P_{o-calc}^{fluid}}{P_{o}^{fluid}} \tag{21}$$

$$F(22) = \frac{\dot{m}_{o}^{fluid} - \dot{m}_{o-calc}^{fluid}}{\dot{m}_{o}^{fluid}} \tag{22}$$

$$F(23) = \frac{h_i^{fluid} - h_{i-calc}^{fluid}}{h_i^{fluid}} \quad 23$$

$$F(24) = \frac{P_i^{fluid} - P_{i-calc}^{fluid}}{P_i^{fluid}} \quad 24$$

$$F(25) = \frac{\dot{m}_i^{fluid} - \dot{m}_{i-calc}^{fluid}}{\dot{m}_i^{fluid}} \quad 25$$

7.3.1.6 A.1.6 Valve

A valve serves solely to drop a stream's pressure to match other system pressures.

The mass conservation equation is:

$$F(26) = \frac{\dot{m}_i^{fluid} - \dot{m}_o^{fluid}}{\dot{m}_i^{fluid}} \quad 26$$

The energy conservation equation is:

$$F(27) = \frac{h_i^{fluid} - h_o^{fluid}}{h_i^{fluid}} \quad 27$$

The pressure rise equation is:

$$F(28) = \frac{K - \frac{\left(P_{in}^{fluid} - P_o^{fluid}\right) * 2 * \rho_{out}}{M_{out}^2}}{1 + K} \quad 28$$

Where *K* is the valve pressure drop coefficient. The normalization factor has a value of one added to it to prevent strange Jacobian values due to the very small value of K in many cases (on the order of 10^{-4}).

7.3.1.7 A.1.7 Recompressing Compressor

The mass conservation equation is:

$$F(29) = \frac{\dot{m}_i^{fluid} - \dot{m}_o^{fluid}}{\dot{m}_i^{fluid}} \quad 29$$

The energy conservation equation is:

$$F(30) = \frac{\dot{W} + \dot{m}_i^{fluid} * h_i^{fluid} - \dot{m}_o^{fluid} * h_o^{fluid}}{\dot{W} + \dot{m}_i^{fluid} * h_i^{fluid}} \quad 30$$

The pressure rise equation is:

$$F(31) = \frac{P_o^{fluid} - P_{o-Calc}^{fluid}}{P_o^{fluid}} \qquad 31$$

Where $P_{o-Calc}^{fluid} = P_i^{fluid} * PR$ and $PR\left(\dot{m}, \omega\right)$ is interpolated with mass flow rate and shaft speed from the recompressing compressor's performance curves.

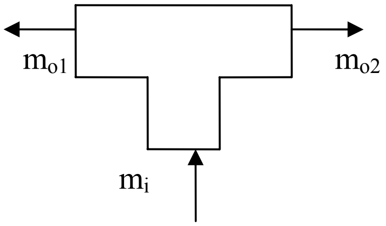
The outlet enthalpy equation is:

$$F(32) = \frac{H_o^{fluid} - H_{o-Calc}^{fluid}}{H_o^{fluid}} \qquad 32$$

Where $H_{o-Calc}^{fluid} = H_i^{fluid} + \frac{\left(H_{o-Ideal}^{fluid} - H_i^{fluid}\right)}{\eta}$ with $\eta\left(\dot{m}, \omega\right)$ is interpolated with mass flow rate and shaft speed from the compressor's performance curves and $H_{o-Ideal}^{fluid}$ is the enthalpy of isentropic compression to P_o^{fluid} .

7.3.1.8 A.1.8 Splitting Junction

A splitting-T junction presents a special case since it has two fluids exiting the component from a single input. This will be handled by treating it as two separate components in practice with the same fluid input. The combined equations are shown below.



Splitting Junction Diagram

The mass conservation for the first outgoing fluid stream is:

$$F(33) = \frac{x * \dot{m}_i^{fluid} - \dot{m}_{o1}^{fluid}}{x * \dot{m}_i^{fluid}} \qquad 33$$

where *x* is the fluid fraction (a boundary condition) which goes to the first junction and is put in the input file for user convenience.

