



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

TRANSIENT MODELING OF TWO-PHASE STEAM IN A PIPING SYSTEM

by

Damien A. Stonhill

June 2023

Thesis Advisor:
Co-Advisor:
Second Reader:

Walter C. Smith
Anthony J. Gannon
Garth V. Hobson

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 2023	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE TRANSIENT MODELING OF TWO-PHASE STEAM IN A PIPING SYSTEM			5. FUNDING NUMBERS RMLDC	
6. AUTHOR(S) Damien A. Stonhill				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) NAVSEA08T			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) The characteristics of two-phase steam flows in piping systems under transient conditions are a critical area of research for the Department of the Navy. Simulating these cases is necessary to improve future reactor designs' safety, efficiency, and performance. Current steam piping systems are designed to prevent moisture accumulation. Such designs are often unnecessarily conservative, increasing complexity and cost while lowering efficiency. This project aimed to build on previous steady-state simulations, using TRAC/RELAP Advanced Computational Engine (TRACE) to predict the moisture distribution in transient two-phase high-quality steam flows. The transient conditions modeled reflected the fluctuations in pressure and moisture content that result from changes in steam plant operation. The performance of TRACE was evaluated for these scenarios and compared with open literature data. These simulations will serve as baseline models for further research concerning transient conditions in steam piping systems.				
14. SUBJECT TERMS TRAC/RELAP Advanced Computational Engine, TRACE, transient, two-phase, pipe, steam, flow, fluid			15. NUMBER OF PAGES 271	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

TRANSIENT MODELING OF TWO-PHASE STEAM IN A PIPING SYSTEM

Damien A. Stonhill
Ensign, United States Navy
BS, United States Naval Academy, 2022

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
June 2023**

Approved by: Walter C. Smith
Advisor

Anthony J. Gannon
Co-Advisor

Garth V. Hobson
Second Reader

Brian S. Bingham
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The characteristics of two-phase steam flows in piping systems under transient conditions are a critical area of research for the Department of the Navy. Simulating these cases is necessary to improve future reactor designs' safety, efficiency, and performance. Current steam piping systems are designed to prevent moisture accumulation. Such designs are often unnecessarily conservative, increasing complexity and cost while lowering efficiency. This project aimed to build on previous steady-state simulations, using TRAC/RELAP Advanced Computational Engine (TRACE) to predict the moisture distribution in transient two-phase high-quality steam flows. The transient conditions modeled reflected the fluctuations in pressure and moisture content that result from changes in steam plant operation. The performance of TRACE was evaluated for these scenarios and compared with open literature data. These simulations will serve as baseline models for further research concerning transient conditions in steam piping systems.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	MOTIVATION	1
B.	OBJECTIVE	2
C.	THESIS OVERVIEW	3
II.	LITERATURE REVIEW	5
III.	TRACE IMPLEMENTATION	17
A.	FLUID FIELD MODELS.....	17
B.	HEAT TRANSFER MODELS	19
IV.	STEADY-STATE TRACE MODELING	23
A.	MODEL DEVELOPMENT	23
B.	SIMULATION RESULTS	28
V.	PRESSURE TRANSIENT TRACE MODELING.....	37
A.	MODEL DEVELOPMENT	38
B.	SIMULATION RESULTS	41
VI.	VAPOR MASS FRACTION TRANSIENT TRACE MODELING.....	55
A.	MODEL DEVELOPMENT	55
B.	SIMULATION RESULTS	57
VII.	PRESSURE OSCILLATION CASE STUDY	59
A.	MODEL DEVELOPMENT	60
B.	SIMULATION RESULTS AND COMPARISON.....	63
1.	Pressure Comparison.....	64
2.	Fluid Temperature Comparison.....	66
3.	Wall Temperature Comparison.....	69
4.	Heat Transfer Coefficient Comparison	71
5.	Mass Flux Comparison.....	74
6.	Vapor Mass Fraction Comparison	75
VIII.	CONCLUSIONS AND RECOMMENDATIONS.....	79
	APPENDIX A. MATLAB SCRIPT FOR VAPOR MASS FRACTION TO	
	VOLUME FRACTION CONVERSION	81

APPENDIX B. STEADY-STATE PHASE MASS DISTRIBUTION FOR VARYING INLET QUALITY	83
APPENDIX C. MOODY DIAGRAM FOR ROUGHNESS IN FULLY TURBULENT PIPE FLOW	85
APPENDIX D. PIPE BEND RESISTANCE COEFFICIENT DOCUMENTATION	87
APPENDIX E. MATLAB FUNCTIONS FOR IMPORTING TRACE DATA WITHOUT USE OF INTERFACE.....	91
APPENDIX F. MATLAB SCRIPT FOR STEADY-STATE TRACE DATA PROCESSING.....	95
APPENDIX G. MATLAB SCRIPT FOR GUO ET AL. OPEN LITERATURE DATA PROCESSING	111
APPENDIX H. MATLAB SCRIPT FOR TRACE INTERFACING AND TRANSIENT CASE GRAPH AND VIDEO GENERATION.....	117
APPENDIX I. MATLAB SCRIPT FOR LITERATURE COMPARISON WITH TRACE DATA	143
APPENDIX J. TRACE INPUTS FOR STEADY STATE MODEL	151
APPENDIX K. TRACE INPUTS FOR 98% INLET QUALITY PRESSURE TRANSIENT MODEL	165
APPENDIX L. TRACE INPUTS FOR VAPOR MASS FRACTION TRANSIENT MODEL	189
APPENDIX M. TRACE INPUTS FOR GUO ET AL. PRESSURE OSCILLATION COMPARISON MODEL.....	213
APPENDIX N. TRACE USER INTERFACE MODEL DEVELOPMENT EXAMPLE.....	237
LIST OF REFERENCES.....	247
INITIAL DISTRIBUTION LIST	249

LIST OF FIGURES

Figure 1.	Two-phase flow regimes. Source: [3].	5
Figure 2.	Two-phase flow regime transition. Source: [4].	6
Figure 3.	Schematic of vertical T-junction. Source: [5].	7
Figure 4.	Schematic of 90° round pipe bend. Source: [6].	8
Figure 5.	Liquid and gas diversion ratios for different T-junction orientations. Source: [7].	8
Figure 6.	Multi-phase water diagrams. Source: [9].	9
Figure 7.	Boiling curve regions as a function of heat flux and temperature gradient. Source: [12].	11
Figure 8.	Axelsson steam pipe geometry. Source: [13].	12
Figure 9.	Guo et al. experimental apparatus. Source: [14].	13
Figure 10.	Experimental two-phase flow pressure oscillation results. Source: [14].	14
Figure 11.	TRACE boiling curve example. Adapted from [15].	21
Figure 12.	Steady-state TRACE pipe model	24
Figure 13.	Vortices induced by circular pipe bends. Adapted from [18].	28
Figure 14.	Pressure drop across pipe length for inlet flow quality range	29
Figure 15.	Pressure change rate along pipe length for inlet flow quality range	30
Figure 16.	Normalized cell interface flow quality along pipe length for inlet flow quality range	31
Figure 17.	Vapor temperature along pipe length for inlet flow quality range	32
Figure 18.	Liquid temperature along pipe length for inlet flow quality range	33
Figure 19.	Cell vapor mass fraction along pipe length for inlet flow quality range	34
Figure 20.	Cell vapor mass fraction along pipe length for 98% inlet flow quality with and without bend K factors	35

Figure 21.	Total fluid mass in pipe for inlet flow quality range	36
Figure 22.	Transient pressure input.....	37
Figure 23.	Pressure transient break control signal	39
Figure 24.	Pressure along pipe for transient pressure with 98% inlet flow quality	42
Figure 25.	Temperatures along pipe length from 600 to 630 seconds for transient pressure with 98% inlet flow quality	44
Figure 26.	Temperatures along pipe length from 635 to 660 seconds for transient pressure with 98% inlet flow quality	45
Figure 27.	Inner wall heat flux along pipe length for transient pressure with 98% inlet flow quality.....	47
Figure 28.	Wall heat transfer coefficient along pipe length for transient pressure with 98% inlet flow quality.....	48
Figure 29.	Flow quality along pipe length for transient pressure with 98% inlet flow quality	49
Figure 30.	Vapor mass flow rate along pipe length for transient pressure with 98% inlet flow quality.....	50
Figure 31.	Liquid mass flow rate along pipe length for transient pressure with 98% inlet flow quality.....	51
Figure 32.	Vapor mass fraction along pipe length for transient pressure with 98% inlet flow quality.....	52
Figure 33.	Total pipe fluid mass along pipe length for transient pressure with 98% inlet flow quality.....	53
Figure 34.	Pipe wall influence on outlet flow quality over pressure transient.....	54
Figure 35.	Vapor volume fraction input signal	55
Figure 36.	Control signal components for transient vapor volume fraction	56
Figure 37.	Pipe pressure over time in cell increments of 10	57
Figure 38.	Guo et al. digitized two-phase flow pressure oscillation data. Adapted from [14].....	59
Figure 39.	Guo et al. TRACE comparison pressure control signal.....	62

Figure 40.	Transient pressure input for Guo et al. comparison.....	63
Figure 41.	TRACE initial model and Guo et al. pressure comparison.....	64
Figure 42.	TRACE smooth pipe model and Guo et al. pressure comparison	65
Figure 43.	TRACE smooth 99.96% flow quality pipe model and Guo et al. pressure comparison.....	66
Figure 44.	TRACE initial model and Guo et al. fluid temperature comparison	67
Figure 45.	TRACE smooth pipe model and Guo et al. fluid temperature comparison.....	68
Figure 46.	TRACE smooth 99.96% flow quality pipe model and Guo et al. fluid temperature comparison.....	69
Figure 47.	TRACE initial model and Guo et al. wall temperature comparison	70
Figure 48.	TRACE smooth pipe model and Guo et al. wall temperature comparison.....	71
Figure 49.	TRACE initial model and Guo et al. heat transfer coefficient comparison.....	72
Figure 50.	TRACE smooth pipe model and Guo et al. heat transfer coefficient comparison.....	73
Figure 51.	TRACE smooth 99.96% flow quality pipe model and Guo et al. heat transfer coefficient comparison.....	74
Figure 52.	TRACE and Guo et al. mass flux comparison.....	75
Figure 53.	TRACE initial model and Guo et al. vapor mass fraction comparison	76
Figure 54.	TRACE smooth pipe model and Guo et al. vapor mass fraction comparison.....	77
Figure 55.	TRACE smooth 99.96% flow quality pipe model and Guo et al. vapor mass fraction comparison	78

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	Guo et al. experimental flow parameters. Source: [14].	14
Table 2.	TRACE boiling models. Adapted from [15].....	20
Table 3.	TRACE fluid parameters for steady-state pipe model. Adapted from [13].	24
Table 4.	Superheated water vapor properties.....	25
Table 5.	Vapor volume fraction to vapor quality conversion	25
Table 6.	Idel’chik pipe bend loss coefficient values	27
Table 7.	Lumped capacitance calculation results.....	41
Table 8.	TRACE input parameters for Guo et al. comparison. Adapted from [14].	61

THIS PAGE INTENTIONALLY LEFT BLANK

NOMENCLATURE

a	area
A	dimensionless friction parameter
B	dimensionless friction parameter
Bi	Biot number
C	dimensionless friction parameter
c_p	constant pressure specific heat
c_v	constant volume specific heat
d	diameter
e	internal energy
g	gravitational acceleration
G	mass flux
h	heat transfer coefficient
h'	enthalpy
h'_{lv}	latent heat of evaporation
K	pipe bend loss coefficient
k	thermal conductivity
K_{bend}	pipe bend resistance coefficient
K_{fric}	pipe bend friction coefficient
l	length
m	mass
M	momentum transfer rate
N	minimum number of nodes
Nu	Nusselt number
P	power
p	pressure
Pr	Prandtl number
Q	heat
q	heat flux

q_d	power deposited directly into fluid
R	Reynolds stress tensor
r°	pipe bend radius of curvature
Re	Reynolds number
T	temperature
t	time, thickness
u	velocity
W	work
X	quality

GREEK

α	vapor volume fraction
Γ	phase change rate
δ°	pipe bend angle
ϵ	absolute roughness
λ	referred length friction coefficient
μ	dynamic viscosity
ρ	density

SUBSCRIPTS

d	characteristic diameter
eq	equilibrium
i	interfacial
l	liquid
n	wall node
sat	saturation
v	water vapor
w	wall
h	hydraulic

ACRONYMS

ASTM	American Society for Testing and Materials
CFD	computational fluid dynamics
CHF	critical heat flux
IAPWS IF-97	International Association for the Properties of Water and Steam Industrial Formulation 97
NIST	National Institute of Standards and Technology
TRACE	TRAC/RELAP Advanced Computational Engine

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I am grateful to Dr. Walter C. Smith, my primary advisor, for his guidance and expertise in this research. I would also like to thank my secondary advisor and second reader, Dr. Anthony J. Gannon and Dr. Garth V. Hobson. Thanks also to Dr. Mark Holowach for his valuable insights on TRACE. Lastly, my love goes to my family and friends who made this whole effort possible.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

The U.S. Navy powers all aircraft carriers and submarines with pressurized water nuclear reactors. As a result, ensuring these systems' safety, efficiency, and performance is vital to national security. One area of focus is the design optimization of the secondary loop, improving steam quality and minimizing losses associated with conservatism in moisture removal systems. Predicting moisture distribution is especially important under transient conditions, where undesirable effects that may cause damage and decreased performance are most prevalent. Computational fluid dynamics (CFD) codes can be used to predict the behavior of complex two-phase steam flows under transient conditions. Such modeling allows for extensive iteration of future designs in naval reactor systems.

Minimal pressure losses and high steam quality are critically important design parameters for performance in naval nuclear propulsion systems. Conversely, collection of liquid water in piping systems and steam turbines has many detrimental effects. The steam traps and moisture separators required to remove moisture not only reduce efficiency by way of pressure losses, but also add complexity and cost. The increased viscous effects of liquid water as well as energy lost due to phase change and interfacial drag further contribute to efficiency losses. If not adequately removed, moisture will also damage the components in a steam cycle. Rapid heat transfer between cold water and hot pipe surfaces can lead to cracking and warping. This heat transfer also results in water rapidly flashing to steam, subjecting the pipes to undesirably high pressures. Entrained water droplets in a two-phase flow lead to increased erosion in the pipes as well as turbines. Turbines have an especially low moisture tolerance, as any water droplets in the steam will impinge on the blades, leading to pitting and reduced system life. While reactors often operate at steady-state, transient steam characteristics resulting from variations in power output, as well as start-up, and shut-down are common. Accidents and malfunctions, though much less common, also produce transient effects in the steam flow. The varying flow conditions caused by these transients require additional measures to ensure adequate steam quality. Therefore, it is necessary to analyze relevant transients in two-phase steam flow so that

excessive design conservatism can be avoided. The benefits and limitations of using codes, such as TRAC/RELAP Advanced Computational Engine (TRACE), to model transient two-phase flow must be evaluated so that future steam piping may be more effectively designed.

B. OBJECTIVE

The primary objective of this study was to use the Nuclear Regulatory Commission's (NRC) system analysis code, TRACE, to develop a model of transient two-phase flow through a piping system. The inlet flow was comprised of dry steam and entrained liquid water droplets for various flow qualities. Two transient cases were of interest, one concerning pressure drops and the other involving decreasing inlet flow quality. In order to properly model the physical features of these transient cases, the steady-state model was refined first. This objective's primary focus areas included modeling superheated steam, entrained saturated water droplets, turbulence, volume fraction, and flow quality. Equally important was modeling the piping system geometry and the associated loss factors due to roughness, bends, and elevation changes.

For transient cases, physical properties for heat transfer at the interface of the pipe wall and fluid flow, as well as within the pipe wall, were modeled. Because of the anticipated latent heat flow between the pipe wall and liquid film, evaluating the different boiling models used in TRACE was also necessary. Once the transient models were developed, they were compared with data from open literature.

This research was a continuation of previous work, refining steady-state models and building to transient case analysis. This effort will be foundational for future analysis and design of naval nuclear propulsion systems. Moisture management in secondary steam loops is essential to avoid damaging pipes and turbines. The transient cases in this research simulate increases in reactor power output, leading to pressure and inlet flow quality drops and introducing unwanted moisture within the system. The model developed in this effort may be used to simulate moisture distribution within steam piping systems as influenced by geometric features, latent heat, and changing flow conditions. This work will increase

understanding of design tools for optimizing moisture removal features in future naval nuclear propulsion systems.

C. THESIS OVERVIEW

This work focused on developing a model to interrogate transient two-phase steam flow in a piping system using TRACE. X Steam, a MATLAB function used to generate steam and water properties based on the International Association for Properties of Water and Steam Industrial Formulation 1997 (IAPWS IF-97), was utilized for data analysis and calculations outside of TRACE [1]. Additional thermophysical fluid properties were taken from the National Institute of Science and Technology (NIST) data tables [2]. The transient parameters of interest were pressure and vapor mass fraction. This effort studied cases with these transients in a setting applicable to steam pipe flow in a nuclear reactor secondary loop prior to expansion in a high-quality turbine. Chapter II discusses the two-phase flow theory in piping systems, latent pipe wall heat transfer, and previous experimental work. Chapter III describes important TRACE solver features and the implementation of TRACE code for solving relevant flow physics in this effort. Chapter IV describes the development of TRACE models for the steady-state portion of the transient cases of interest and the associated results. Chapter V discusses the development of the pressure transient flow model and the following results. Chapter VI describes the development of the vapor mass fraction transient model and the simulation results. Chapter VII describes the case study developed to evaluate TRACE performance. This chapter describes the experimental setup and results, the development of a corresponding model in TRACE, and a comparison of the experimental and simulated results. Chapter VIII summarizes this effort, followed by conclusions and recommendations for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

Extensive data has been collected for steady-state two-phase fluid flows through piping systems. These efforts have led to the development of correlations and empirical models used by CFD codes to improve the distribution of phases within the geometry of a fluid flow. However, research and model development involving transient fluctuations in pressure and flow quality for two-phase pipe flow is limited. Therefore, this effort must include a literature review of the physics and fluid mechanics relevant to the individual features developed for the steady-state and transient CFD models. Understanding applicable concepts and previous work will result in higher fidelity simulations that predict real fluid flow behavior more accurately.

Proper modeling of steam pipe systems requires an understanding of multiphase flow. Such flows consist of two or more fluid phases, such as liquid and vapor, moving together within the same geometry. Each phase has independent thermodynamic properties and accounts for a portion of the total volume. Common two-phase flow patterns are shown in Figure 1.

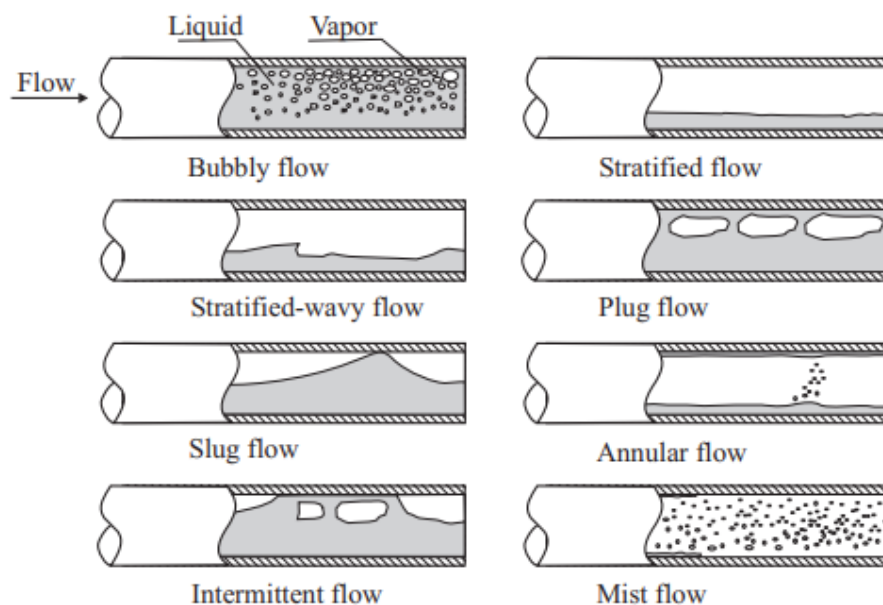


Figure 1. Two-phase flow regimes. Source: [3].

The flow regime of each phase relative to the other is highly dependent on factors such as mass fraction and Reynolds number. Figure 2 illustrates how two-phase flow patterns may change as a function of increasing vapor mass fraction.

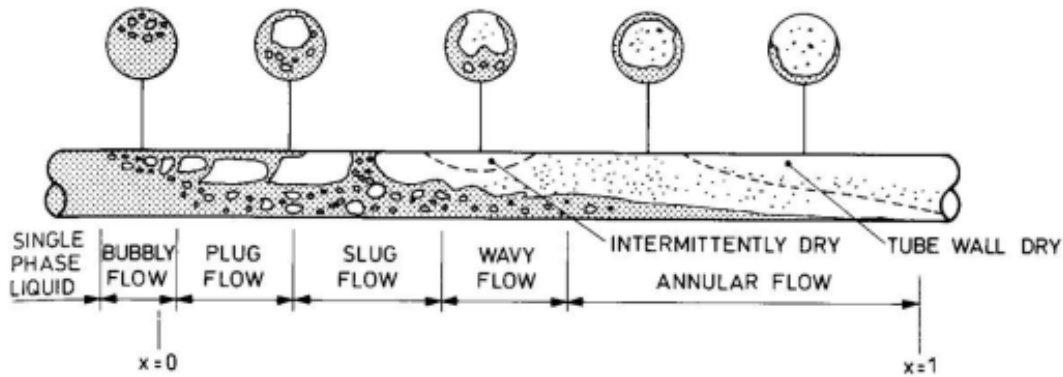


Figure 2. Two-phase flow regime transition. Source: [4].

These flow patterns have tangible effects on pressure drops and heat transfer coefficients. These, in turn, are critical for modeling phase changes through mechanisms such as condensation, de-entrainment, and evaporation. Because this research concerned high-quality steam, annular flow patterns with entrained liquid droplets were most applicable. When analyzing such flows, defining the nomenclature used to describe the ratios of vapor to liquid is crucial. Vapor volume fraction, or void fraction, describes the fraction of vapor volume within a total discrete volume, while vapor mass fraction describes the fraction of vapor mass within a total discrete mass. The term vapor mass fraction is used interchangeably with instantaneous quality. Flow quality is the ratio of the vapor mass flow rate to the total mass flow rate through a finite area. Flow quality and vapor mass fraction may often be equal but are very different parameters in cases with stagnant or recirculating liquid flow within a geometry. Finally, quality can be calculated using an energy balance. This is known as the thermal equilibrium quality. Assuming thermal equilibrium and a saturated liquid, the vapor produced is proportional to the heat added to the system, the fluid mass, and the latent heat of evaporation. Equation (1) illustrates this relationship, where

$$X_{eq} = \frac{Q}{m \cdot h_{lv}}. \quad (1)$$

This method of calculating quality has limitations. This effort is concerned with superheated vapor with entrained saturated liquid droplets. Heat added to such a flow would not necessarily result in droplet evaporation, increasing the vapor superheat. As a result, the thermal equilibrium quality and quality calculated from mass or mass flow rate often have different values.