The energy conservation equation for the first stream is:

$$F(34) = \frac{h_i^{fluid} - h_{o1}^{fluid}}{h_i^{fluid}} \qquad 34$$

The first stream assumes no pressure drop thus:

$$F(35) = \frac{P_i^{fluid} - P_{o1}^{fluid}}{P_i^{fluid}} \qquad 35$$

The mass conservation for the second outgoing fluid stream is:

$$F(36) = \frac{\dot{m}_i^{fluid} - \dot{m}_{o1}^{fluid} - \dot{m}_{o2}^{fluid}}{\dot{m}_i^{fluid}} \qquad 36$$

The energy conservation equation for the second stream is:

$$F(37) = \frac{h_i^{fluid} - h_{o2}^{fluid}}{h_i^{fluid}} \qquad 37$$

The second stream assumes no pressure drop thus:

$$F(38) = \frac{P_i^{fluid} - P_{o2}^{fluid}}{P_i^{fluid}} \qquad 38$$

This formulation allows the calculation of the six outgoing fluid stream properties (two mass flow rates, two enthalpies, and two pressures) with six equations while simultaneously forcing the matching of outgoing pressures and enthalpies which would physically happen at a splitting-T and is therefore physically and numerically consistent. Furthermore, by using a flow split fraction the user can control the outgoing fluid mass flow rates at will.

7.3.1.9 A.1.9 Precooler

The precooler (PRE) is a Heatric™ printed circuit heat exchanger whose hot fluid outputs and cold fluid inputs are calculated from the detailed 1D nodalized PCHE code (see *Chapter 4*).

$$F(39) = \frac{h_o^{fluid} - h_{o-calc}^{fluid}}{h_o^{fluid}} \qquad 39$$

$$F(40) = \frac{P_o^{fluid} - P_{o-calc}^{fluid}}{P_o^{fluid}} \qquad 40$$

$$F(41) = \frac{\dot{m}_o^{fluid} - \dot{m}_{o-calc}^{fluid}}{\dot{m}_o^{fluid}} \qquad 41$$

$$F(42)=\frac{h_i^{cold}-h_{i-calc}^{cold}}{h_i^{cold}} \quad 42$$

$$F(43)=\frac{P_i^{cold}-P_{i-calc}^{cold}}{P_i^{cold}} \quad 43$$

$$F(44)=\frac{\dot{m}_i^{cold}-\dot{m}_{i-calc}^{cold}}{\dot{m}_i^{cold}} \quad 44$$

7.3.1.10 A.1.10 Main Compressor

The mass conservation equation is:

$$F(45)=\frac{\dot{m}_i^{fluid}-\dot{m}_o^{fluid}}{\dot{m}_i^{fluid}} \quad 45$$

The energy conservation equation is:

$$F(46)=\frac{\dot{W}+\dot{m}_i^{fluid}*h_i^{fluid}-\dot{m}_o^{fluid}*h_o^{fluid}}{\dot{W}+\dot{m}_i^{fluid}*h_i^{fluid}} \quad 46$$

The pressure rise equation is:

$$F(47)=\frac{P_o^{fluid}-P_{o-Calc}^{fluid}}{P_o^{fluid}} \quad 47$$

Where $P_{o-Calc}^{fluid}=P_i^{Work}*PR$ and $PR\left(\dot{m},\omega\right)$ is interpolated with mass flow rate and shaft speed from the main compressor's performance curves.

The outlet enthalpy equation is:

$$F(48)=\frac{H_o^{fluid}-H_{o-Calc}^{fluid}}{H_o^{fluid}} \quad 48$$

Where $H_{o-Calc}^{fluid}=H_i^{fluid}+\frac{\left(H_{o-Ideal}^{fluid}-H_i^{fluid}\right)}{\eta}$ with $\eta\left(\dot{m},\omega\right)$ is interpolated with mass flow rate and shaft speed from the compressor's performance curves and $H_{o-Ideal}^{fluid}$ is the enthalpy of isentropic compression to P_o^{fluid}.