The geometric features in the piping system, such as the T-junction and bend shown in Figure 3 and Figure 4, affect liquid droplet de-entrainment in two-phase annular flow. Azzopardi and Whalley [5] determined that for high gas volume fraction two-phase air-water annular flows, a large amount of liquid phase was diverted into the branch of a T-junction relative to the amount of liquid in the primary flow.

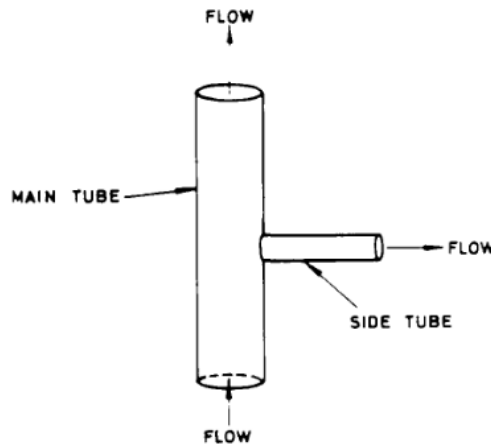


Figure 3. Schematic of vertical T-junction. Source: [5].

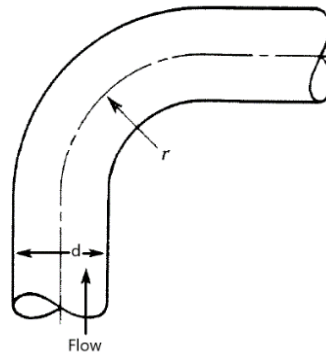


Figure 4. Schematic of 90° round pipe bend. Source: [6].

Azzopardi [5] determined that the fluid flow regime prior to the T-junction greatly impacted the liquid mass fraction diverted at the junction. The annular liquid film moving on the pipe wall diverted most of the liquid into the side tube. Water droplets entrained in the airflow did not readily collect in the side tube of the junction due to high inertial effects governing their movement. Several studies determined that the orientation of a T-junction pipe and branch has a significant impact on the ratio of gas to liquid taken from the main flow [7]. Figure 5 summarizes the results of these studies.

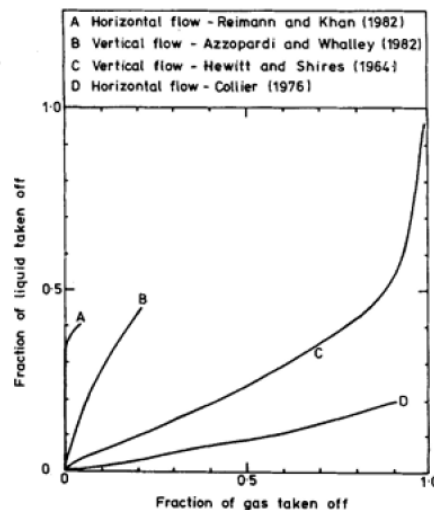


Figure 5. Liquid and gas diversion ratios for different T-junction orientations. Source: [7].

Little work has been conducted on two-phase annular flow behavior at pipe bends. The high temperature, pressure, velocity, and vapor mass fraction flows of interest to this study create a harsh environment that is very difficult to interrogate experimentally. Recent work concerning two-phase flow through pipe bends was conducted under steady-state conditions that are not directly pertinent to this effort. A study by Al-Tameemi [8] investigated flow regime changes at horizontal to vertical 90° pipe bends for low Reynolds number two-phase air and water flows. Al-Tameemi observed that the pipe bend induced vortices in the flow, causing the annular flow to transition to a swirling annular flow. Further, Al-Tameemi found that the bend did not affect the upstream and downstream pressure gradients and that its magnitude was overpredicted by frictional loss models that did not account for flow regime changes.

Understanding the distribution of liquid and vapor water phases relative to pressure, volume, and temperature is critical to qualitatively predicting and evaluating the effects of transient conditions on steam piping systems. Figure 6 offers a concise visual representation of water phase distribution. This effort was not concerned with the solid state but rather with the parabolic region where two-phase wet steam exists. During a transient case, when a property such as pressure is changed, the phase of the fluid will change along a different line of constant pressure.

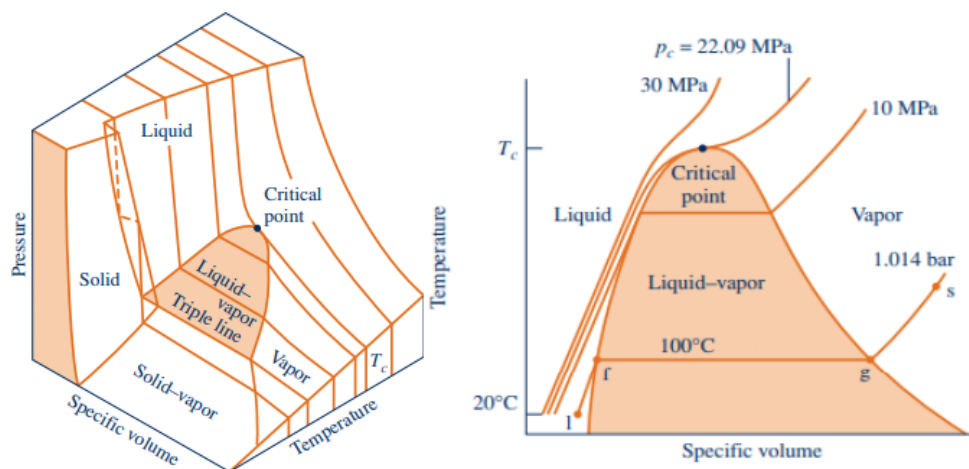


Figure 6. Multi-phase water diagrams. Source: [9].

Thermal interactions between the pipe material and the fluid phases heavily influence transient cases in steam piping systems. Latent heat transfer from the metal results in a radial temperature gradient in the pipe wall. The pipe wall may be modeled using a finite number of concentric rings, or nodes, each with a discrete temperature. This simplification is known as the lumped capacitance parameter. This assumption sees each node as a body at a uniform internal temperature with no gradient. Any heat transfer between the body and its surroundings must also be one-dimensional in nature [10]. Ito and Kuramae [11] concluded that axial heat transfer in a pipe wall is negligible in transient cases where the pipe's latent heat transfer dominates. Therefore, it is reasonable to assume one-dimensional heat transfer in adiabatic, unheated piping systems. When conducting a lumped capacitance analysis approach, pipe wall Biot number, defined in (2) as

$$Bi = \frac{h_w}{k_w} t_w < 0.1, \quad (2)$$

must be considered. The Biot number is a function of the convective heat transfer coefficient at the interface of the wall and the internal fluid, h_w , the thermal conductivity of the pipe material, k_w , and the characteristic thickness of the thermal body, t_w . A Biot number much lower than one indicates that the conduction resistance within the thermal mass is substantially lower than the convection resistance to the surroundings. As the Biot number decreases, the temperature gradient within the thermal mass may be neglected in favor of a spatially uniform temperature. The error introduced by using the lumped capacitance simplification is negligible when the Biot number is less than 0.1, as previously indicated in (2).

Under the transient conditions of interest to this effort, heat transfer between the pipe wall and fluid flow is expected to result in phase change due to boiling and condensation. The heat transfer coefficient for forced convection boiling in a pipe is a function of many parameters and is difficult to estimate. In particular, the type of fluid flow regime, flow quality, and mean fluid velocity hold a heavy influence [10]. Figure 7 illustrates flow regime trends for boiling at different heat fluxes and temperature gradients between the inner pipe wall and fluid flow.

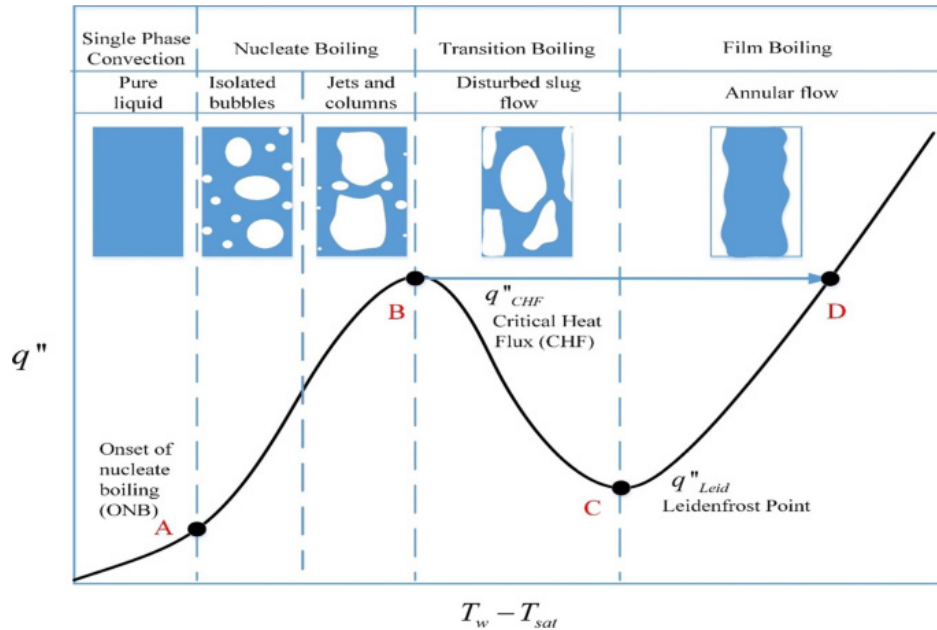


Figure 7. Boiling curve regions as a function of heat flux and temperature gradient. Source: [12].

The pipe wall heat transfer coefficient is a function of heat flux and wall superheat and varies with fluid flow and boiling regimes. The fluid at the inlet of the pipe in Figure 7 has a flow quality of zero. Before boiling, heat transfer is only due to forced liquid convection. The onset of nucleate boiling occurs when isolated vapor bubbles form at the surface of the pipe and detach to form bubbly flow. Heat transfer increases in this first boiling region due to increased mixing at the inner pipe wall [10]. As the temperature gradient and heat flux increase, the bubble formation rate increases until vapor pockets at the surface of the pipe impede heat transfer. This drying out of the pipe wall leads to a peak in heat transfer known as critical heat flux [10]. After this point, the vapor layer around the pipe wall grows, and the surface temperature increases until an insulating vapor cushion levitates the remaining liquid [12]. For high-quality vapor flows beyond the Leidenfrost point, liquid droplets are entrained in the vapor and primarily driven by the inertia of the flow. The boiling regions for inverted annular flows and mist flows are most applicable to the high flow qualities simulated in this effort.

Previous work conducted by Axelsson [13] concerning steady-state two-phase flow Ansys CFX and TRACE simulations in a piping system was foundational to this effort.

Axelsson developed pipe geometry and steady-state fluid flow parameters representative of steady steam plant characteristics. The pipe geometry is displayed in Figure 8.

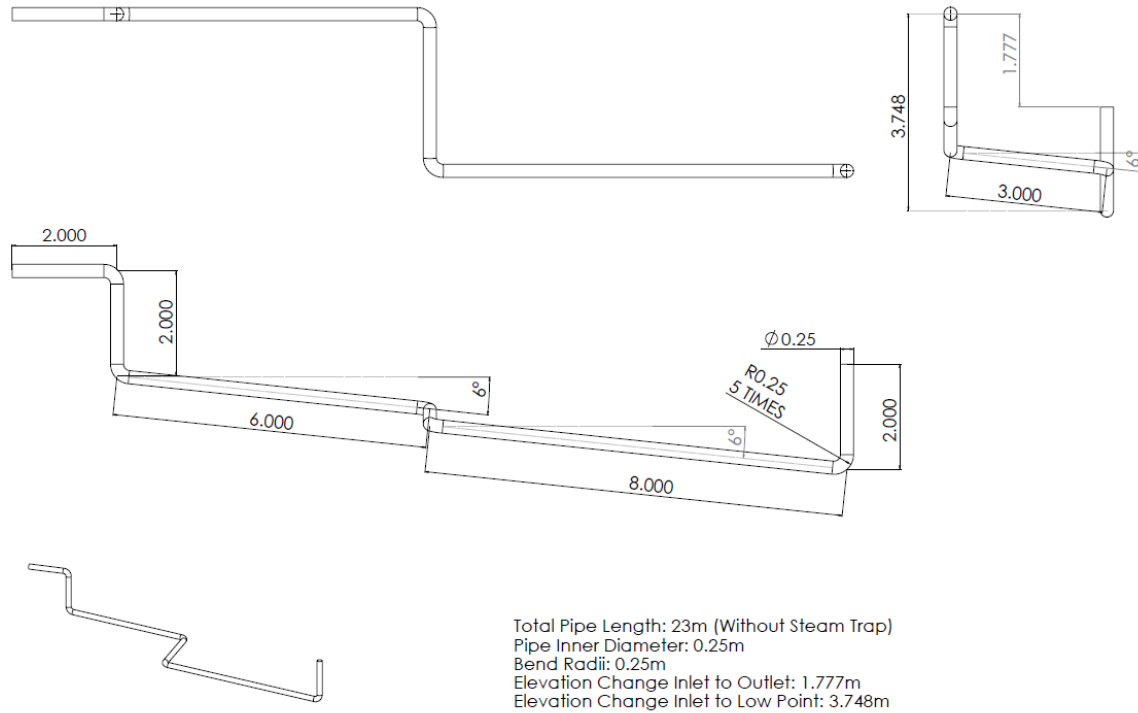


Figure 8. Axelsson steam pipe geometry. Source: [13].

Axelsson [13] also simulated fluid flow in a modified pipe geometry with two steam traps prior to the third and fifth bends. The inlet fluid flow for both geometries was defined as high-speed, high-pressure saturated steam with entrained water droplets at 85-99% inlet flow quality. Both pipe geometries were analyzed in Ansys CFX. The geometry with steam traps was not modeled in TRACE due to the one-dimensional limitations of the code. Axelsson evaluated pressure drop and distribution of moisture across the pipe geometry for both geometries and codes. Axelsson observed that the simulated pressure drop across the piping system was approximately 50.3% higher in TRACE than in CFX, with pipe roughness neglected in both models [13]. Axelsson also modeled several steam condensation case studies conducted by Borishanskiy, et al. in TRACE. The code performed reasonably well, deviating from experimental data by 12.1 to 23.4%. This effort

used the pipe geometry, steady-state fluid parameters, and TRACE results as a baseline for model development.

Guo et al. [14] collected radial heat transfer data from fluid to pipe wall in a study evaluating both single and two-phase steam flow under transient pressure drop oscillations. The experimental setup is shown in Figure 9.

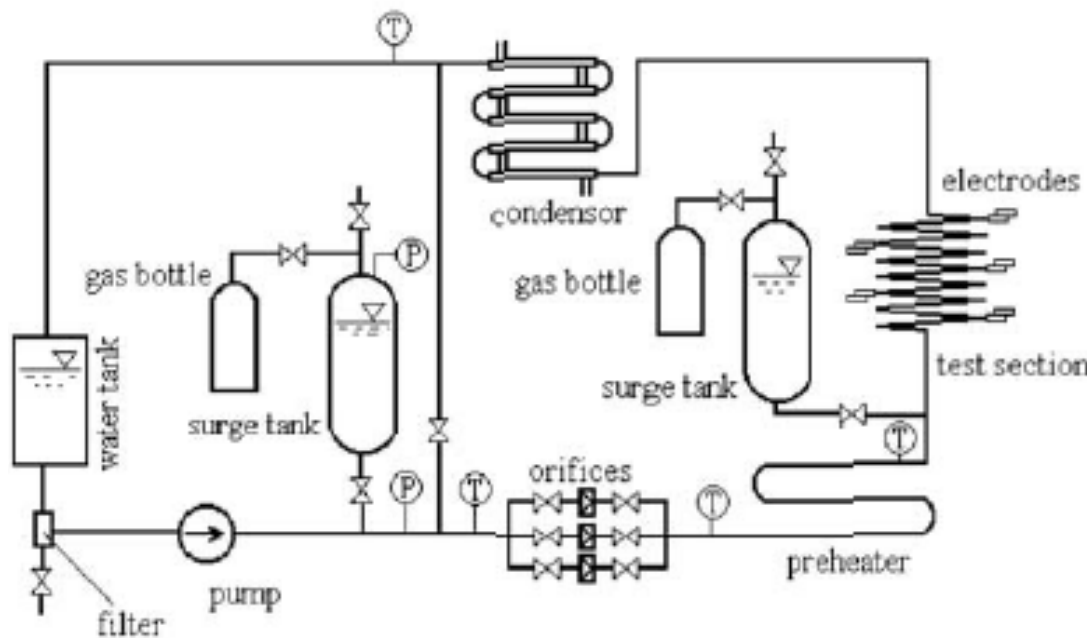


Figure 9. Guo et al. experimental apparatus. Source: [14].

The fluid temperature was regulated using a preheater positioned before the inlet of the test section, 100 kW of ohmic heating into the test section tubing, and a condenser downstream from the test section outlet. For two-phase data collection, a surge tank located just before the inlet of the test section controlled the pressure oscillations. Data was taken from a test section comprised of insulated horizontal helical tubing. Relevant flow parameters from the test section are displayed in Table 1.

Table 1. Guo et al. experimental flow parameters. Source: [14].

Experiment Parameter	Value
Hydraulic Diameter, d_h (mm)	15
Wall Thickness, t_w (mm)	2
Inlet Pressure, p (MPa)	1.5-3.5
Mass Flux, G ($\frac{\text{kg}}{\text{m}^2 \cdot \text{s}}$)	150-25000
Inner Wall Heat Flux, q_w ($\frac{\text{kW}}{\text{m}^2}$)	0-540

The results from the test section measurements for the two-phase fluid pressure oscillation case are shown in Figure 10.

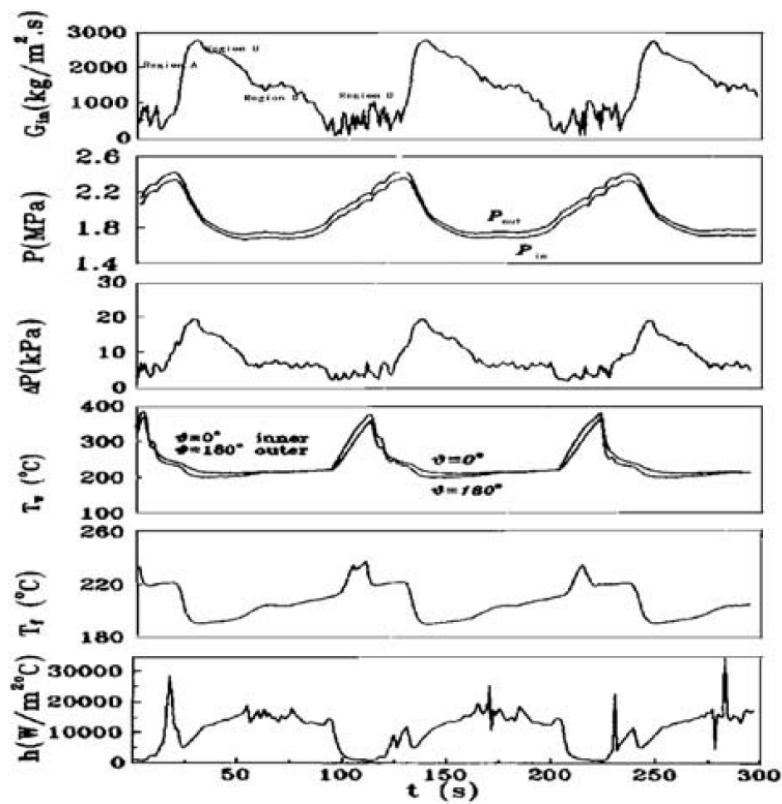


Figure 10. Experimental two-phase flow pressure oscillation results. Source: [14].

Guo et al. [14] concluded that the pressure drops across the test section and heat transfer coefficients agreed favorably with previous experimental and empirical correlations. Additionally, the location of the surge tank compressible volume had a considerable influence on the characteristics of the pressure oscillations and other flow parameters. The oscillatory effects of heat transfer from the two-phase flow to the wall were most affected by the Reynolds number, Prandtl number, and the frequency and magnitude of the pressure oscillations. Additionally, the heat transfer coefficient and mass flux were found to oscillate at an opposite phase relative to temperature and pressure. Relevant experimental data from the two-phase flow portion was compared with TRACE simulation data modeled in this study.

THIS PAGE INTENTIONALLY LEFT BLANK

III. TRACE IMPLEMENTATION

Many three-dimensional CFD codes, such as Ansys CFX and Fluent, are well suited to model the flow characteristics in this effort. However, they can be prohibitively expensive regarding computational power and time. In contrast, TRACE is a one-dimensional code, allowing for greater modeling simplicity and significantly less computational effort. TRACE is a thermal hydraulics code, meaning that it was engineered to analyze large-scale systems, such as pressurized water reactors. As a result, TRACE may not simulate the flow physics in individual components, such as a single pipe, to a high degree of fidelity. Regarding usage limitations, the TRACE theory manual explicitly states, “The TRACE code is not appropriate for modeling situations in which transfer of momentum plays an important role at a localized level. For example, TRACE makes no attempt to capture, in detail, the fluid dynamics in a pipe” [15]. TRACE was expected to have limitations in modeling the flows in this study. In order to properly construct the simulations in this effort, the ability to model two-field vapor and liquid droplet flow and the associated mass and heat transfer were paramount. Additional feature development, including pipe bend losses, changes in elevation, pipe surface roughness, latent pipe wall heat, and transient fluid properties, was also necessary.

A. FLUID FIELD MODELS

Wet steam flow was modeled in TRACE using a two-field equation set for the vapor and liquid phases. The liquid phase was modeled as a droplet field dispersed homogeneously within the vapor field. Unlike Ansys CFX, which allows multiple options for two-phase material selection, such as a homogeneous binary mixture model, TRACE utilizes one-dimensional Navier-Stokes equations for continuity, energy and momentum of each fluid field [15]. Additionally, relations for the transfer of mass, energy, and momentum between these two sets of equations are necessary for closure and will be discussed in the following section of this chapter. TRACE’s formulation of the time averaged continuity equation for the liquid is defined in (3) as

$$\frac{\partial[(1-\alpha)\bar{\rho}_l]}{\partial t} + \nabla \cdot \left[(1-\alpha)\bar{\rho}_l \bar{\mathbf{u}}_l \right] = -\bar{\Gamma}, \quad (3)$$

while the corresponding continuity equation for the vapor field is listed in (4) as

$$\frac{\partial[(\alpha)\bar{\rho}_v]}{\partial t} + \nabla \cdot \left[\alpha \bar{\rho}_v \bar{\mathbf{u}}_v \right] = \bar{\Gamma}. \quad (4)$$

Equation (3) and (4) illustrate the relationship between liquid and vapor field continuity, where the rates of mass transfer to and from each are always equal and opposite. Overbars and arrows denote time-averaged and vector quantities respectively. Γ represents the rate of mass transfer between the two phases. This jump relation will be discussed in greater detail later in the chapter. Equation (5) defines TRACE's time averaged energy equation for the liquid field as

$$\begin{aligned} & \frac{\partial \left[(1-\alpha)\bar{\rho}_l \left(\bar{\mathbf{e}}_l + \frac{\bar{\mathbf{u}}_l^2}{2} \right) \right]}{\partial t} + \nabla \cdot \left[(1-\alpha)\bar{\rho}_l \left(\left(\bar{\mathbf{e}}_l + \frac{\bar{\mathbf{p}}}{\bar{\rho}_l} + \frac{\bar{\mathbf{u}}_l^2}{2} \right) \bar{\mathbf{u}}_l \right) \right] \\ & = -\nabla \cdot \left[(1-\alpha)\bar{\mathbf{q}}_l \right] + (1-\alpha)\bar{\rho}_l \bar{\mathbf{g}} \cdot \bar{\mathbf{u}}_l - \Gamma \bar{h}_l' + \bar{W}_l + \bar{q}_{dl}. \end{aligned} \quad (5)$$

Equation (6) defines the time-averaged vapor field energy equation as

$$\begin{aligned} & \frac{\partial \left[\alpha \bar{\rho}_v \left(\bar{\mathbf{e}}_v + \frac{\bar{\mathbf{u}}_v^2}{2} \right) \right]}{\partial t} + \nabla \cdot \left[\alpha \bar{\rho}_v \left(\left(\bar{\mathbf{e}}_v + \frac{\bar{\mathbf{p}}}{\bar{\rho}_v} + \frac{\bar{\mathbf{u}}_v^2}{2} \right) \bar{\mathbf{u}}_v \right) \right] \\ & = -\nabla \cdot \left[\alpha \bar{\mathbf{q}}_v \right] + \alpha \bar{\rho}_v \bar{\mathbf{g}} \cdot \bar{\mathbf{u}}_v + \Gamma \bar{h}_v' + \bar{W}_v + \bar{q}_{dv}. \end{aligned} \quad (6)$$

Given the focus of this effort on pressure transients, it is important to note that the energy equation ties changes in pressure to the overall energy in the flow. A pressure drop was expected to result in a lower fluid enthalpy. Equation (7) shows TRACE's time averaged momentum equations for the liquid field as

$$\frac{\partial \left[(1-\alpha)\bar{\rho}_l \bar{\mathbf{u}}_l \right]}{\partial t} + \nabla \cdot \left[(1-\alpha)\bar{\rho}_l \bar{\mathbf{u}}_l \bar{\mathbf{u}}_l \right] = \nabla \cdot \left[(1-\alpha)\bar{\mathbf{R}}_l \right] + (1-\alpha)\bar{\rho}_l \bar{\mathbf{g}} - \bar{\mathbf{M}}_l, \quad (7)$$

while (8) shows the time averaged liquid momentum equation,

$$\frac{\partial [\alpha \overline{\rho_v} \overline{\mathbf{u}_v}]}{\partial t} + \nabla \cdot \alpha \overline{\rho_v} \overline{\mathbf{u}_v} \overline{\mathbf{u}_v} = \nabla \cdot [\alpha \overline{\mathbf{R}_v}] + \alpha \overline{\rho_v} \overline{\mathbf{g}} + \overline{\mathbf{M}_i}. \quad (8)$$

$\mathbf{M}_i$ represents the rate of interfacial momentum transfer between the phases and $\mathbf{R}$ is the combined Reynolds stress and viscous shear stress tensor. The velocity in both momentum equations is split for a time averaged value and a fluctuating value due to the Reynolds stress. TRACE balances the terms on the right side of the equation using empirical correlations [15].

B. HEAT TRANSFER MODELS

Interfacial heat transfer between the two fluid fields forms a thermal energy jump relation shown in (9), also known as a heat conduction limited model, defined by

$$\Gamma = \frac{-(q_{iv} + q_{il})}{(h'_v - h'_l)}. \quad (9)$$

The vapor field interface heat flux is given in (10) as

$$q_{iv} = \frac{p_v}{p} h_{iv} a_i (T_{sat} - T_l), \quad (10)$$

while the corresponding liquid field interface heat flux is defined in (11) as

$$q_{il} = h_{il} a_i (T_{sat} - T_l). \quad (11)$$

Equation (9) through (11) dictate the rate of phase change between liquid and vapor. In these equations, Γ denotes the phase change rate of mass per unit volume, q denotes heat flux, h' denotes enthalpy, h denotes the heat transfer coefficient, a is the area at the fluid interface per unit volume, P denotes pressure, and T denotes temperature. The subscripts i , l , sat , and v denote the interface between the two fluids, liquid, saturated vapor, and vapor. TRACE nomenclature defines the word “gas,” denoted by the subscript g , as a mixture of condensable and non-condensable gases [15]. Because non-condensable gases were not modeled in this study, there was no distinction between the two in (10) and (11) for the purpose of this research. A positive rate of phase change indicates evaporation, while a

negative rate indicates condensation. TRACE determines the energy and momentum transfer rates between the vapor and liquid fields in a similar fashion using separate jump relation conditions.

Transient scenarios in this effort included models of heat flow between the pipe wall and the flow and at the interface of the liquid and vapor fluid fields. TRACE uses five wall heat transfer models based on the temperature differences between the wall and the fluid fields and the flow regime type. These models are summarized individually in Table 2.

Table 2. TRACE boiling models. Adapted from [15].

Heat transfer model	Applicable boiling scenarios
1. Pre-CHF	wall-liquid convection, nucleate boiling, and subcooled boiling
2. CHF	peak heat flux in the nucleate boiling heat transfer regime and the wall temperature at which it occurs
3. Post-CHF	models for transition and film boiling heat transfer
4. Minimum Film Boiling Temperature	the temperature above which wall-liquid contact does not occur
5. Condensation Heat Transfer	models for film condensation and the non-condensable gas effect

TRACE includes blending functions to combine the many heat transfer correlations from these five models to produce a boiling curve based on the wall-to-fluid temperature gradient for input of constant flow parameters and pipe characteristics. An example of this function is shown in Figure 11.

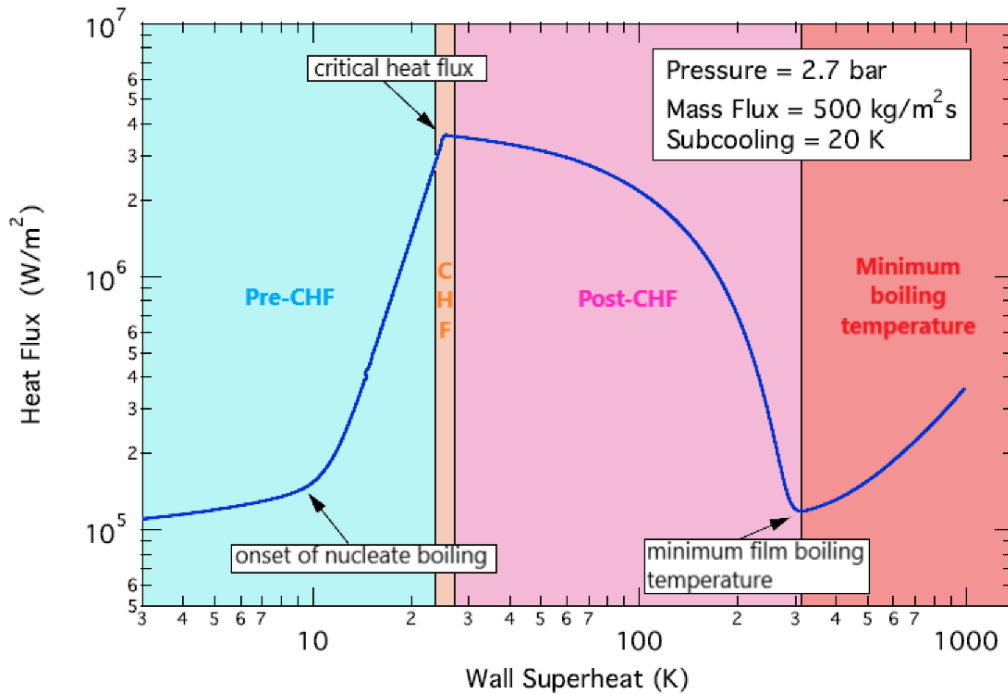


Figure 11. TRACE boiling curve example. Adapted from [15].

The first TRACE heat transfer model is initially characterized by pure wall-to-liquid convection. As the vapor mass fraction increases, heat transfer is increasingly driven by bubble formation, departure, and surface quenching at the wall due to nucleate boiling [15]. The critical heat flux (CHF) point is the wall's peak effective boiling heat. After CHF, contact between the liquid and the wall is increasingly intermittent, leading to a completely dry wall. Within the post-CHF model, TRACE assumes dispersed film boiling for vapor volume fractions greater than 99% [15]. This regime encompassed all of the flows in this effort. Primary heat transfer mechanisms for this model include forced convection and thermal radiation to the superheated vapor and dispersed water droplets, respectively. The wall heat transfer models used by TRACE for this boiling region are tailored for tube and rod bundles found in reactor vessels rather than pipe walls. [15]. TRACE was engineered to model loss of coolant reactor accidents. In such cases, the flow regimes and associated heat transfer mechanisms up to CHF are most important. After CHF, the reactor accident has already occurred, and the heat transfer does not need to be modeled in great detail. As

a result, TRACE may not model the high void fraction post-CHF heat transfer in this effort with a high degree of fidelity.

IV. STEADY-STATE TRACE MODELING

The model for steady-state two-phase steam pipe flow developed by Axelsson [13] was generated in TRACE, refined, and tested to properly model the transient flow conditions of interest in this effort. The Axelsson model did not account for inner wall surface roughness losses. The pipe had a constant hydraulic diameter and bend radius of curvature of 0.25 m. The geometry of the Axelsson pipe model in Figure 8 included five round bends, all with a curvature radius of 0.25 m. From inlet to outlet, these bends had turn angles of 90°, 84°, 84°, 84°, and 96°, respectively. Axelsson simplified this geometry in TRACE and modeled all pipe bends with 90° pipe bend loss coefficients. This study improved Axelsson's approximation by calculating K factors specific to the angle of each bend in the model. All model inputs and new geometric features were implemented and simulated individually so that the expected effects on the flow model could be isolated and examined. The completed steady-state model was interrogated for a range of inlet qualities, and the resulting data was analyzed to gain insight into TRACE performance and anticipated transient modeling.

A. MODEL DEVELOPMENT

Axelsson's steady-state model pipe geometry specified in Figure 8 was modeled using three components included in the TRACE model editor; a pipe, an inlet, and an outlet. TRACE nomenclature recognizes the latter two components as a "fill" and a "break," respectively. The TRACE manuals for theory and model inputs were consulted for proper implementations of the desired features [15]. The pipe was modeled using one node for every quarter meter, with 92 nodes for the pipe length of 23 m. The angle of each pipe node with respect to the horizontal axis was also modeled so that the geometric model reflected the elevation changes in Figure 8. This portion of the model, displayed in Figure 12, only represented gravitational effects and did not account for lateral pipe bends or head loss associated with pipe bends.

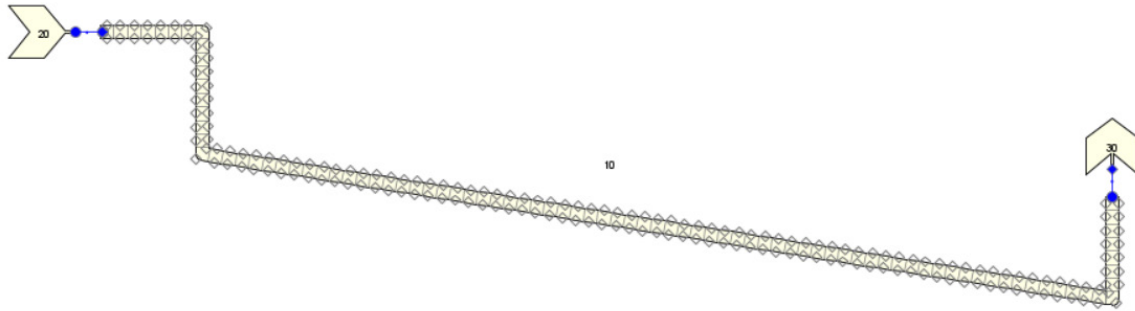


Figure 12. Steady-state TRACE pipe model

In this model, component 10 is the pipe, component 20 is the fill, and component 30 is the break. Axelsson's input fluid parameters are shown in Table 3. The fluid properties were taken from the NIST database for saturated steam properties at 2.639 MPa and superheated to 505.89 K from a saturation temperature of 500 K.

Table 3. TRACE fluid parameters for steady-state pipe model. Adapted from [13].

Input Property	Component		
	Fill (inlet)	Pipe	Break (outlet)
Initial Flow Quality, X (%)	85-99	100	100
Initial Pressure, p (MPa)	2.639	2.639	2.639
Liquid Velocity, u_l (m/s)	40	-	-
Vapor Velocity, u_v (m/s)	40	-	-
Initial Liquid Temperature, T_l (K)	505.89	505.89	505.89
Initial Vapor Temperature, T_v (K)	505.89	505.89	505.89

The thermophysical properties for superheated steam at the temperature and pressure specified in Table 3 were derived from the NIST database and listed in Table 4.

Table 4. Superheated water vapor properties

Property	Value
Temperature, T_v (K)	505.89
Pressure, p_v (MPa)	2.639
Density, ρ_v (kg/m ³)	12.918
Specific Heat, c_v (J/g·K)	2.1439
Specific Heat, c_p (J/g·K)	3.2296
Dynamic Viscosity, μ_v (Pa·s)	1.6884E-5
Thermal Conductivity, k_v (W/m·K)	0.044378

The ratio of liquid to vapor cannot be specified in terms of mass in the TRACE input interface. TRACE defines this inlet parameter volumetrically. In order to convert between gravimetric and volumetric vapor fractions, a MATLAB script was developed and attached in Appendix A. This script calculated vapor volume fractions for a range of vapor mass fractions, or qualities, given a fixed thermodynamic state and inlet cell volume. The calculated qualities from this script are shown in Table 5.

Table 5. Vapor volume fraction to vapor quality conversion

Inlet Quality, X (%)	Inlet Vapor Volume Fraction, α
85	0.997223
90	0.998256
95	0.999181
96	0.999355
97	0.999525
98	0.999691
99	0.999855
99.96	0.999994

Pipe flow transitions from laminar to turbulent for Reynolds numbers greater than 4000 [16]. Correlations for absolute roughness and bend head loss factors are affected mainly by turbulence. In order to calculate the Reynolds number and degree of turbulence, the two-phase flow in the complex pipe geometry was simplified as a developed flow in a

straight pipe with a flow quality of 100%. Equation (12) was used to calculate this approximated Reynolds number as

$$\text{Re}_d = \frac{\rho_v u_v d_h}{\mu_v}. \quad (12)$$

Data from Table 3 and Table 4 was used to calculate a simplified Reynolds number of 7.651E6 [2]. Following this strong indication of turbulent flow, this study selected roughness correlations experimentally derived by Moody [17] for completely turbulent flow through unfouled commercial steel piping. From the diagram attached in Appendix C, the relative roughness was 4E-4, and the absolute roughness used in the TRACE input model was 1E-4 m.

This study used Idel'chik's [18] experimentally derived correlations for pressure losses in curved pipe fittings for all modeled pipe bends. Idel'chik developed bend loss correlations regarding most parameters relevant to this simulation, including Reynolds numbers, wall surface roughness, bend angle, pipe shape, and radius of curvature. Idel'chik's correlations were only applicable for single-phase flows. The following calculations used the Reynolds number approximation in (9) to simplify the flow as dry steam. Idel'chik's pipe bend loss coefficients, K , were calculated in (13) as

$$K = K_{\text{bend}} + K_{\text{fric}}. \quad (13)$$

The bend resistance coefficient, K_{bend} , was calculated as

$$K_{\text{bend}} = \pi \frac{\delta^\circ}{180^\circ} \left(\lambda \frac{r^\circ}{d_h} \right). \quad (14)$$

The bend friction coefficient, K_{fric} , was given by

$$K_{\text{fric}} = ABC. \quad (15)$$

Equations (13) and (14) were derived from Idel'chik diagrams 2-3 and 6-1, and (15) was adapted from the Abramovich fluid resistance correlation [18]. The absolute roughness of 4E-4, taken from Moody's experimental data, and the Reynold number were used to select a value of 0.016 for the referred length friction coefficient, λ , extrapolated from data

taken by Nikuradze [19]. This value, in addition to the bend radius of curvature, r° , bend angle, δ° , and hydraulic diameter, d_h , were used to calculate the bend resistance coefficients in (14). The coefficients, A, B, and C, were dimensionless parameters calculated in terms of the bend angle, the ratio of radius of curvature to hydraulic diameter, and the cross-sectional area ratio of bend inlet to outlet, respectively [18]. More detailed documentation relating to these calculations is available in Appendix D. Using these coefficients, the pipe bend loss coefficients were calculated for all bend angles. The values relevant to (13) through (15) are displayed in Table 6.

Table 6. Idel'chik pipe bend loss coefficient values

Parameter	Pipe Bend $\delta^\circ = 84^\circ$	Pipe Bend $\delta^\circ = 90^\circ$	Pipe Bend $\delta^\circ = 96^\circ$
λ	0.016	0.016	0.016
A	0.95	1	1.04
B	0.23	0.23	0.23
C	1	1	1
K_{bend}	0.0235	0.0251	0.0268
K_{fric}	0.2185	0.23	0.2392
K	0.242	0.2551	0.2660

Pressure losses in pipe bends are largely due to vortices that form at the outer edge of the bend and at the inner surface just after the bend, shown in Figure 13 [18]. This effort implemented the calculated pipe bend loss coefficients at the first nodal interface after each bend in order to emulate this behavior.

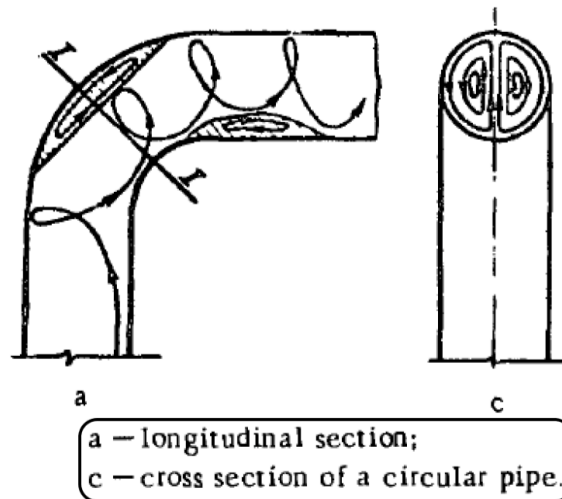


Figure 13. Vortices induced by circular pipe bends. Adapted from [18].

Because there is no heat transfer between the pipe wall and the fluid flow at steady-state, modeling the thickness and thermal properties associated with the pipe wall material was unnecessary for this portion of the effort. Appendix J includes a detailed report of all steady-state model inputs. This model was run in the TRACE solver for the range of inlet vapor volume fractions in Table 4.

B. SIMULATION RESULTS

The model converged to steady-state after 201 iterations and approximately 10 seconds for all inputs of inlet flow quality with respective minimum and maximum timestep sizes of 1E-6 and 1E-2 seconds. Fluid properties were examined for all cases to evaluate TRACE's ability to model pipe fluid dynamics for medium to high-quality flows. Fluid pressures along the length of the pipe for all inlet flow qualities are shown in Figure 14.

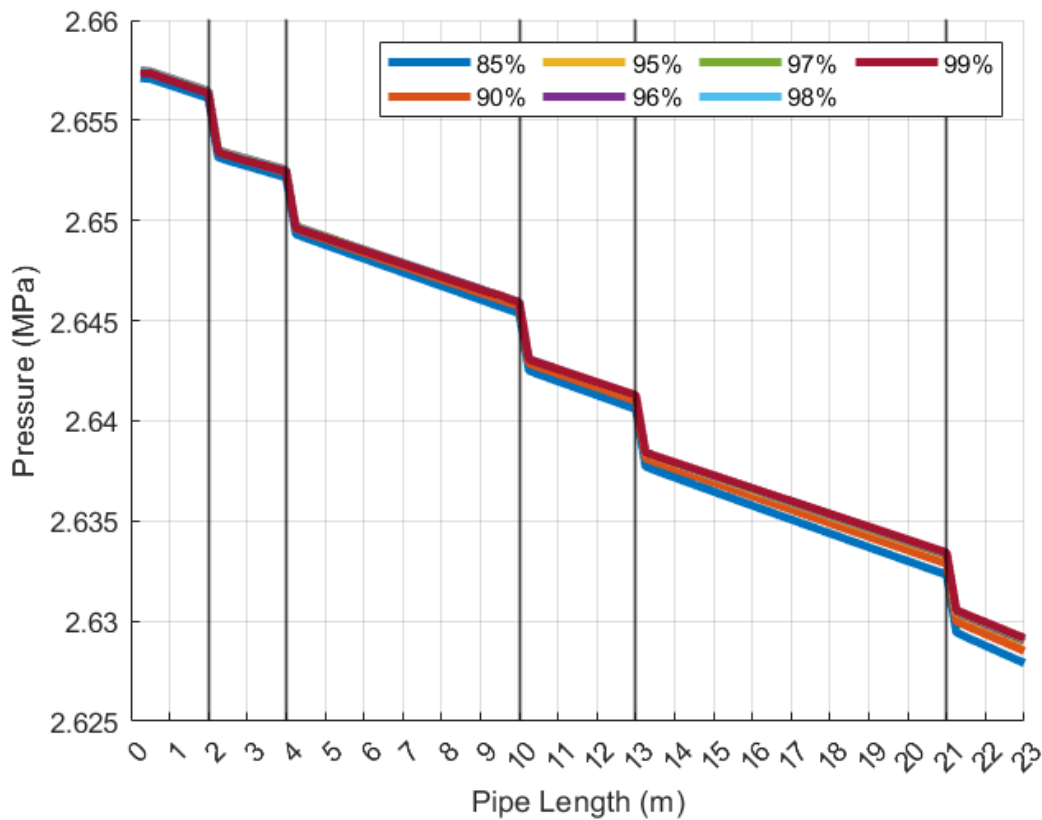


Figure 14. Pressure drop across pipe length for inlet flow quality range

Cases with lower inlet flow qualities experienced a greater pressure drop, likely due to increased interfacial drag between the liquid and vapor phases. Figure 15 illustrates the pressure drop rate as a function of pipe length. The areas of the pipe with a positive incline had a higher pressure drop rate because of the additional loss of hydrostatic head, and vice versa. Pipe bend locations, indicated by vertical black lines, were placed at pipe lengths corresponding to half the total angle for each bend. The pipe bends with an 84° bend angle induced relatively small pressure losses. However, the pressure drop in the 96° bend was slightly lower than in the 90° bend. The pressure differential between fill and break was approximately double the values predicted in Axelsson's simulation results for TRACE and Ansys CFX, mainly due to the implementation of wall roughness and refined bend loss coefficients. [13]. The placement of pipe bend loss coefficients reflected the pressure drop distribution due to the formation of bend vortices discussed in the model development.