7.3.1.11 A.1.11 Inventory

The mass of CO_2 is computed within each component in the cycle and summed to give a total fluid inventory. The fluid density is calculated at the outlet enthalpy, and pressure and the volumes are inputs.

$$F(49) = \frac{TotMass - \sum_{k} Vol^{k} * \rho^{\mathrm{k}}_{\mathrm{o}(h,P)}}{TotMass} \qquad 49$$

7.3.1.12 A.1.12 Shaft

Conservation of mechanical energy for the turbomachinery shaft gives

$$F(50) = \frac{\sum_{k} \dot{W}^{k}_{Gen} - \sum_{k} \dot{W}^{k}_{\mathrm{Re}\,m}}{\sum_{k} \dot{W}^{k}_{Gen}} \qquad 50$$

Note that the removal term includes $W_{drag} = K_{drag}\omega^{2}$ where ω is the rotational speed of the common shaft and W_{gen}. In effect, this equation allows the calculation of the energy supplied (or demanded) by the generator.

A.2 Time-Difference Equations

The time-difference equations are identical to the steady state equations with the addition of mass and energy storage terms. Only equations with the additional terms will be shown below, otherwise they are the same as in the steady state. The left hand superscript ^{n} refers to the current time step while $^{n-1}$ is from the previous time step. The first time an equation is shown a brief derivation will be shown. Future equations of the same type assume a similar derivation.

7.3.1.13 A.2.1 Intermediate Heat Exchanger

The transient PCHEs are solved in a similar manner to the steady state with the addition of mass storage and energy storage terms. More detail regarding the transient PCHE solver code may be found in *Chapter 4*.

7.3.1.14 A.2.2 Turbine

The turbine is the sole source of work in this cycle. The reader should note that all the turbomachines are on the same shaft; thus turbomachine work is transferred directly. These first equations will show a brief derivation for the readers benefit.

The energy conservation equation is:

TR-7

$$\frac{dE}{dt} = {}^{n}\dot{E}_{in} - {}^{n}\dot{E}_{out}$$

$$\frac{{}^{n}E - {}^{n-1}E}{\Delta t} = {}^{n}\dot{m}_{i}^{fluid} * {}^{n}h_{i}^{fluid} - {}^{n}\dot{m}_{o}^{fluid} * {}^{n}h_{o}^{fluid} - {}^{n}\dot{W}$$

$$0 = {}^{n}\dot{m}_{i}^{fluid} * {}^{n}h_{i}^{fluid} - {}^{n}\dot{m}_{o}^{fluid} * {}^{n}h_{o}^{fluid} - {}^{n}\dot{W} - \frac{{}^{n}E - {}^{n-1}E}{\Delta t}$$

$$F(7) = \frac{{}^{n}\dot{m}_{i}^{fluid} * {}^{n}h_{i}^{fluid} - {}^{n}\dot{m}_{o}^{fluid} * {}^{n}h_{o}^{fluid} - {}^{n}\dot{W} - \frac{{}^{n}E - {}^{n-1}E}{\Delta t}}{{}^{n}\dot{m}_{i}^{fluid} * {}^{n}h_{i}^{fluid}}$$

Note that the energy stored in a component is the sum of the internal energy of the fluid and the internal energy of the structure. At any time step:

$${}^{n}E = {}^{n}E^{fluid} + {}^{n}E^{struct}$$

$${}^{n}E^{fluid} = {}^{n}M^{fluid} * {}^{n}U^{fluid}$$

$${}^{n}E^{struct} = {}^{n}M^{struct} * {}^{n}C^{struct} * {}^{n}T^{struct}$$

The pressure drop equation does not change.

The outlet enthalpy equation does not change.

Once again there is no mass conservation equation to prevent a singular matrix. Instead inventory control is used.

7.3.1.15 A.2.3 High Temperature Recuperator

The high temperature recuperator (HTR) is a Heatric™ printed circuit heat exchanger whose transient outputs are calculated from the detailed 1D transient nodalized PCHE code (see *Chapter 4*).

7.3.1.16 A.2.4 Low Temperature Recuperator

The low temperature recuperator (LTR) is a Heatric™ printed circuit heat exchanger whose transient outputs are calculated from the detailed 1D transient nodalized PCHE code (see *Chapter 4*).