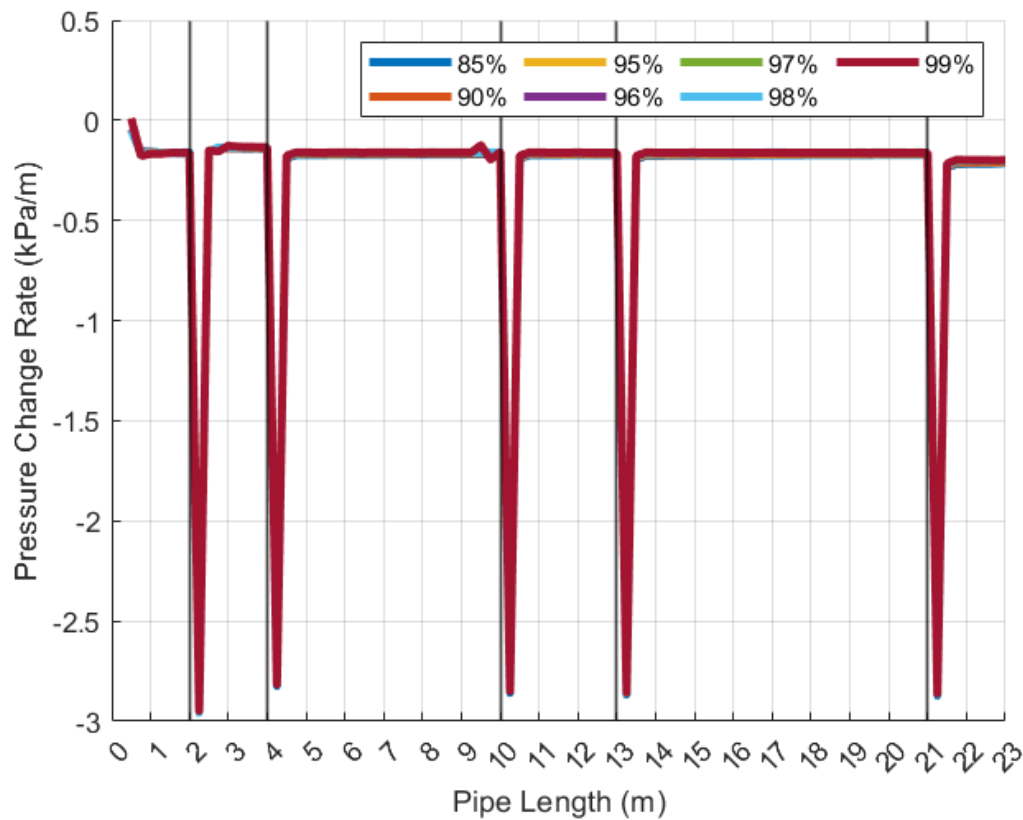


Figure 15. Pressure change rate along pipe length for inlet flow quality range

The TRACE solver calculates pressure drop using initial pressure inputs at the fill and break as well as pipe characteristics, then iterates to determine final inlet and outlet pressures. TRACE tended to maintain a break pressure nearer to the input value and to allow the calculated pressure differential to manifest in the steady-state fill pressure. The resulting inlet pressure was 16.5 kPa higher than the specified value of 2.639 MPa. In this case, setting static boundary conditions for pressure at the fill or break over-constrained the problem and led to failure. The pressure drop across the pipe was approximately 29 kPa for all input flow qualities. The model was updated and iterated with respective fill and break initial pressures of 2.639 and 2.6105 MPa to emulate the previously simulated head loss for the desired fill and break pressures. This change did not affect the simulation results. Figure 16 illustrates the change in cell interface flow quality with respect to pipe length for each flow quality input.

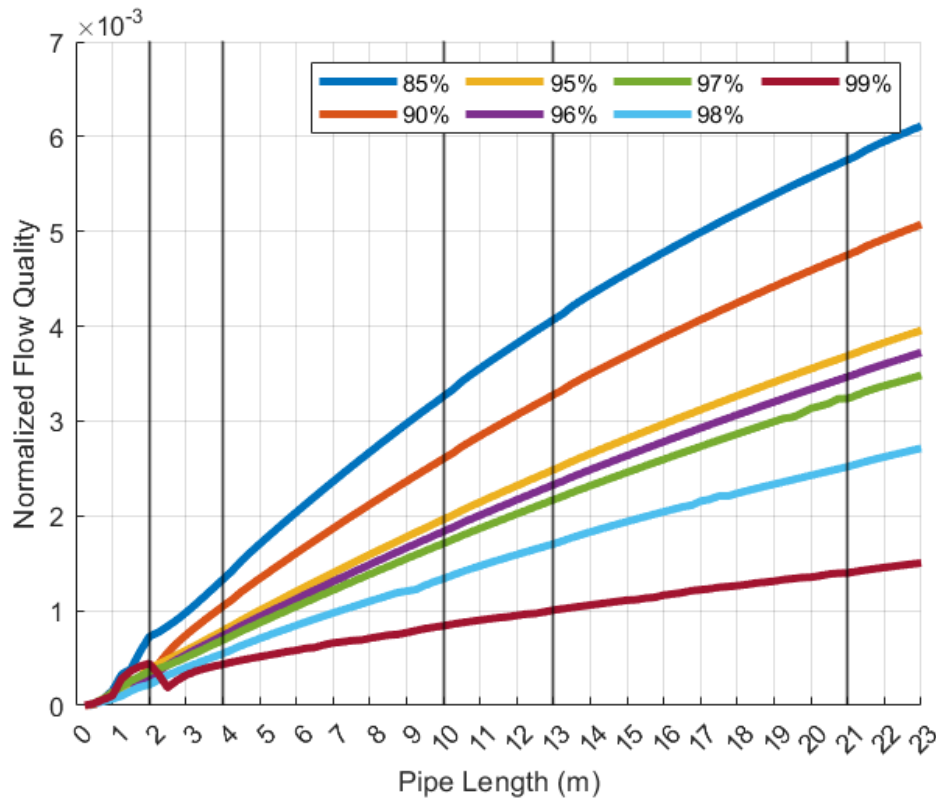


Figure 16. Normalized cell interface flow quality along pipe length for inlet flow quality range

Flow quality within the pipe was calculated by dividing the vapor mass flow rate by the respective total mass flow rate at the interfaces of the 92 cells in the model. The data was normalized by dividing the flow quality along the pipe length by the set inlet flow quality for each case and subtracting one from the result. This normalization shows the change rate in flow quality from the inlet of the pipe. The flow quality increased at a near-constant rate along the pipe length for all cases and at a greater rate for lower inlet flow qualities. This trend indicated a positive mass transfer rate from the droplet field to the vapor field across the pipe which was more pronounced at lower initial flow qualities. The change in flow quality near the inlet for the 99% inlet flow quality case was irregular, indicating limitations in the TRACE solver for modeling high quality flows.

As the water droplets evaporated, the temperature of the vapor field decreased similarly, with sharper rates of decline for lower inlet qualities. Figure 17 indicates this was likely due to the droplets pulling heat from the vapor field during phase change.

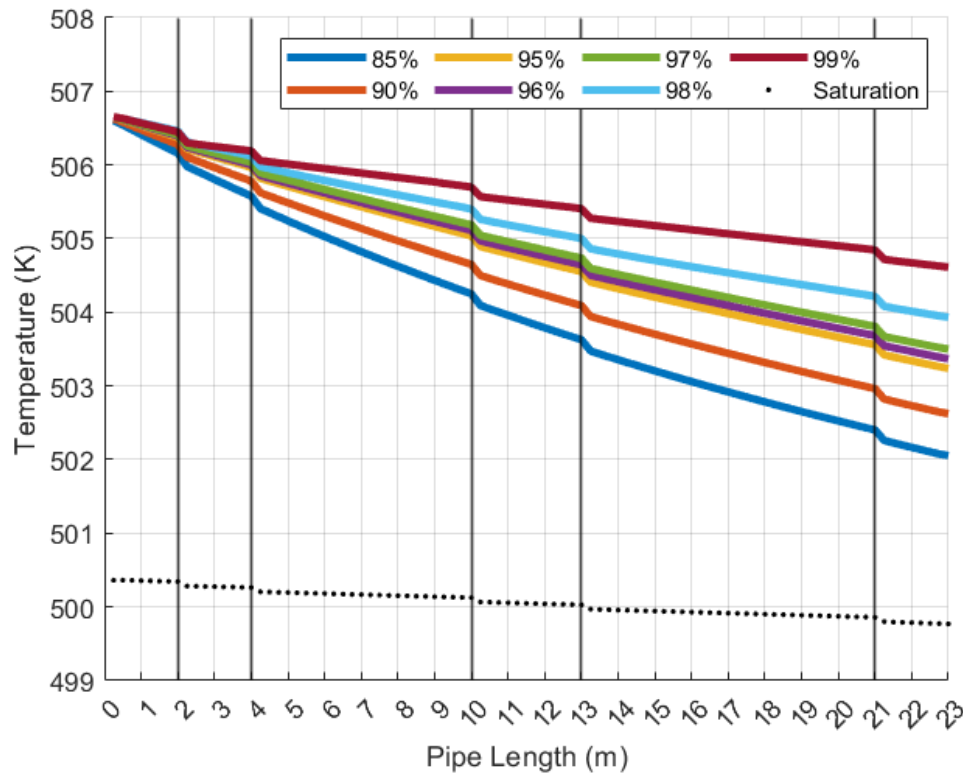


Figure 17. Vapor temperature along pipe length for inlet flow quality range

Because more energy was required for the greater mass phase change for lower inlet flow qualities, the rate of vapor temperature drop was higher for these cases. Despite the temperature drop, the vapor temperature remained superheated above the saturation temperature. The inlet temperature increased by approximately one degree from initialization, due to the increased inlet pressure.

Figure 18 illustrates a decline in liquid temperature over the pipe length. Similar to the decrease in vapor temperature, energy from the liquid was absorbed in the phase change, resulting in a temperature drop.

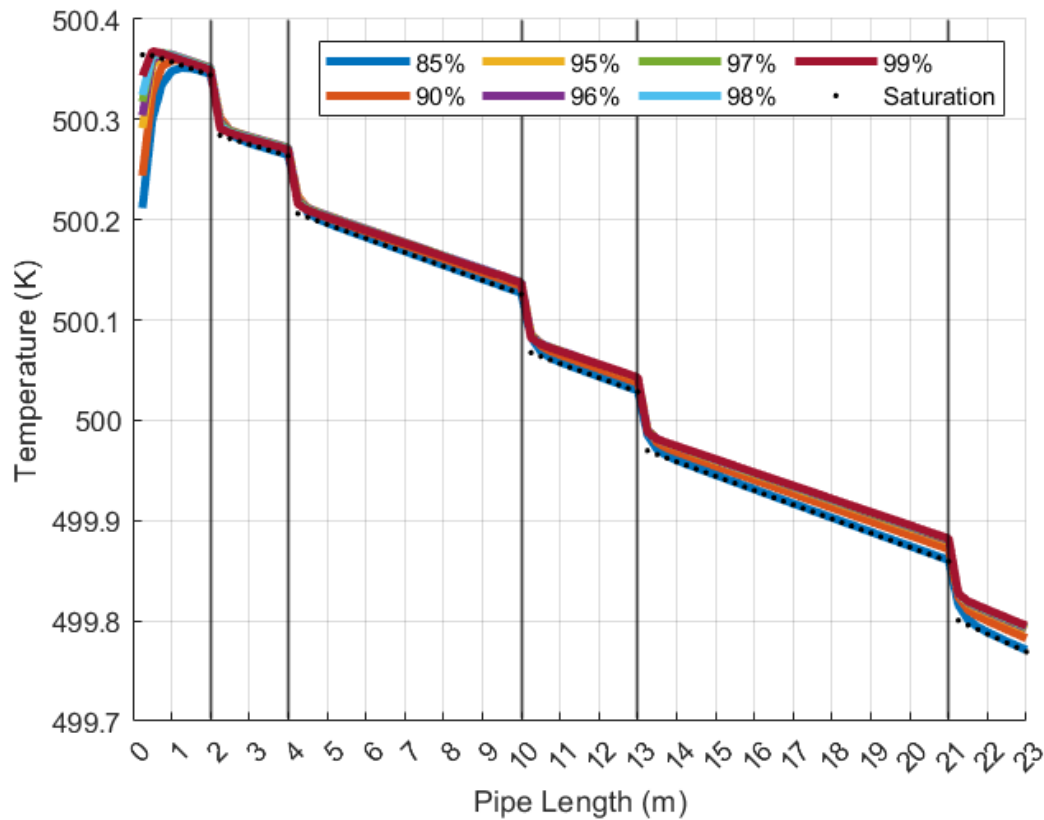


Figure 18. Liquid temperature along pipe length for inlet flow quality range

For all cases, the TRACE solver forced the liquid to the saturation temperature, approximately five degrees lower than the initial specified temperature. For higher inlet flow qualities, the liquid was marginally superheated. The increased pressure loss at pipe bends resulted in a coupled decrease in temperature.

The steady-state effort was especially concerned with the distribution of cell liquid mass in relation to the pipe geometry. The cell vapor mass fraction along the pipe length, shown in Figure 19, was calculated as the ratio of vapor mass to total fluid mass for each of the 92 pipe cells.

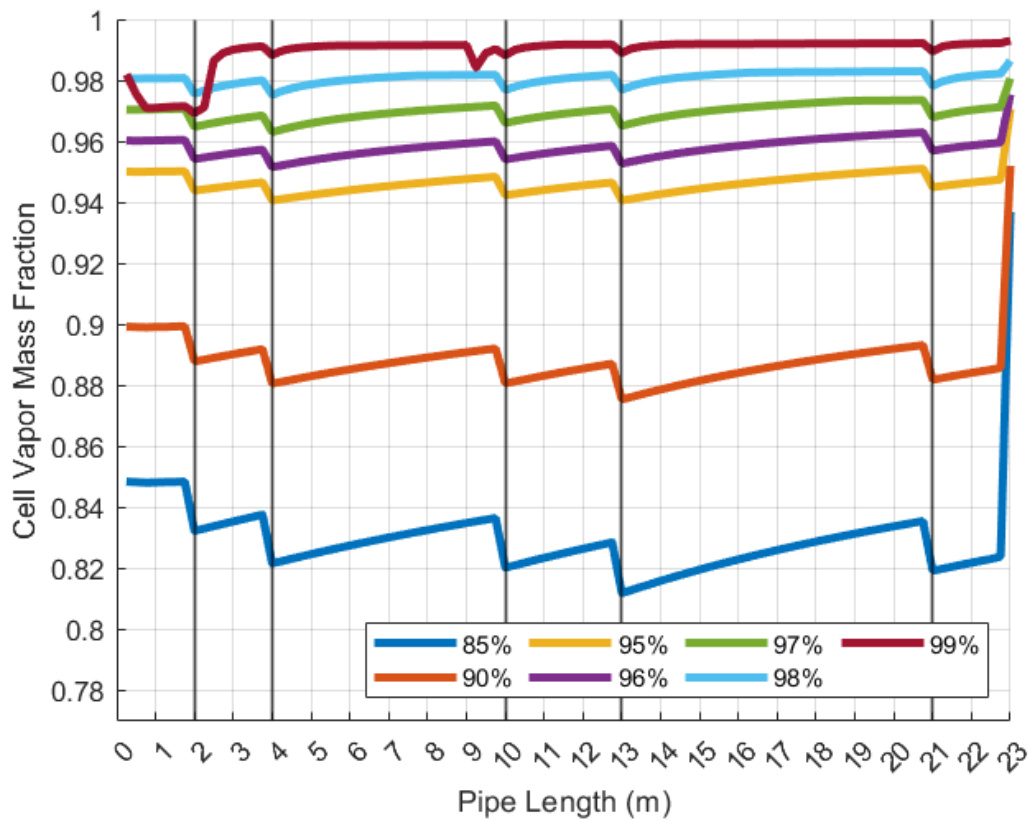


Figure 19. Cell vapor mass fraction along pipe length for inlet flow quality range

TRACE calculates a homogenous velocity for each of the cells in the fluid fields and does not model recirculation regions [15]. The distribution of fluid phases in terms of mass flow rate and cell mass were very different, as seen through a comparison of Figure 18 and Figure 19. This distribution indicated that there was stagnant fluid around the pipe bends. It was noted that the trends for phase distribution across the pipe were shared for inlet flow qualities of 85 to 98%. However, the 99% inlet flow quality simulation did not yield reasonable results. The vapor mass fraction dropped to 97% before the first bend and dipped and recovered just before the third bend. These non-physical results indicated that the TRACE solver is not well suited for high-quality flows, beginning at some threshold between 98 and 99%. Because TRACE is a one-dimensional code, the effects of bends on the flow were characterized through elevation changes and bend K factors. Three-dimensional effects such as vortices and recirculation regions were not modeled. To further

investigate the decrease in vapor mass fraction at the bends, the K factors were removed from the model for the 98% inlet flow quality case and the results were compared with the original model in Figure 20.

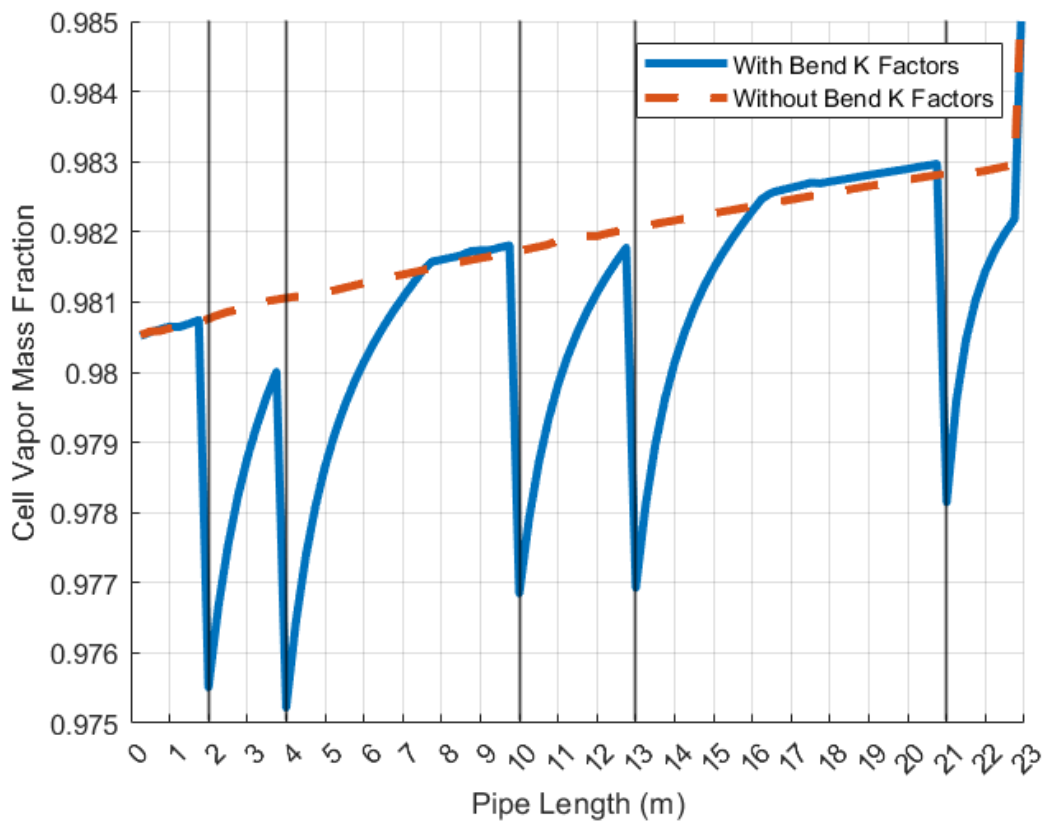


Figure 20. Cell vapor mass fraction along pipe length for 98% inlet flow quality with and without bend K factors

The bend K factors caused the cell vapor mass fraction to drop just before the middle of each bend. This was unusual because the rate of pressure loss increased after the middle of each bend, indicating that the lower vapor mass fraction was not due to the K factor pressure drop. The vapor mass fraction increased after each bend, and in the cases where it fully recovered before the subsequent bend, it was higher than in the case where there were no bends.

The masses of fluid phases in each pipe were added to calculate the total mass of each phase for the range of inlet flow qualities. The phase mass distribution for each case is displayed in Figure 21 and tabulated in Appendix B

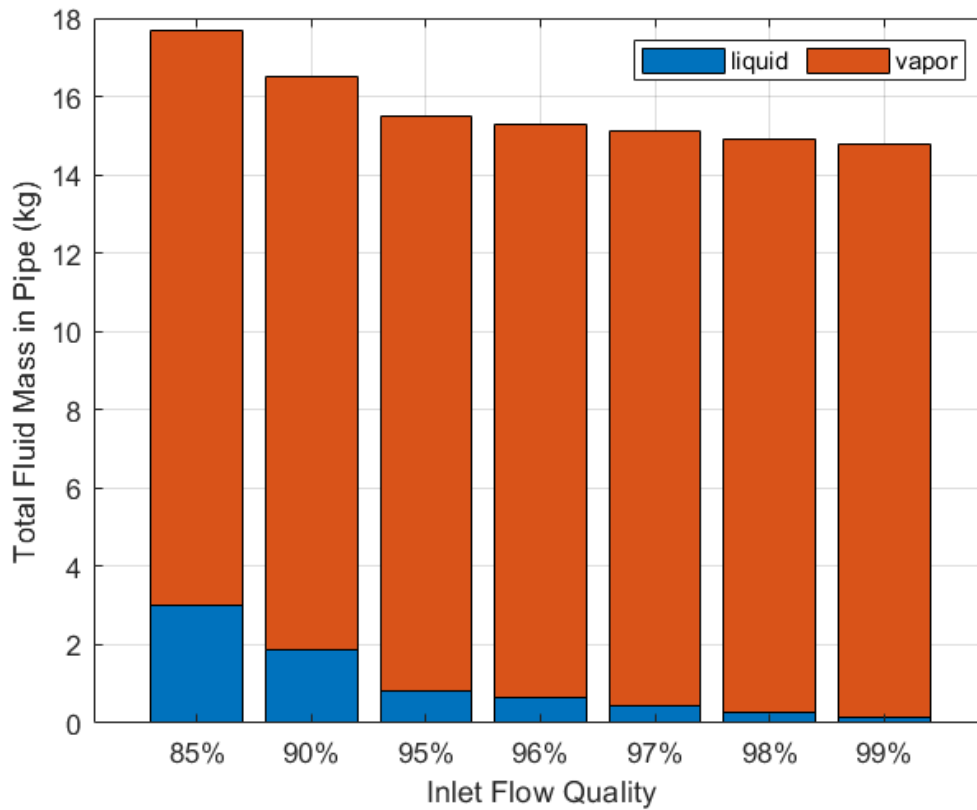


Figure 21. Total fluid mass in pipe for inlet flow quality range

The mass of liquid in the pipe was greater for higher inlet flow quality cases, resulting in a greater mixture density and mass.

V. PRESSURE TRANSIENT TRACE MODELING

The pressure transient model emulated changing flow conditions in a secondary steam loop resulting from variations in reactor power output. This transient was characterized by a 15% drop in pressure distributed over 30 seconds, and a subsequent recovery to the steady-state pressure input after an additional 30 seconds. Figure 22 illustrates the pressure transient over time.

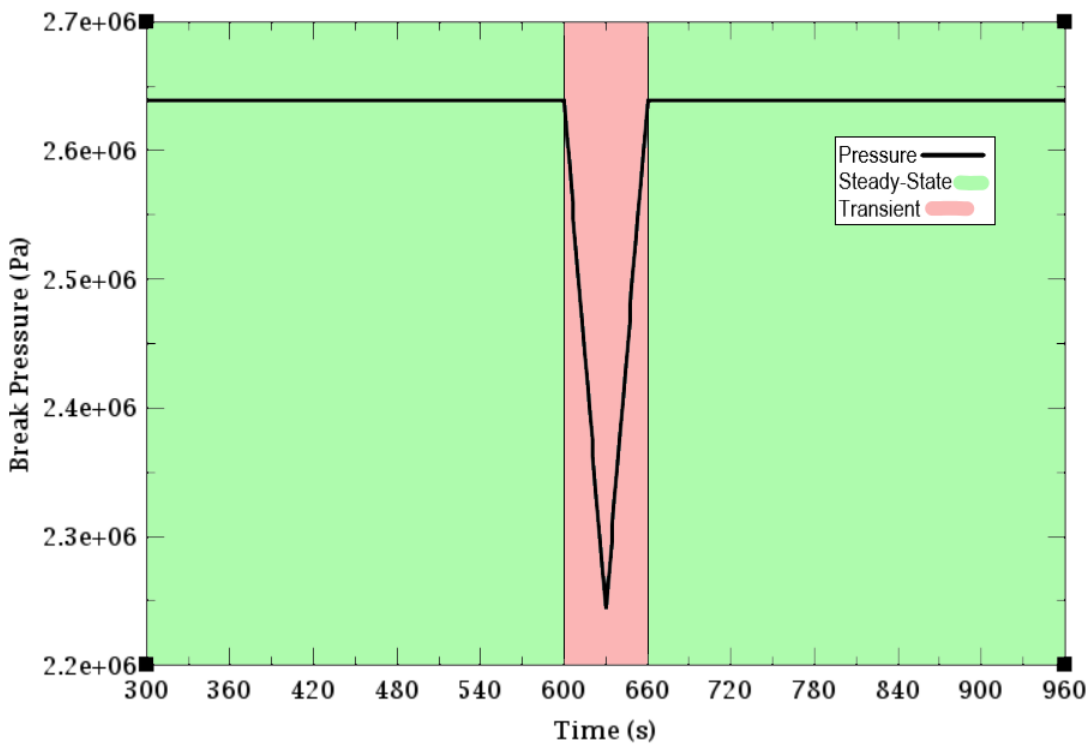


Figure 22. Transient pressure input

The steady-state flow conditions modeled before and after the pressure fluctuation were identical to those implemented in Chapter IV. Because the steady-state results indicated that TRACE is better suited for predicting pipe moisture distribution at inlet flow qualities under 99%, a 98% inlet flow quality was modeled in this transient study.

Because heat transfer between the pipe wall and the fluid flow does not occur at steady-state, the pipe wall's thermal properties were not modeled in Chapter IV. The pressure drop in the transient effort was expected to result in a lower fluid temperature and heat flux from the pipe wall to the fluid. To further emulate steam turbine systems, this effort selected American Society for Testing and Materials (ASTM) A508 carbon steel with a thermal conductivity of 57 W/m·K for pipe wall modeling [15]. The wall thickness was 10 mm to approximate schedule 80 pipe. The latent heat flux in the pipe wall to the fluid flow was an important feature in this transient study. The wall thermal mass was built using concentric radial nodes around each pipe cell, leading to a lumped capacitance approach in model development. An analysis for the chosen wall material and thickness was conducted to ensure a conservative Biot number of 0.05.

A. MODEL DEVELOPMENT

The model developed in Chapter IV. was used as a baseline for transient simulations. The residuals threshold that TRACE uses to determine whether the solution iterated to steady-state were not used in transient cases. Instead, the simulation run time was set manually. The steady-state cases in the previous chapter terminated at a problem timestep of 201 seconds. Accounting for the steady-state lag due to the added thermal mass of the pipe wall, the transient case was iterated for 600 seconds before the pressure transient. The pressure transient was followed by an additional 600 seconds for steady-state iterations, reflected in Figure 22. Following a timestep sensitivity study, the timestep size range from the steady-state model development was retained for the transient pressure study.

For this portion of the effort, the constant pressure defined at the break for steady-state simulations was replaced by a pressure control signal which varied based on the accumulated timestep. A diagram of the control signal components is shown in Figure 23.

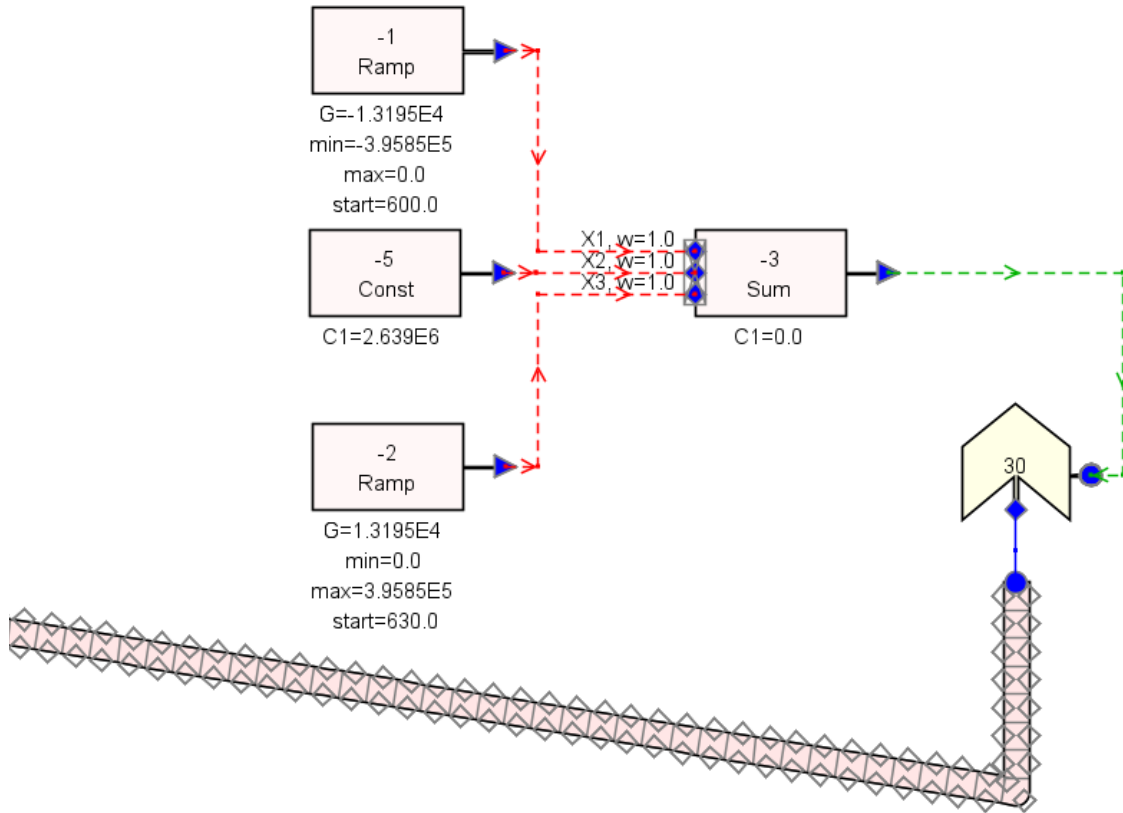


Figure 23. Pressure transient break control signal

This control signal was equal to the sum of one constant signal and two ramp function signals. The constant signal was set to $2.639\text{E}6$ for all timestep values, representing the steady pressure input in Pascals. The upper ramp function signal was on for time step values above 600 seconds with a gain equal to the pressure drop rate. The lower ramp function signal had an equal and opposite gain and was on for timestep values greater than 630 seconds. The sum of these three control blocks produced a signal for break pressure identical to Figure 22.

Pipe wall heat transfer was modeled one-dimensionally in the radial direction, with adiabatic boundaries at cell interfaces and outer wall surfaces in accordance with Ito and Kuramae's findings [11]. ASTM A508 carbon steel model was selected from available TRACE material models. The minimum thickness of each radial node was limited by the

lumped capacitance approximation in the TRACE solver. The Prandtl number, defined in (16) as

$$\text{Pr} = \frac{\mu_v c_p}{k_v}, \quad (16)$$

was calculated in order to determine the Nusselt number. The Dittus-Boelter correlation [10] Nusselt number and its range of application is given in (17) as

$$\text{Nu}_d = 0.023 \text{Re}_d^{4/5} \text{Pr}^{2/5} \quad (17)$$

$$\left[\begin{array}{l} 0.7 \leq \text{Pr} \leq 16,700 \\ \text{Re}_d \geq 10,000 \\ \frac{1}{d_h} \geq 10 \end{array} \right].$$

Finally, the Nusselt number was used to calculate the wall heat transfer coefficient in (18), shown as

$$h_w = \text{Nu}_d \frac{k_v}{d_h}. \quad (18)$$

While the Dittus-Boelter correlation had no upper Reynolds number limit, overprediction for highly turbulent flows was as much as 25% [20]. Because the Reynolds number for steady-state flow was calculated as 7.651E6, the expected error associated with this correlation was high. The Gnielinski Nusselt number correlation [10] has a maximum error of 10%. However, this correlation was only applicable for Reynolds numbers up to 5E6. Overpredictions in the Dittus-Boelter Nusselt number resulted in a higher Biot number and greater conservatism in estimating minimum radial node thickness. Steady-state fluid properties and a conservative Biot number of 0.05 were used for calculations in (16) through (18). The Biot number definition in (2) was rearranged in terms of the radial node thickness in (19), given by

$$t_n = \frac{\text{Bi} \cdot k_w}{h_w}. \quad (19)$$

The number of nodes in the pressure transient TRACE input model was calculated for a wall thickness of 10 mm using (20), given as

$$N = \frac{t_w}{t_n}. \quad (20)$$

The inputs and results of the calculations from (16) through (20) are listed in Table 7.

Table 7. Lumped capacitance calculation results

Parameter	Value
Vapor c_p (J/g·K)	3.2296
Vapor Dynamic Viscosity, μ_v (Pa·s)	1.6884E-5
Pipe Hydraulic Diameter, d_h (mm)	250
Vapor Thermal Conductivity, k_v (W/m·K)	0.044378
Wall Thermal Conductivity, k_w (W/m·K)	57
Reynolds Number, Re_d	7.651E6
Prandtl Number, Pr	1.2287
Nusselt Number, Nu_d	8025.6597
Wall Heat Transfer Coefficient, h_w (W/m ² ·K)	1424.6509
Wall Thickness, t_w (mm)	10
Node Thickness, t_n (mm)	2.0004
Minimum Number of Nodes, N	5
Biot Number, Bi	0.05

Following the lumped capacitance calculations, seven radial nodes were implemented in the TRACE thermal pipe wall model. All TRACE input parameters for the pressure transient model are tabulated in Appendix K.

B. SIMULATION RESULTS

The simulation ran for approximately four minutes for the problem time of 1260 seconds with minimum and maximum timestep sizes of 1E-6 and 1E-2 seconds, respectively. The steady-state flow conditions before and after the pressure transient were identical to the steady-state conditions 98% inlet flow quality simulation in Chapter IV. This relationship indicated that ample time was allotted for both solutions to fully develop and that the addition of the pressure transient did not corrupt the baseline model. All of the transient cases run in this chapter produced a solution within five minutes. The speed of

TRACE for calculating transient solutions was many magnitudes faster compared to CFD codes such as Ansys CFX and Fluent. While the results were not as sophisticated as those produced by three-dimensional codes, TRACE's low speed and computational requirements are optimal for early estimates and investigation. The transient flow characteristics were interrogated in increments of five seconds over the length of the pressure transient. Pressure along the pipe length over time is displayed in Figure 24.

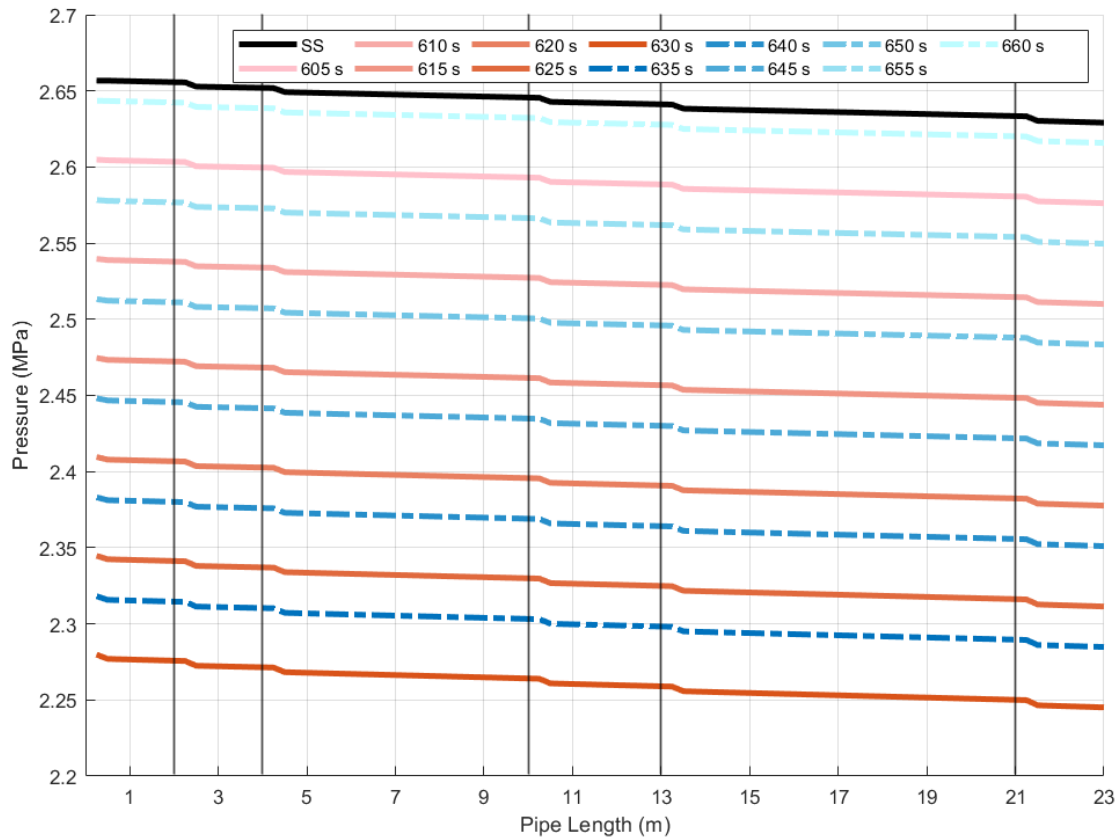


Figure 24. Pressure along pipe for transient pressure with 98% inlet flow quality

The break pressure control signal achieved the desired 15% pressure drop over 60 seconds, beginning at 600 seconds and ending at 660 seconds. The rate of pressure drop along the pipe length with respect to bends and inclination was constant throughout the transient. As with steady-state simulations, the TRACE solver settled at an inlet pressure 16.5 kPa higher than initially specified. This resulted in slightly higher temperatures within

the pipe. The vapor, liquid, and inner and outer wall temperatures are displayed over time in Figure 25 and Figure 26.

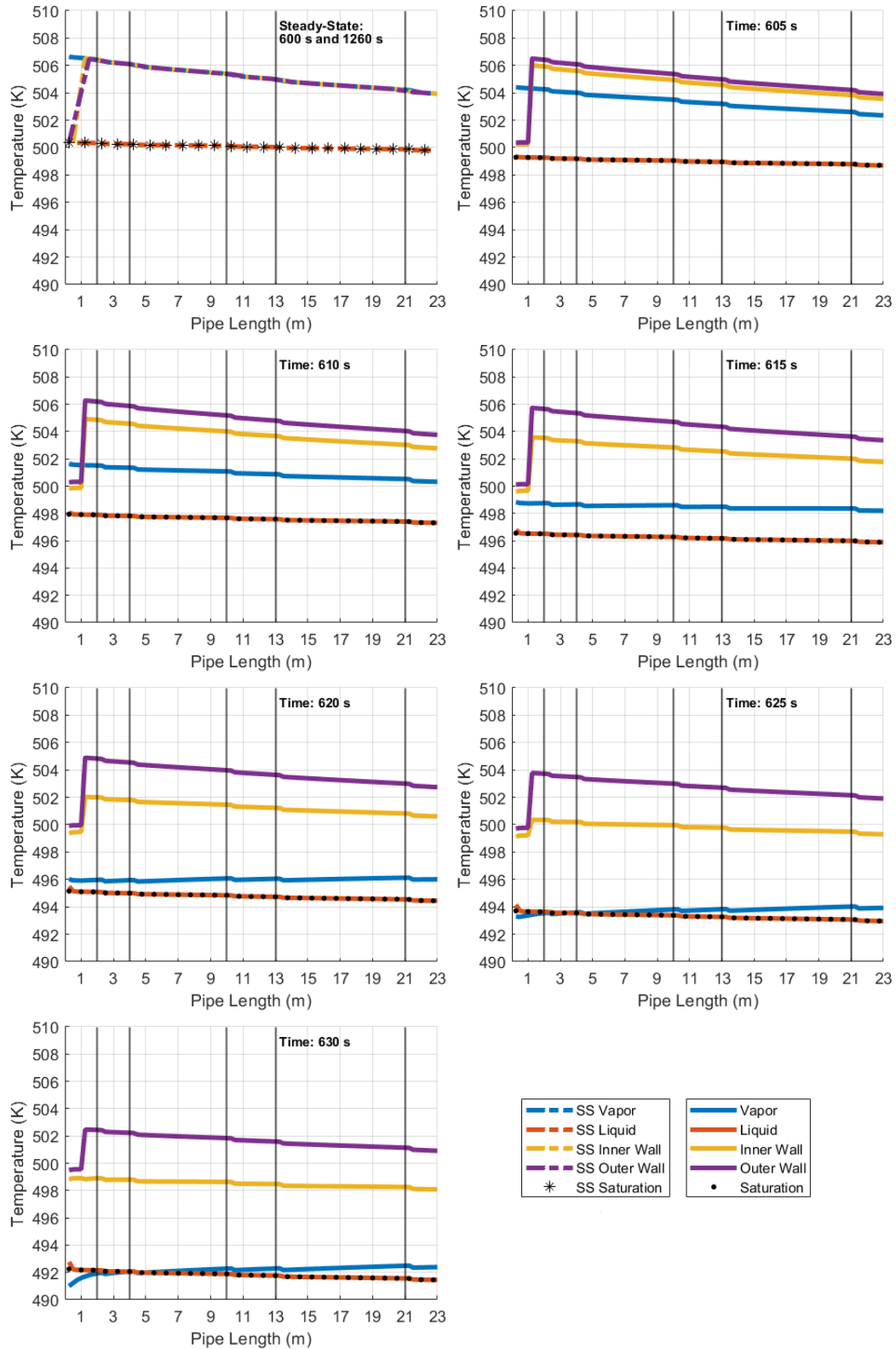


Figure 25. Temperatures along pipe length from 600 to 630 seconds for transient pressure with 98% inlet flow quality

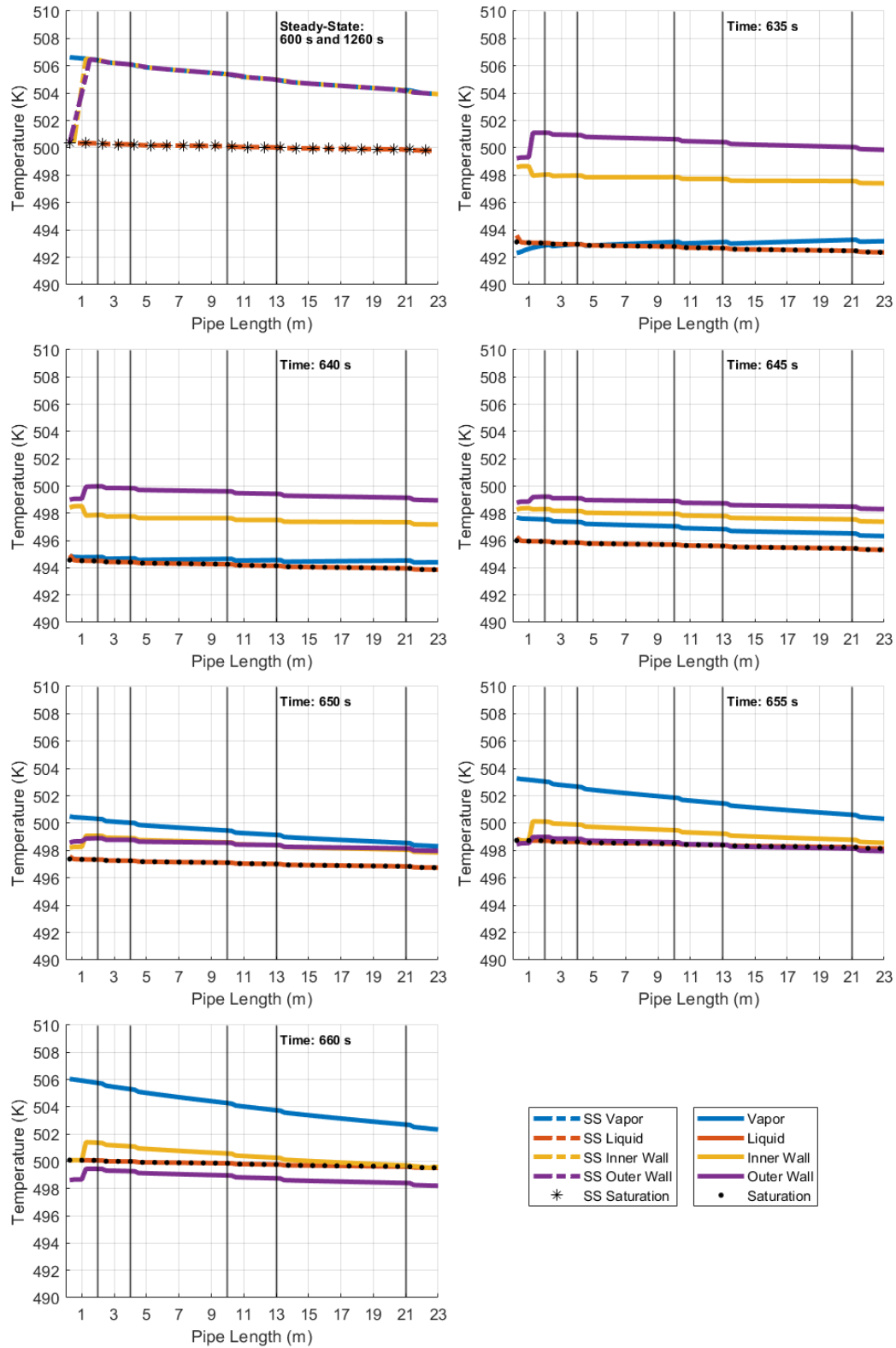


Figure 26. Temperatures along pipe length from 635 to 660 seconds for transient pressure with 98% inlet flow quality

The steady-state vapor and liquid temperature distribution in the pipe mirrored the previous results in Chapter IV. The higher pressure resulted in an additional vapor superheat from the specified 505.89 K to nearly 507 K. The liquid temperature was at the saturation point for the entire transient. All temperatures decreased at the onset of the transient. The vapor temperature led the pipe wall temperatures and approached the liquid and saturation temperatures and reached the liquid and saturation temperatures near the minimum pressure value at 630 seconds. The vapor temperature decreased due to the pressure drop, resulting in a coupled decrease in liquid temperature due to interfacial heat transfer. The inner and outer wall temperatures decreased over time as the latent wall heat transferred to the fluid flow. The outer wall temperature was greater than the inner wall temperature for the first half of the transient, indicating heat flow from the pipe to the fluid. The pipe wall's thermal mass damped the pressure transient's effects. This is illustrated in the lag between the fluid and pipe wall temperature drops over the pressure transient. At 630 seconds, the wall temperature was approximately 7 degrees greater than the fluid temperature. At that time, the vapor temperature near the pipe inlet decreased below saturation conditions for unknown reasons. The inner and outer wall temperatures were also irregular near the pipe inlet, remaining near the specified inlet saturation point and only changing by approximately two degrees over the course of the transient.