7.3.1.17 A.2.5 Mixing Junction

The mixing-T equations do not change since it has too small a volume to make a significant difference in transient calculations but would increase the numerical difficulty by forcing the convergence of a very quickly changing component.

7.3.1.18 A.2.6 Main Compressor

The main compressor equations do not change. Gas-Pass was tested with main compressor storage enabled and it was discovered that the fluid in this region is nearly incompressible and transients occurred very quickly, requiring very small time steps. Although energy storage could be modeled, the main compressor is a relatively small component and is therefore neglected in initial analysis. It is expected that neglecting storage here will make a negligible difference in expected transients but this will be more carefully quantified at a later date.

7.3.1.19 A.2.7 Recompressing Compressor

The recompressing compressor equations do not change for the same reasons listed under the main compressor.

7.3.1.20 A.2.8 Splitting Junction

The splitting-junction equations do not change since it has a very small volume.

7.3.1.21 A.2.9 Precooler

The precooler (PRE) is a Heatric™ printed circuit heat exchanger whose transient outputs are calculated from the detailed 1D nodalized PCHE code (see *Chapter 4*).

7.3.1.22 A.2.10 Inventory

The mass inventory equation does not change.

7.3.1.23 A.2.11 Shaft

The transient shaft equation includes shaft inertia.

$$0={}^{n}\omega-{}^{n-1}\omega-\frac{\sum_{k}\dot{W}^{k}_{Gen}-\sum_{k}\dot{W}^{k}_{\mathrm{Re}\,m}}{{}^{n}\omega I}\Delta t \qquad \text{TR-50}$$

$$F(50)=\frac{{}^{n}\omega-{}^{n-1}\omega-\dfrac{\sum_{k}\dot{W}^{k}_{Gen}-\sum_{k}\dot{W}^{k}_{\mathrm{Re}\,m}}{{}^{n}\omega I}\Delta t}{{}^{n}\omega}$$

Appendix B: Roots of Nonlinear Systems

This appendix will briefly survey the available techniques for finding roots of a non-linear system of equations that were reviewed as possible candidates for GAS-PASS/CO_2. Finding the roots of systems of non-linear equations tends to revolve around a few key techniques.

B.1 Mathematical Methods

The information in this section is taken directly from the following sources:

- Fiares and Burden, *Numerical Methods*, PWS Publishing Company, Boston, 1993.
    - Chapter 10 provides a good general overview and goes into a little depth on the Newton, Broyden, and steepest descent methods.
- Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Society for Industrial and Applied Mathematics, Philadelphia, 1995.
    - This text offers 5 chapters detailing Newton's, inexact Newton, Broyden's, and globally convergent methods.
- Iserless and Powell, *The State of the Art in Numerical Analysis*, Conference proceedings, Clarendon Press, Oxford, 1987.
    - Chapter 9 presents a paper by Schnabel and Frank that analyses solving systems of nonlinear equations by tensor methods.

7.3.1.24 B.1.1 Good Guess (Linear Leaps)

Newton's method and the Secant method are the most common root finding techniques. These methods give rapid convergence at the cost of requiring a good initial guess.

Newton's method approximates n^2 partial derivatives (the Jacobian) and solves an n by n linear system at each step. The method requires n^2+n function evaluations at each step and $O(n^3)$ operations for solution. This method can be written to minimize round off error and has quadratic convergence.

The secant method (Broyden's method is the key implementation) reduces the amount of computation to n function evaluations at each step and $O(n^2)$ operations for solution. This method is known as a least-change secant update and is a quasi-Newton. It replaces the Jacobian with an approximation matrix that is updated with each iteration. Broyden's method gives up quadratic convergence for super linear convergence and generally does not correct for round off with successive iterations. In most cases it is considered to be faster than Newton especially for dense Jacobians or functions that are expensive to evaluate.

7.3.1.25 B.1.2 Global Convergence (Curve Crawlers)

The steepest-descent method offers only linear convergence but it is global in nature. It takes the gradient of the system to find the steepest path a local minimum.

Homotopy and continuation, line search and the Armijo rule all appear to work on a similar principle. If a linear leap method failed then it was probably because the step it took towards the solution was too large. These algorithms backtrack upon solution failure and decrease the step size until the new function evaluation is better than the last evaluation. This will sacrifice the quadratic convergence of Newton's method for a globally convergent algorithm. Typically, these algorithms spend a lot of time on the first iterations then quadratically converge to the solution. It may be important to add a safeguard parameter to avoid becoming stuck in local minima.

These methods are often used as a hybrid technique to get close enough to use a good guess method but they will fail on ill-conditioned or singular systems.

7.3.1.26 B.1.3 Tensor Methods

Tensor methods are a class of general purpose methods for solving systems of non-linear equations. They are especially intended to efficiently solve problems where the Jacobian matrix at the solution is singular or ill-conditioned, while remaining at least as efficient as standard methods on nonsingular problems. They are based upon a quadratic (possibly higher) model of the nonlinear function. The model has a simple second term which provides more information than standard linear methods without significantly increasing computational cost. Two types of methods exist just as in the linear model: Jacobian and Secant methods.

B.2 Computational Implementations

In general, Newtonian, Broyden, and curve crawler solvers are readily (and commercially) available but more advanced techniques like Krylov tenser solvers are still in development.

7.3.1.27 B.2.1 The IMSL Solution Method

GAS-PASS/CO_2 currently uses the solver included in the IMSL programming package. The non-linear system root finding methods in the IMSL libraries are based on two subroutines HYBRDI and HYBRDJ contained in MINPACK, a public domain package. Both methods use the Levenberg-Marquardt method, which is a weighted average of Newton's method and the Steepest-Descent method. The weight is biased towards the Steepest-Descent method until convergence is detected, at which time the weight is shifted to the more rapidly convergent Newton's method.

7.3.1.28 B.2.2 SUNDIALS

Lawrence Livermore has embarked upon an ambitious project, SUNDIALS, to write a “Suite of Nonlinear and Differential/Algebraic Equation Solvers”. Of interest is NKSOL a Newton-Krylov solver for nonlinear algebraic systems. This code is written in C (at the time C could support parallelism better than Fortran – 90/95 didn’t yet exist) from the original Fortran featured in NKSOL by Taylor and Hindmarsh in 1998.

7.3.1.29 B.2.3 TRILINOS

Sandia has embarked upon the Trilinos project which includes a code, NOX, Nonlinear Object-Oriented Solutions. This solves nonlinear systems of equations using Krylov-Tensor methods (I think with preconditioning) and is written in C.

Appendix C: References

DRAFT

[1] M.J. Driscoll, P. Hejzlar, N. Carstens and Y. Wang, “Simulation of Supercritical CO_2 Brayton Cycle Plants”

[2] N.A. Carstens, P. Hejzlar and M.J. Driscoll, “Description of Supercritical CO_2 Systems Control Model,” MIT-GFR-027, Sept. 2005

[3] V. Dostal, M.J. Driscoll and P. Hejzlar, “A Supercritical Carbon Dioxide Cycle for Next Generation Nuclear Reactors,” MIT-ANP-TR-100, March 2004

[4] V. Dostal, P. Hejzlar and M.J. Driscoll, “High-Performance Supercritical Carbon Dioxide Cycle for Next-Generation Nuclear Reactors,” *Nuclear Technology*, Vol. 154, No. 3, June 2006

[5] J.P. Gibbs, P. Hejzlar and M.J. Driscoll, “Applicability of Supercritical CO_2 Power Conversion Systems to GEN IV Reactors,” MIT-GFR-037, Sept. 2006.

[6] Y. Gong et al., “”Analysis of Radial Compressor Options for Supercritical CO_2 Power Conversion Cycles,” MIT-GFR-034, June 2006

[7] NIST REFPROP CODE

[8] Eric Lemmon, *REFPROP computer code help*, NIST Standard Database 23 Version 7.0.

[9] Ronald Aungier, *Axial-Flow Compressors; A Strategy for Aerodynamic Design and Analysis*, ASME Press, 2003, page 37.