During pressure recovery in the second half of the transient, the movement of fluid and wall temperatures relative to one another was similar to the first half of the transient. The increasing pressure raised the vapor temperature, which was followed by the liquid and saturation temperatures. The fluid temperatures approached and surpassed the cooled wall temperature near the end of the pressure recovery. As the temperatures reached steady-state, heat transfer from the fluid flow to the wall brought the system back to steady-state. The vapor temperature drop rate along the pipe length was negative for steady-state conditions and became more positive over the transient pressure drop. At 630 seconds, the vapor temperature increased along the pipe length due to heat transfer from the wall. Following the pressure recovery, the vapor temperature along the pipe length returned to the initial distribution. The heat flux from the inner pipe wall to the fluid is shown over the course of the pressure transient in Figure 27.

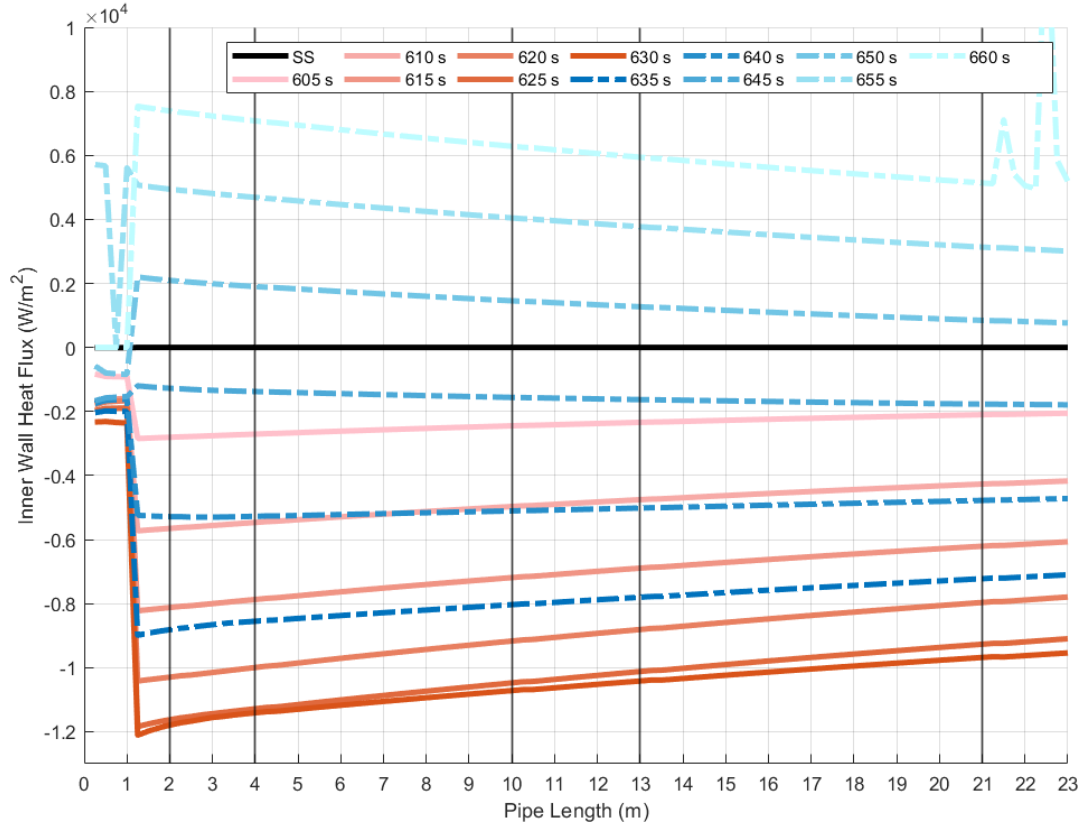


Figure 27. Inner wall heat flux along pipe length for transient pressure with 98% inlet flow quality

The heat flux from the pipe wall increased from zero along the pipe length in the first half of the pressure transient as the fluid temperature dropped and pulled the latent heat from the pipe wall, indicated by the negative heat flux value. At the beginning of pressure recovery, the heat flux out of the wall decreased until just after 645 seconds when vapor temperature surpassed the wall temperature and the heat flux became positive. From the beginning of the transient, there was a 12 second lag before the pipe wall began to reject heat to the fluid flow. The heat flux magnitude decreased along the length of the pipe for the duration of the transient due to lower temperature differences between the fluid and pipe wall. Many non-physical features were seen in the heat flux. The rate of heat flux change was not affected by the pipe bends or changing phase distribution along the pipe. Large spikes in wall heat flux were seen near the pipe inlet and outlet over the last 15 seconds of the transient. The maximum timestep size was decreased from 1E-2 to 1E-4 to

remove this non-physical flow behavior. This change had no effect. Additional non-physical flow features were seen with the transient heat transfer coefficient, shown in Figure 28.

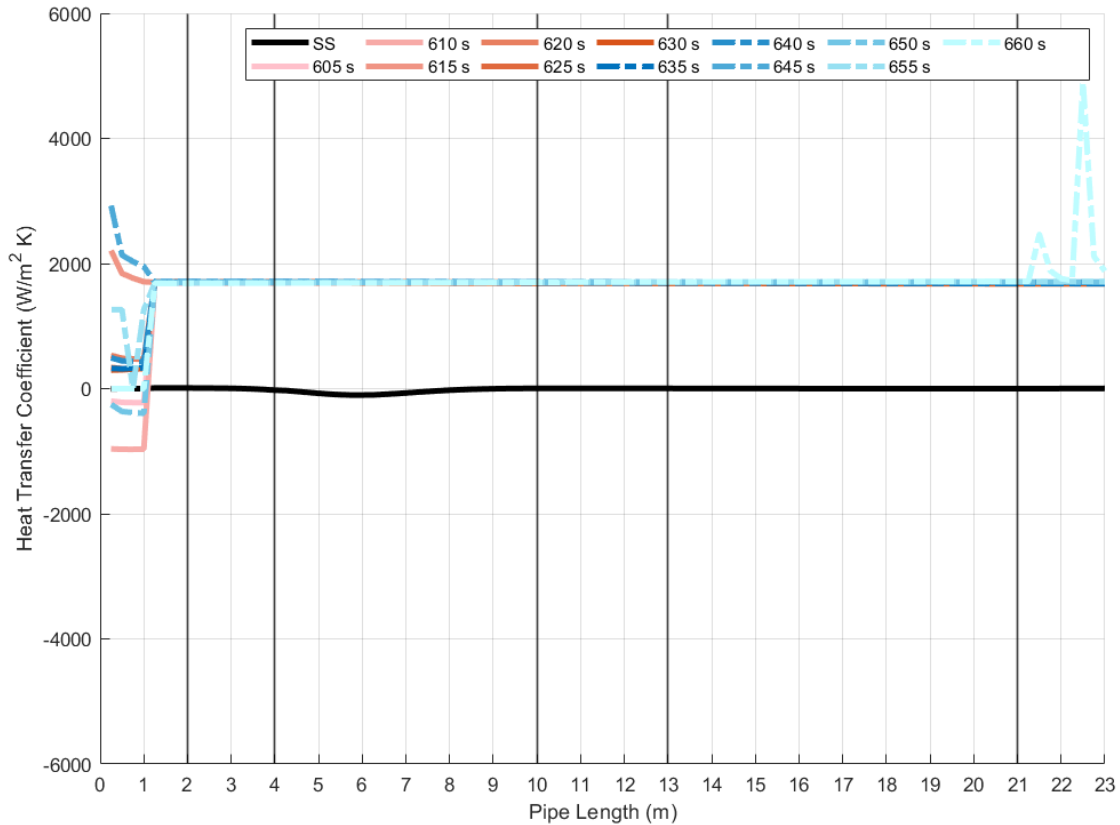


Figure 28. Wall heat transfer coefficient along pipe length for transient pressure with 98% inlet flow quality

As discussed in Chapter II., the heat transfer coefficient in pipe flow was heavily influenced by the vapor mass fraction, heat flux, and flow regime. The wall heat transfer coefficient was expected to vary over with respect to pipe geometry and pipe length. The TRACE solver calculated a constant heat transfer coefficient of approximately 1600 W/m²K over the entire pipe length and transient. At steady-state, when there was no heat flux between the pipe wall and fluid, the heat transfer coefficient was zero as expected. However, it bowed downwards just after the second bend. This feature still present for additional simulations with excessively low timestep residual thresholds. As with heat flux,

instabilities were seen at the pipe inlet and outlet near the end of the transient. The fill boundary conditions caused jumps in the heat transfer coefficient near the inlet, similar to those seen with the transient wall temperatures. Because TRACE was designed to model reactor accidents, the heat transfer coefficient correlations that are most developed in the code are tailored for the coolant flow over the rods in a reactor vessel [15]. This heat transfer is characterized as pre-CHF, with a very low vapor mass fraction. The dry annular boiling region applicable in this transient was beyond the intended use of the TRACE code, resulting in a single heat transfer coefficient over the entire transient case. The flow quality with respect to pipe length and simulation time is displayed in Figure 29.

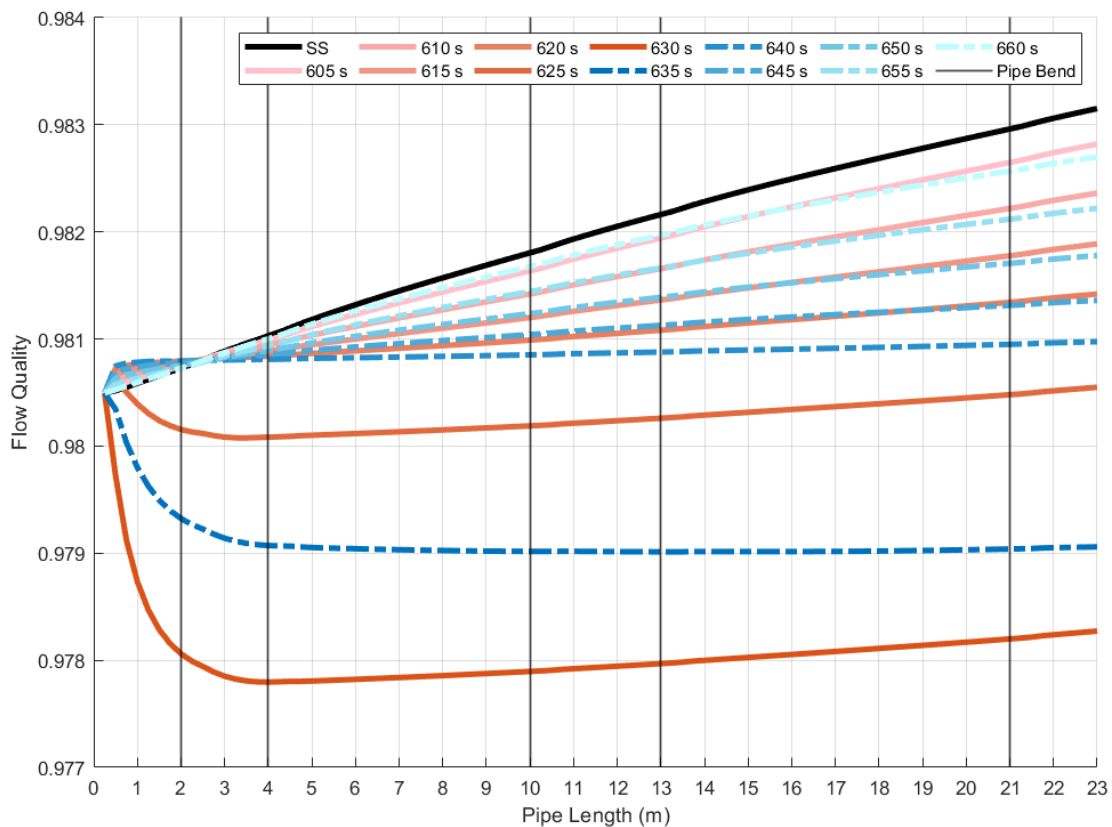


Figure 29. Flow quality along pipe length for transient pressure with 98% inlet flow quality

The flow quality near the inlet was 98% during the entire transient because the fill vapor volume fraction was unaffected by the break pressure. For steady-state and transient

conditions, the liquid mass in the flow decreased through mass transfer to the vapor field and moisture collection in the pipe bends. These mechanisms caused a positive rate of flow quality change with respect to pipe length. At lower transient pressures, this rate was lower in magnitude. The flow quality was nearly constant along the pipe length from 630 to 640 seconds, just after the onset of pressure recovery. The flow was further interrogated by examining the vapor and liquid mass flow rates in Figure 30 and Figure 31.

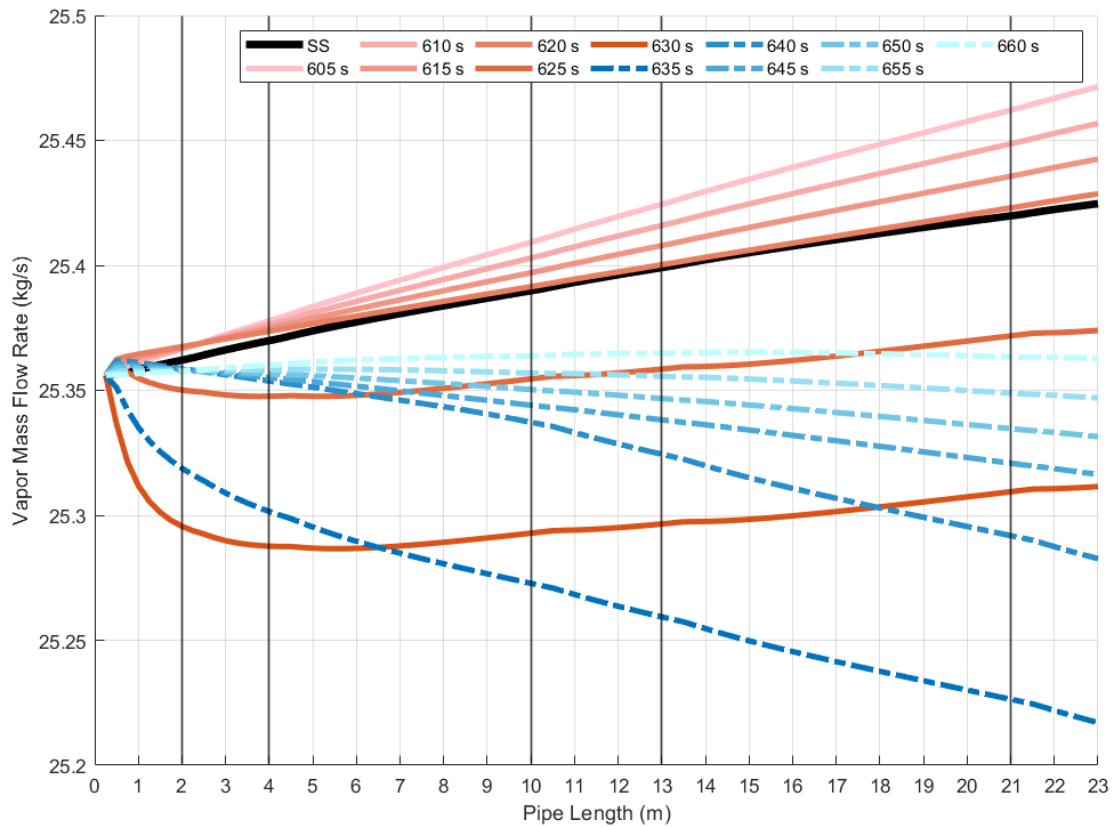


Figure 30. Vapor mass flow rate along pipe length for transient pressure with 98% inlet flow quality

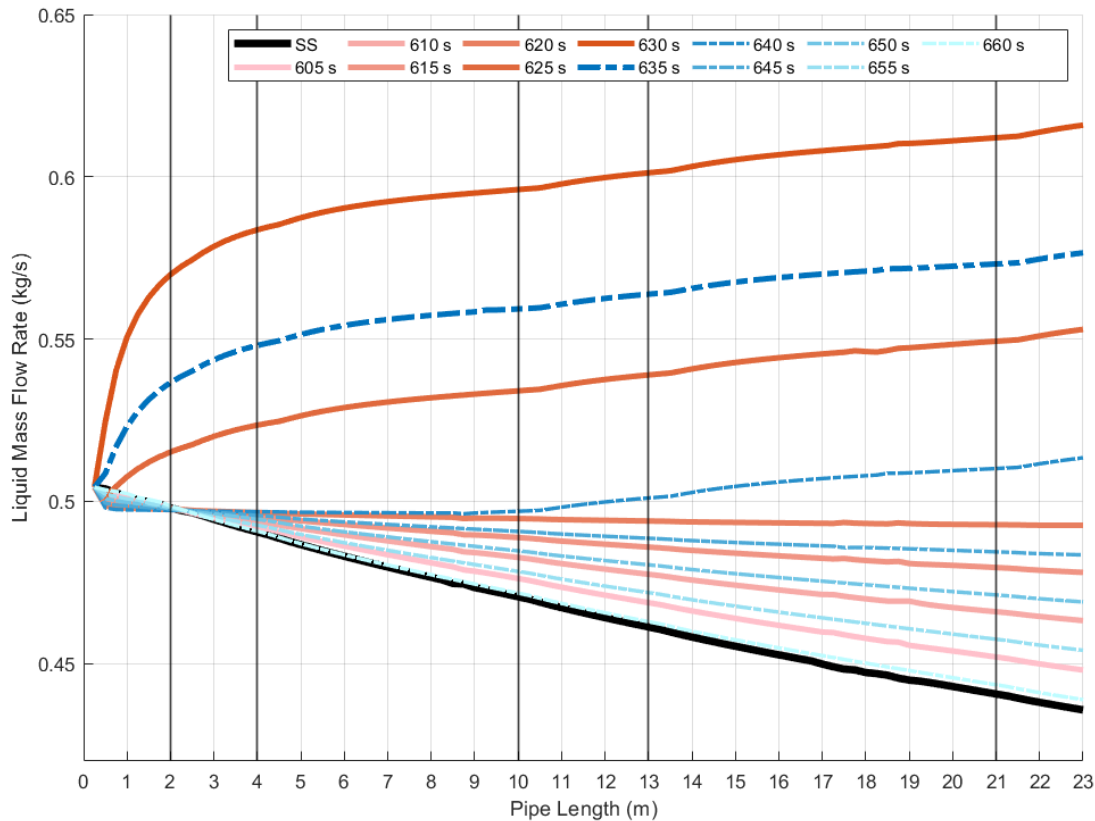


Figure 31. Liquid mass flow rate along pipe length for transient pressure with 98% inlet flow quality

The vapor and liquid mass flow rates reflected many of the flow characteristics seen in Figure 29. While the flow quality increased along the pipe length over the entire transient, vapor mass flow only increased over the pipe length when the pressure decreased. The vapor mass flow rate was lower towards the pipe outlet during pressure recovery. As in the steady-state simulations, the fluid mass flow rates and flow qualities only represented fluid flow across the areas at cell interfaces. Figure 32 displays the ratio of vapor mass to total mass in each pipe cell throughout the transient.

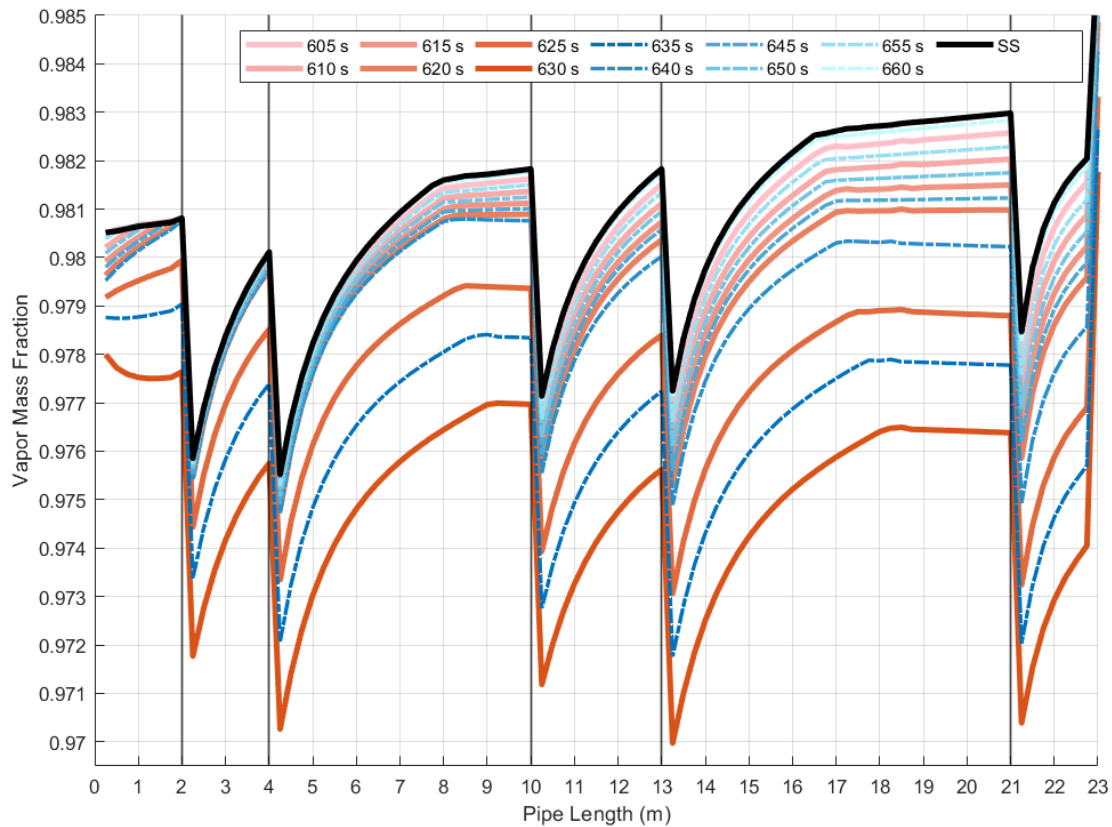


Figure 32. Vapor mass fraction along pipe length for transient pressure with 98% inlet flow quality

As expected, the liquid mass collecting around the pipe bends increased with the decreased flow quality, reaching a maximum at the bottom of the pressure transient. While the distribution pattern of stagnant liquid along the pipe length stayed constant over the course of the transient, the vapor mass fraction near the pipe bends were approximately 16% lower than at steady-state. The total mass of each phase and the total fluid mass in the pipe over time are shown in Figure 33.