[10] Ronald Aungier, *Axial-Flow Compressors; A Strategy for Aerodynamic Design and Analysis*, ASME Press, 2003, page 32.

[11] WIKI REFERENCE TO BRENT”S METHOD

[12] GONGS RADIAL COMPRESSOR REPORT

[13] JEFF”S REPORT AGAIN

[14] Richard Vilim, *International Nuclear Energy Research Initiative #2001-002-F; Development of Gen IV Advanced Gas-Cooled Reactor with Hardened/Fast Neutron Spectrum; Passive Full Power Safety*, Argonne National Laboratory, January 2004, Doc. GFR-016.

[15] Richard Vilim, *International Nuclear Energy Research Initiative #2001-002-F: Development of Gen IV Advanced Gas-Cooled Reactor with Hardened/Fast Neutron Spectrum; Passive Full Power Safety*, Argonne National Laboratory.

[16] JEFF”S TURBO REPORT

[17] ISAAC”S CORREALTION REPORT

[18] Vilim et. al, GAS-PASS/H: A Simulation Code for Gas Reactor Plant Systems, September 2003, Argonne National Laboratory.

[19] Ronald Aungier, *Axial Flow Compressor; A Strategy for Aerodynamic Design and Analysis*, ASME Press, 2003.

[20] Japikse and Baines, *Introduction to Turbomachinery*, Concepts ETI, Inc. and Oxford University Press, 1994.

[21] Numerical Recipes in Fortran 77,

[22] Ibid.,

[23] Ibid.,

[24] Vaclav Dostal, *A Supercritical Carbon Dioxide Cycle for Next Generation Nuclear Reactors*, MIT PhD Thesis, January 2004.

[25] Ibid.

[26] Ishizuka et. al., *Thermal Hydraulic Characteristics of a Printed Circuit Heat Exchanger in a Supercritical CO_2 Loop*, NURETH-11, 2005.

[27] Wang, Jianfeng and Hihar, Eiji, *Study on carbon dioxide gas cooler heat transfer process under supercritical pressures*, International Journal of Energy Research, 2002.

[28] Petukhov BS. 1974. *Turbulent heat transfer in tubes with variable fluid properties*. Heat Exchangers – Design and Theory Sourcebook, Afan NH, Schlunder EU (eds), Chapter 11. McGraw-Hill Book Company: new York, 240-251.

[29] Gnielinski, V. *New Equations for heat and mass transfer in turbulent pipe and channel flow*, International Chemical Engineering, April 1976

[30] Wang, Jianfeng and Hihar, Eiji, *Study on carbon dioxide gas cooler heat transfer process under supercritical pressures*, International Journal of Energy Research, 2002.

[31] Olsen, Douglas, *Heat Transfer of Supercritical Carbon Dioxide Flowing in a Cooled Horizontal Tube*, NISTIR 6496, May 2000.

[32] Hesselgreaves, J., E., *Compact Heat Exchangers, Selection, Design, and Operation*, 1st. Edition, Pergamon, 2001.

[33] Williams W., Hejzlar P, Driscoll M. J., Lee W-J., Saha P., *Analysis of a Convection Loop for GFR Post-LOCA Decay Heat Removal from a Block-Type Core*, MIT-ANP-TR-095, March, 2003.

[34] Williams W., Hejzlar P, Driscoll M. J., Lee W-J., Saha P., *Analysis of a Convection Loop for GFR Post-LOCA Decay Heat Removal from a Block-Type Core*, MIT-ANP-TR-095, March, 2003.

[35] Hejzlar, Pavel, *Printed Circuit Heat Exchangers with Zigzag Channel,* internal MIT paper, January, 2006.

[36] Ibid.

[37] Williams W., Hejzlar P, Driscoll M. J., Lee W-J., Saha P., *Analysis of a Convection Loop for GFR Post-LOCA Decay Heat Removal from a Block-Type Core*, MIT-ANP-TR-095, March, 2003.

[38] NIST RefProp Code, Version 7.0.

[39] Ibid, page 256.

[40] IMSL SOVLER PACKAGE

[41] Ibid.

[42] RICK'S COMMENTS ON EQUATION NORMALIZATION