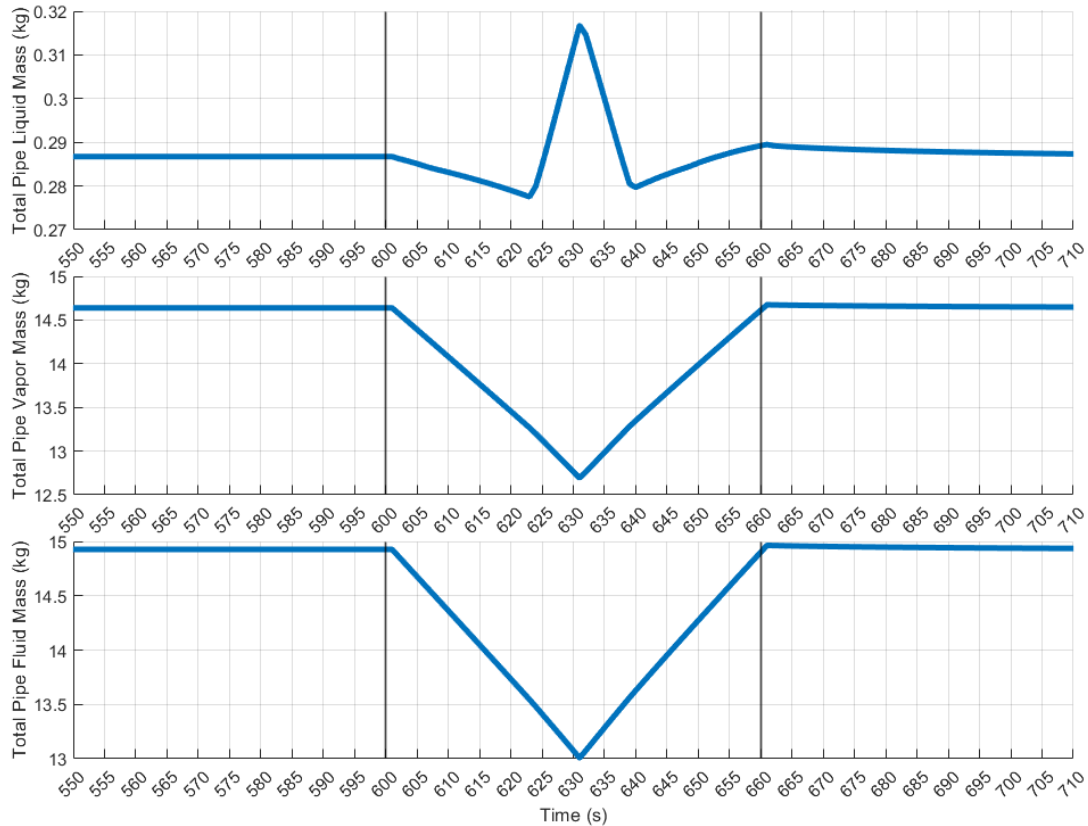


Figure 33. Total pipe fluid mass along pipe length for transient pressure with 98% inlet flow quality

The change in total pipe liquid mass was primarily influenced by pressure during the initial and final stages of the transient. The pipe wall's thermal mass and the vapor superheat acted as a buffer, resulting in a phase change lag. From 623 to 637 seconds, the total liquid mass change was primarily influenced by temperature. The liquid in the pipe began to boil back to vapor just after the bottom of the pressure transient. Phase change did not significantly affect the drop and recovery of vapor mass within the pipe. The vapor density decreased with the pressure drop, resulting in a lower vapor mass within the pipe's volume and vice versa.

The pipe wall was removed from the model and the results were re-evaluated to isolate the effects of the latent pipe wall heat on phase change. The exit flow qualities over time for the cases with and without the pipe wall are compared in Figure 34.

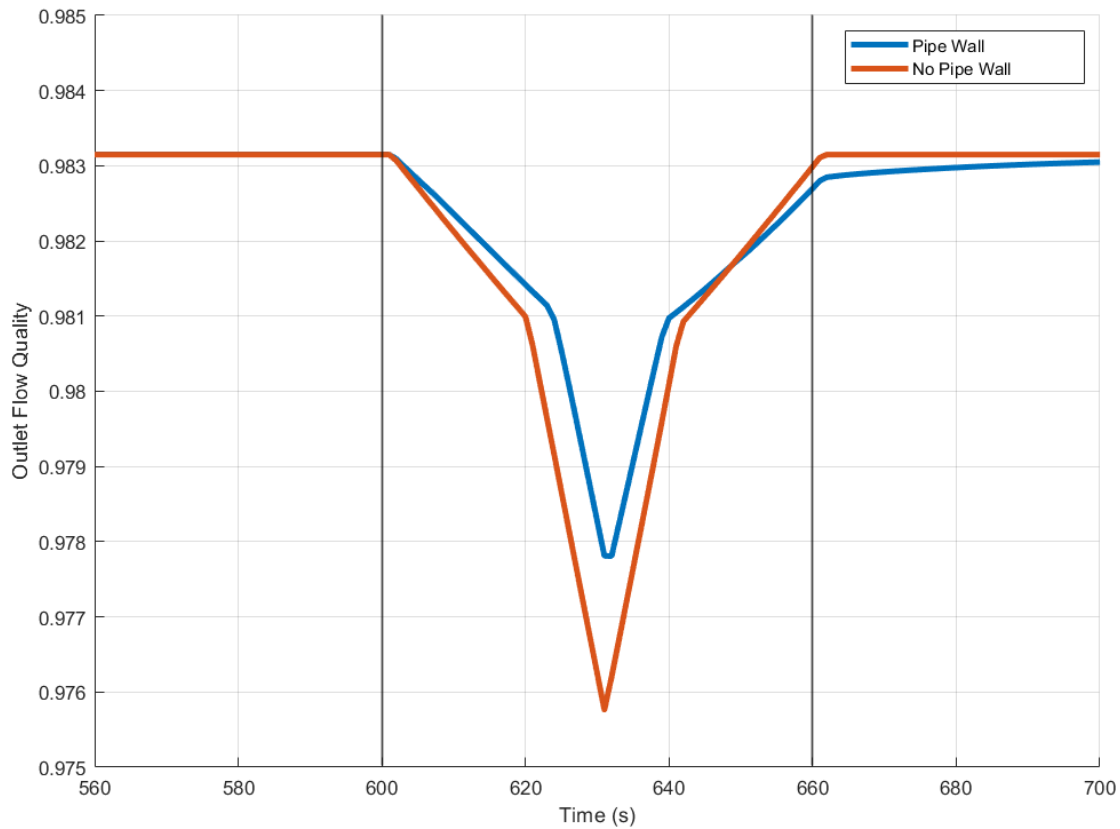


Figure 34. Pipe wall influence on outlet flow quality over pressure transient

The addition of the pipe wall resulted in a lesser drop in flow quality at the bottom of the pressure transient. The heat in the pipe wall delayed the transient effects and increased the recovery time. The change in slope during the transient for both cases was due to the vapor reaching saturation temperature and allowing for a greater rate of phase change. This affect was reversed during the latter half of the transient.

VI. VAPOR MASS FRACTION TRANSIENT TRACE MODELING

This effort's second transient of interest was similar to the simulation in the previous chapter. Rather than pressure, the fill vapor volume fraction was lowered at a constant rate, resulting in a 15% drop in fill flow quality over 30 seconds and a subsequent recovery over an additional 30 seconds to the steady-state vapor volume fraction corresponding to an initial inlet flow quality of 98%. The transient vapor volume fraction is illustrated in Figure 35.

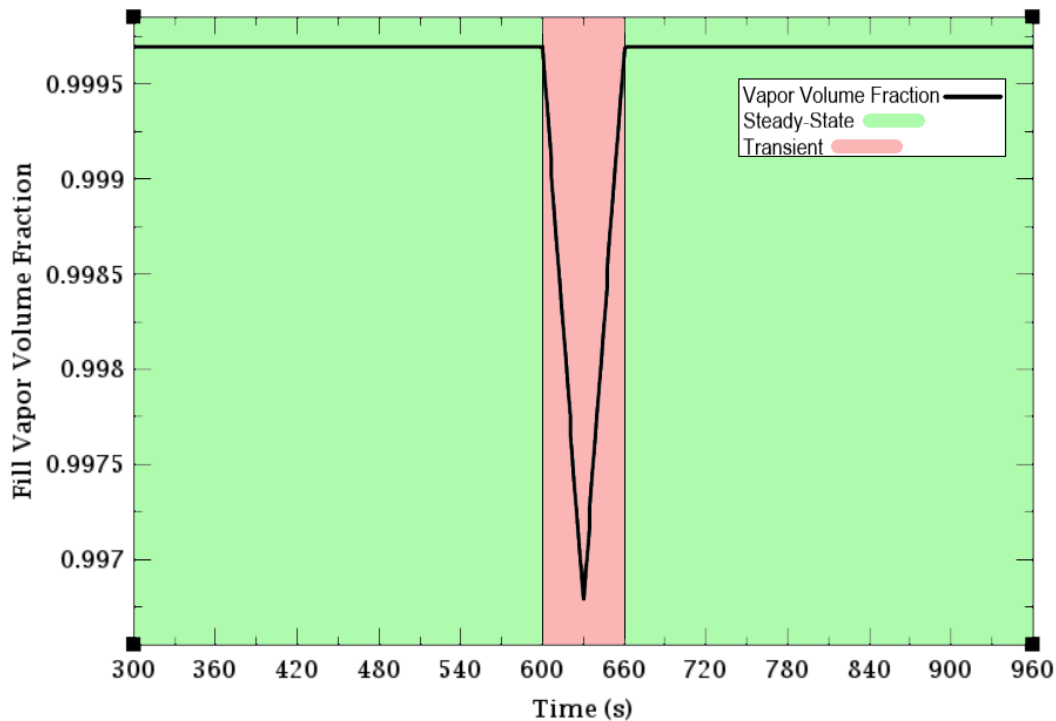


Figure 35. Vapor volume fraction input signal

A. MODEL DEVELOPMENT

The vapor volume fraction transient model was identical to the pressure transient model, except for the varying flow parameter. For this model, the input pressure was held constant while the vapor volume fraction was replaced by a control signal which varied

based on the elapsed problem time. A diagram of the control signal components is shown in Figure 36.

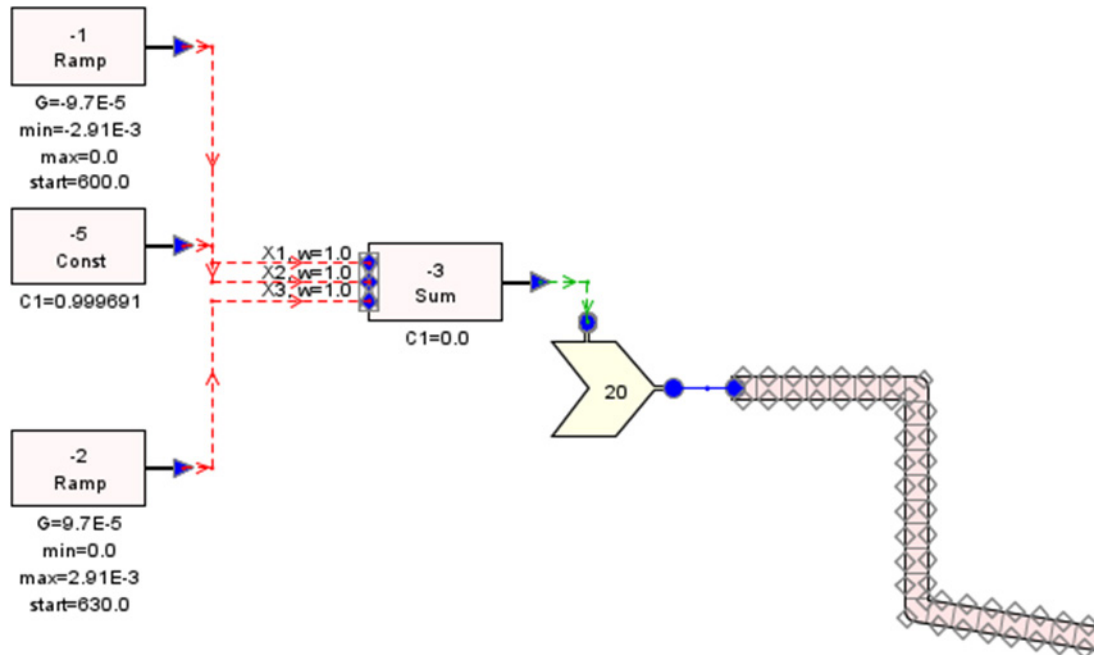


Figure 36. Control signal components for transient vapor volume fraction

This control signal was equal to the sum of one constant signal and two ramp function signals. The constant signal was set to 0.999691 for all timestep values, representing the vapor volume fraction for an input flow quality of 98%. The upper ramp function signal was on for time step values above 600 seconds with a gain equal to the vapor volume fraction drop rate. The lower ramp function signal had an equal and opposite gain and was on for timestep values greater than 630 seconds. The sum of these three control blocks produced a signal for break pressure identical to the signal in Figure 36. The minimum vapor volume fraction value corresponded to an inlet flow quality of 83%. The fill boundary conditions were set to retrieve inlet mass flow rates based on the fluid velocity inputs and changing vapor volume fraction. A complete report of the initial vapor mass fraction transient TRACE model is tabulated in Appendix L.

B. SIMULATION RESULTS

Before the simulation results were fully interrogated, the model was evaluated within the TRACE analysis package. The TRACE solver decreased the pressure during the vapor volume fraction transient, as seen in Figure 37.

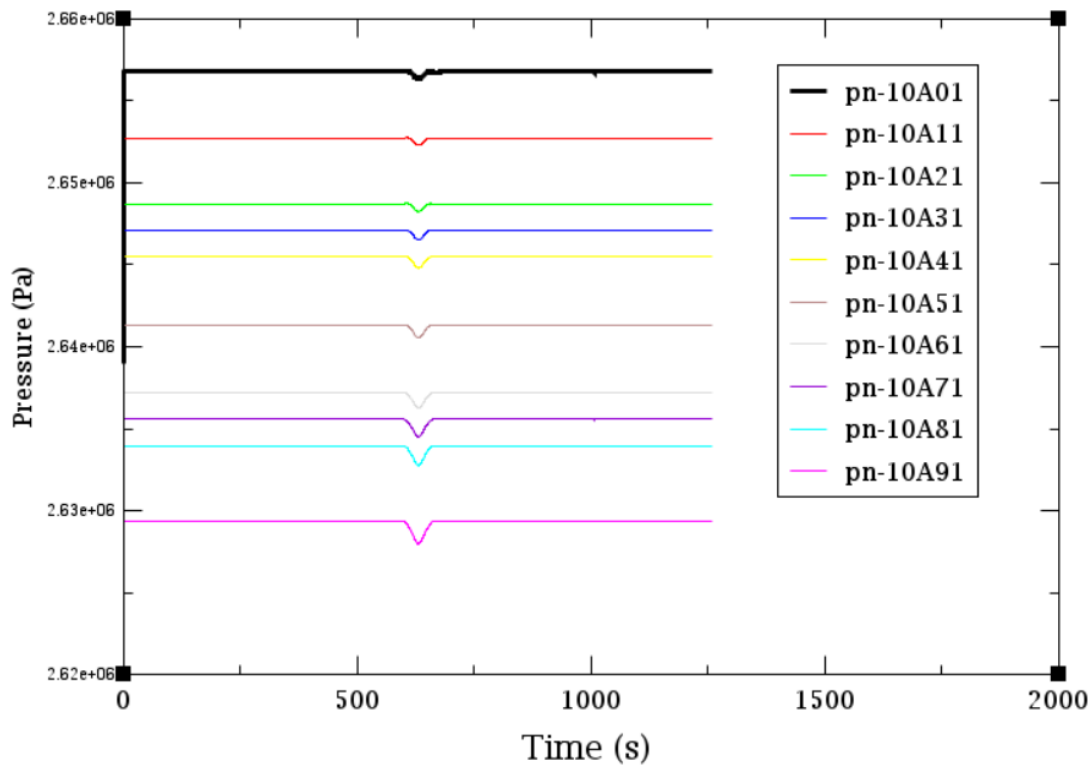


Figure 37. Pipe pressure over time in cell increments of 10

In this transient case, the vapor volume fraction was to decrease at a constant specified pressure. The unwanted pressure changes in the simulation resulted in a varying saturation temperature and influenced the moisture distribution within the pipe. Several changes to the TRACE input model were implemented independently to mitigate this unwanted effect.

Rather than controlling the vapor volume fraction at the inlet, a new control signal was developed to adjust the liquid and vapor mass flow rates and produce the same inlet flow quality transient. This method produced the desired vapor volume fraction transient

but did not eliminate the pressure transient. The fill boundary conditions were changed to retrieve transient fluid inlet velocities from steady-state mass flow rates. This change forced the fluid velocity at the inlet to zero. The other fill type available in TRACE for transient simulations assumed an interfacial slip of zero, modeling the vapor and liquid with a single velocity and total mass flow rate. Because previous simulations indicated that both of these parameters change independently over the pipe's length and the transient's duration, this inlet type was not appropriate. Constant value control signals were added to fix the fill pressure over the course of the transient. This change overconstrained the solution, and the run failed. The pressure at the break and the pressure drop over the pipe were fixed in two independent runs, both resulting in an over-constrained problem.

This effort was unsuccessful in modeling cases with transient quantities of liquid in the inlet flow without a coupled change in the pipe pressure. TRACE was developed as a system analysis code for large-scale models with many components. In such systems, components such as the pipe have black box characteristics. Because the component inputs and outputs, and computational efficiency, are more important on a macro scale, the fidelity of modeled fluid dynamics internal to the component is less crucial. With the available modeling options, TRACE was unsuitable for simulating the effects of vapor volume fraction transients within a pipe component.

VII. PRESSURE OSCILLATION CASE STUDY

Flows featuring high pressure, temperature, and velocity are extremely challenging to interrogate experimentally due to the harsh environment that limits data collection instrumentation. The void fraction is particularly difficult to measure in such cases. As a result, little open literature experimental data was available for comparison with this effort. An experiment conducted by Guo et al. [14] had similar physical characteristics as the TRACE simulations investigated in this study. This effort modeled the Guo et al. experiment in TRACE and compared the simulation results to the experimental data. Two-phase data was taken from Guo et al., digitized, and plotted in Figure 38.

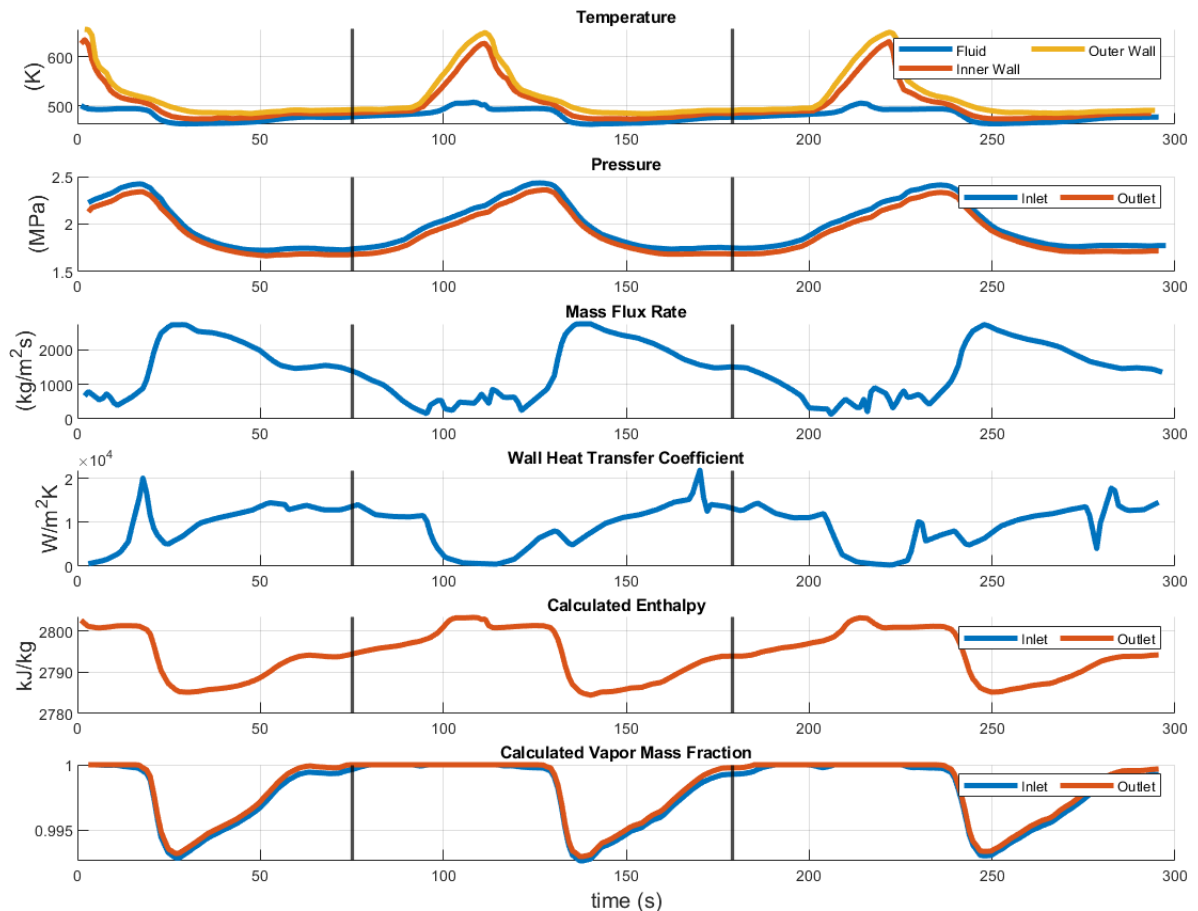


Figure 38. Guo et al. digitized two-phase flow pressure oscillation data.
Adapted from [14].

The enthalpy and static quality in the flow test section were estimated using the IAPWS IF97 data and X Steam in MATLAB. Because the Guo et al. two-phase experiments had a very high void fraction, fluid enthalpy was approximated for a saturated vapor and calculated using measured pressures at the inlet and outlet of the test section. Quality was calculated at the test section inlet and outlet using these enthalpy values and the corresponding pressure data. The quality was calculated as 99.96% at steady-state and approximated as 100% in the initial model. A single period from the pressure oscillations was approximated as the section of the data bounded in black lines in Figure 38, with steady-state parameters taken just before the beginning of the transient. One pressure oscillation period was approximated as 104 seconds. Complete documentation for these calculations is included in Appendix G.

A. MODEL DEVELOPMENT

Steady-state flow parameters from Figure 38 were implemented in TRACE and listed in Table 8.

Table 8. TRACE input parameters for Guo et al. comparison. Adapted from [14].

Input Parameter	Component		
	Fill (inlet)	Pipe	Break (outlet)
Initial Flow Quality, X (%)	99.96-100	100	100
Initial Pressure, p (MPa)	1.692	1.692	1.692
Transient Pressure Differential, Δp (MPa)	-	-	0.6708
Transient Time Duration, Δt (s)	104	104	104
Liquid Velocity, u_l (m/s)	110	-	-
Vapor Velocity, u_v (m/s)	110	-	-
Initial Liquid Temperature, T_l (K)	477.22	477.22	477.22
Initial Vapor Temperature, T_v (K)	477.22	477.22	477.22
Initial Wall Temperature, T_w (K)	-	477.22	-
Wall Power, P_w , (kW)	-	100	-
Material	-	304 SS	-
Absolute Roughness, ϵ (m)	-	1E-4	-
Wall Thickness, t_w (mm)	-	2	-
Pipe Length, l (m)	-	6.448	-
Hydraulic Diameter, d_h (mm)	-	15	-
Axial Nodes	1	100	1
Radial Nodes	-	8	-

Because the helix shape of the test section was horizontally oriented, no elevation change was modeled in TRACE. Form losses due to flow turning within the helix shape were also neglected. Guo et al. did not specify a test section material. This simulation was modeled using 304 stainless steel with a conservative eight radial nodes and an absolute roughness of 1E-4 m. The experimental setup designed by Guo et al. included 100 kW of ohmic heating in the wall of the test section. This heat flux was modeled with a constant power input dispersed across the pipe component in TRACE. Because the wall was uniformly heated and well insulated, axial wall heat transfer was neglected.

The isolated experimental pressure oscillation was approximated with a control signal similar to those developed for previous transient models in this effort and displayed in Figure 39.

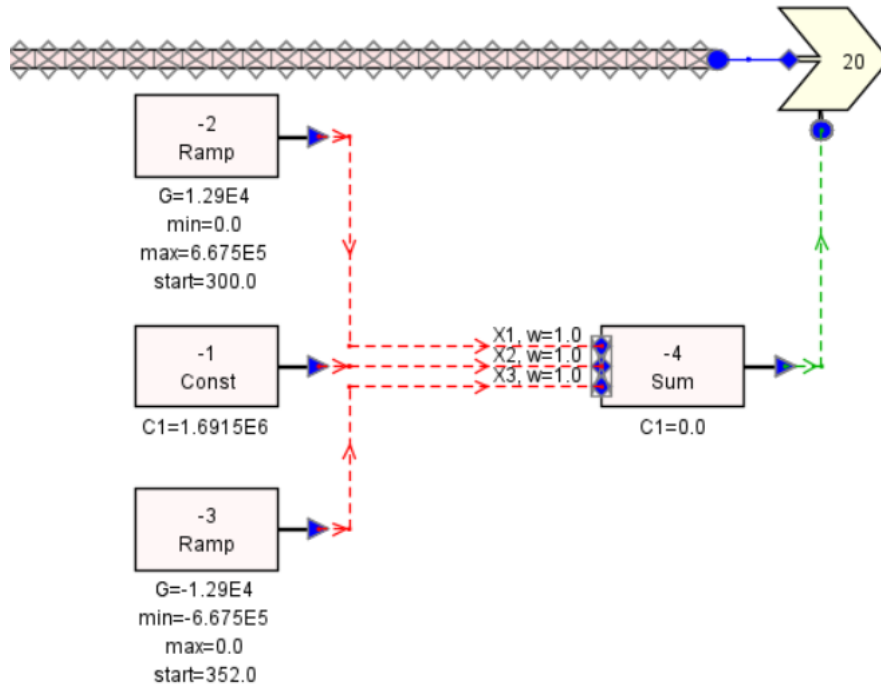


Figure 39. Guo et al. TRACE comparison pressure control signal

This control signal was equal to the sum of one constant signal and two ramp function signals. The constant signal was set to 1.6915×10^6 for all timestep values, representing the steady pressure input in Pascals. The upper ramp function signal was on for time step values above 300 seconds with a gain equal to the pressure increase rate. The lower ramp function signal had an equal and opposite gain and was on for timestep values greater than 352 seconds. The sum of these three control blocks produced a 104 second transient signal for break pressure identical to Figure 40.

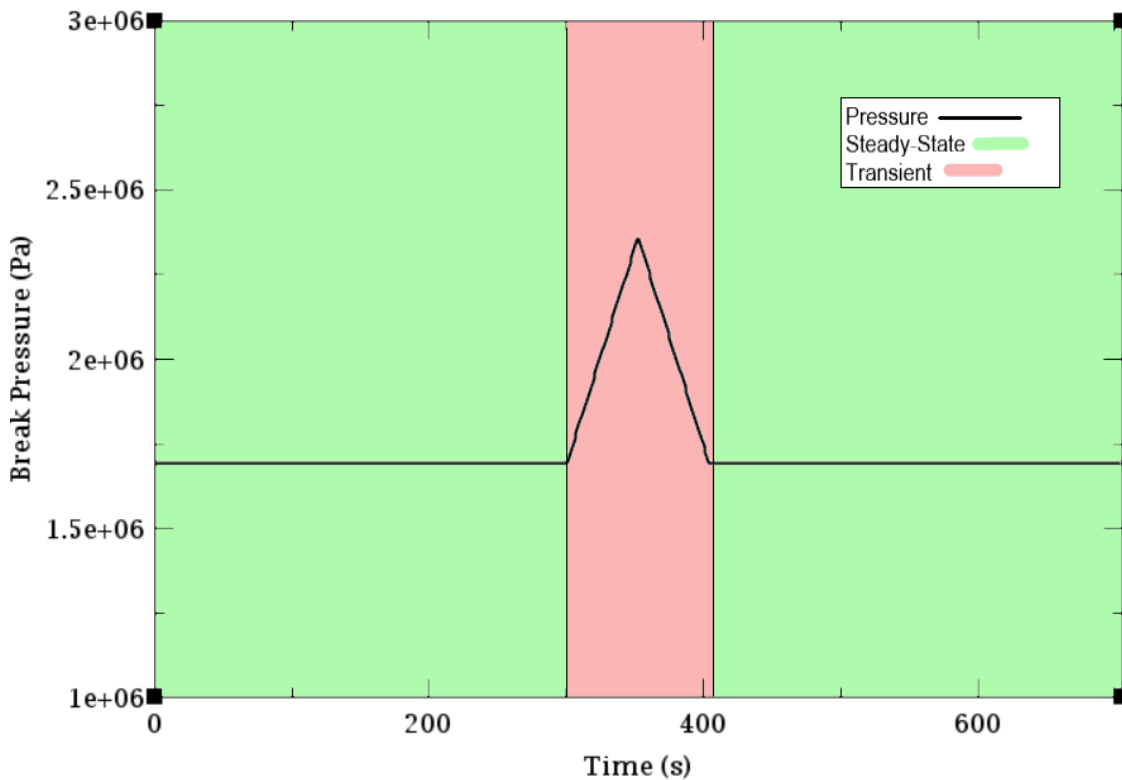


Figure 40. Transient pressure input for Guo et al. comparison

The TRACE model was run for minimum and maximum timestep sizes of 1E-6 and 1E-2 seconds and compared with a single period of the Guo et al. data, shown in Figure 38. A complete report of the initial Guo et al. TRACE comparison inputs is attached in Appendix M. A guide for basic model implementation using the TRACE user interface is attached in Appendix N.

B. SIMULATION RESULTS AND COMPARISON

As indicated in the steady-state simulations, TRACE was ineffective in simulating transient cases with an input flow quality higher than 98%. Because the quality calculated from the experimental data in this comparison case ranged from 99.4 to 100%, the simulation results were expected to show non-physical trends. While only one case study was conducted, updating the TRACE model to simulate the experimental data more effectively was iterative, and the effects on the flow parameters brought about by each change were equally important as the results of the final model. The first simulation

discussed in this chapter was implemented as described in the development of this model. The second model was identical to the first but with the wall roughness removed. In the final simulation, the roughness was also neglected. The fill vapor volume fraction was changed to model an inlet flow quality of 99.96% rather than the 100% modeled in the first two simulations. All simulations concluded after approximately two minutes of run time. The flow parameters were compared with the experimental data for the three TRACE model iterations. For all results, the transient initiation and termination were indicated with black vertical lines.

1. Pressure Comparison

The initial TRACE model vastly overpredicted the pressure differential across the test section for both the transient and steady-state portions of the simulation, as displayed in Figure 41.

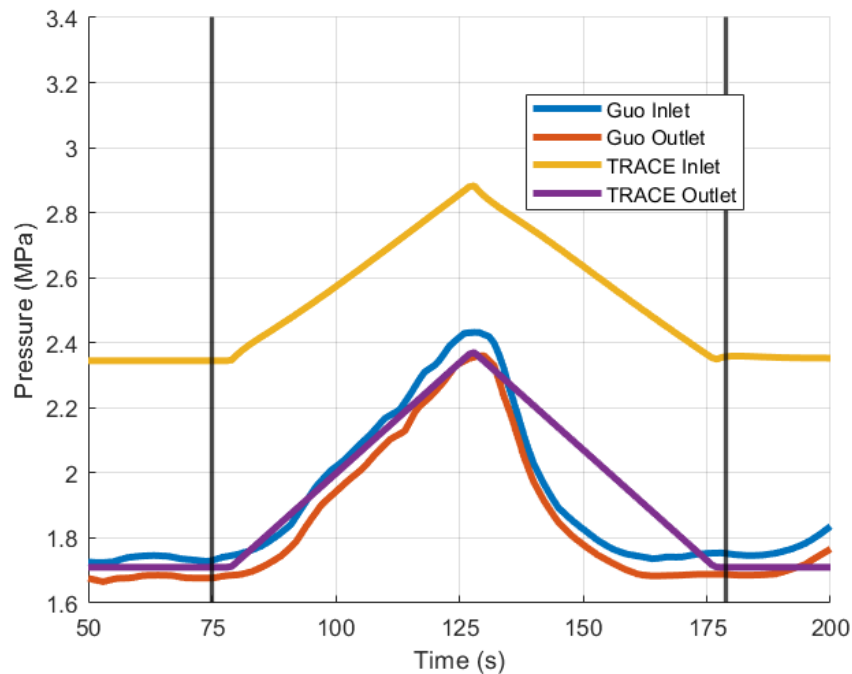


Figure 41. TRACE initial model and Guo et al. pressure comparison

While the magnitude was overpredicted, TRACE adequately predicted the rate of pressure change over time for the transient portion of the simulation. The wall roughness

was removed from the TRACE model to eliminate the pressure differential overprediction. As indicated in Figure 42, this change produced the desired effects.

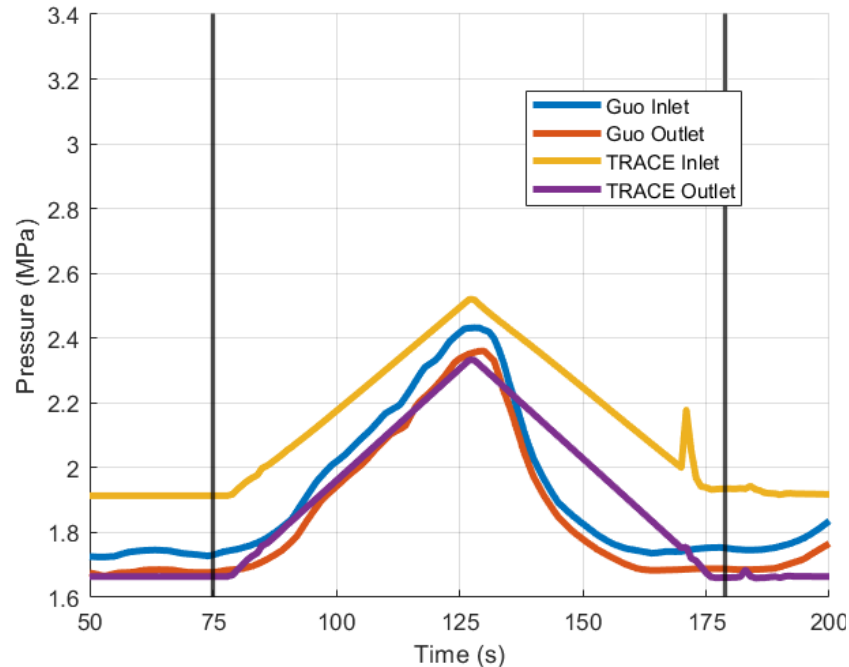


Figure 42. TRACE smooth pipe model and Guo et al. pressure comparison

Removing roughness from the pipe model had unexpected, non-physical effects, including a significant spike in the inlet pressure at 120 seconds, during the latter half of the transient. The fill flow quality change from 100% to 99.96% in the TRACE model changed the intensity and pattern of instabilities in the pressure. As shown in Figure 43, the sharp spike seen in the 100% fill flow quality model was replaced by dispersed instabilities with lower amplitudes near the peak of the transient at 130 seconds. This tiny change in specified flow quality at the fill produced a disproportionately large response in the calculated pressure that would not exist in a real system.

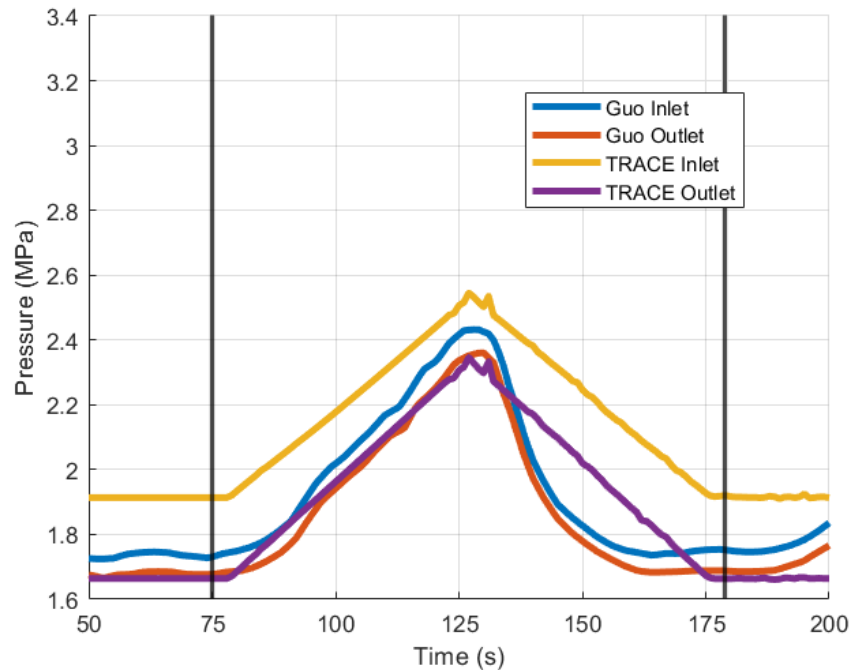


Figure 43. TRACE smooth 99.96% flow quality pipe model and Guo et al. pressure comparison

2. Fluid Temperature Comparison

The vapor and liquid temperatures from the initial TRACE model are shown with the experimental fluid temperature data in Figure 44. The initial pressure differential overprediction led to a coupled increase in vapor temperature. The liquid temperature was indirectly affected due to interfacial heat transfer between the two phases. As observed in previous simulations, TRACE did not model wall heat transfer to a high degree of fidelity, leading to much more uniform trends in temperature changes over the course of the transient.

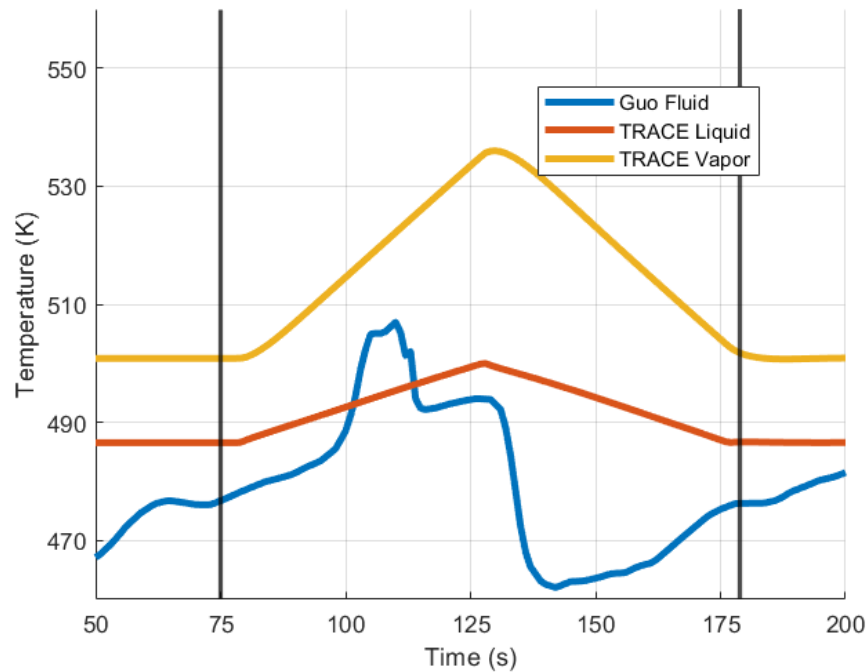


Figure 44. TRACE initial model and Guo et al. fluid temperature comparison

Experimental temperature measurements did not distinguish between fluid phases. Because the calculated flow quality was nearly 100% for the entire transient, the TRACE vapor temperature was expected to match the experimental fluid temperature. The rate of transient temperature change was similar for experimental and simulated data. However, TRACE overpredicted the instantaneous fluid temperature values. The removal of wall roughness in the second TRACE run resulted in a marginally improved agreement between the TRACE results and experimental data, shown in Figure 45.

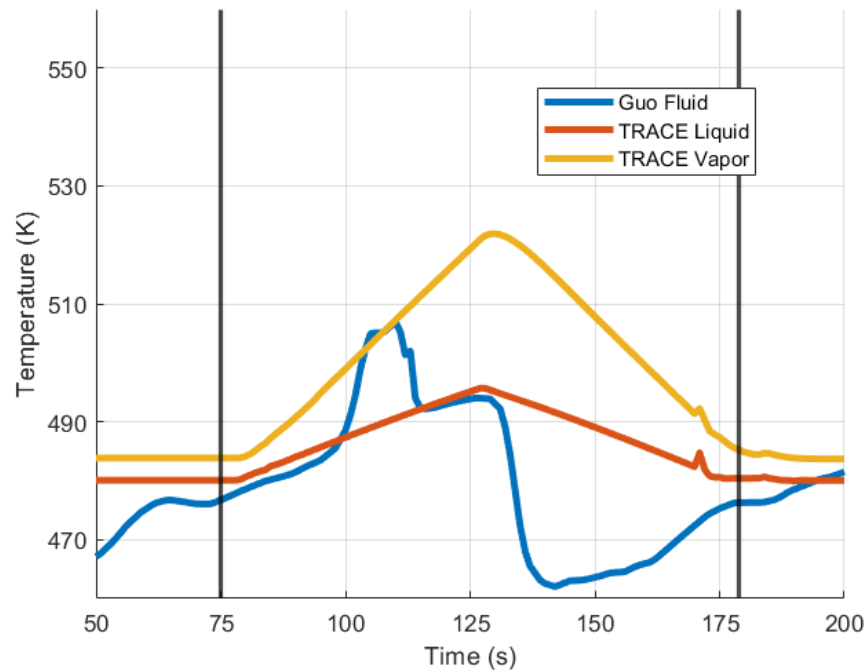


Figure 45. TRACE smooth pipe model and Guo et al. fluid temperature comparison

The smooth TRACE pipe model nearly halved the temperature overprediction, reducing the vapor temperature by 15 K at the peak of the transient. This change did not affect the rate of temperature change. The temperature spike seen in Figure 41 was also present in the vapor and liquid temperatures during the latter half of the transient. The change in fill flow quality has similar effects on the fluid temperatures as with the pressure. As seen in Figure 46, the spike at the end of the transient was replaced by dispersed instabilities at the peak of the transient.

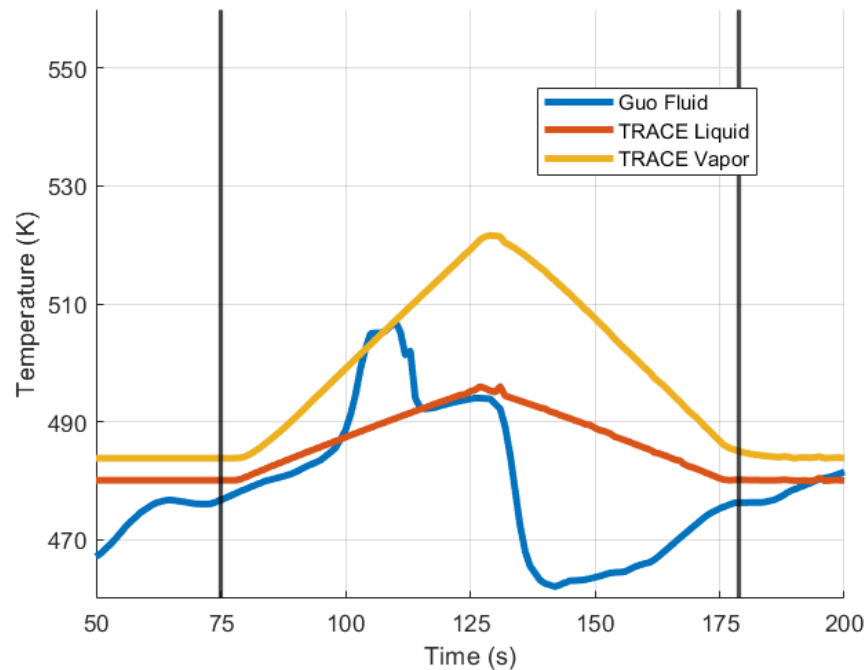


Figure 46. TRACE smooth 99.96% flow quality pipe model and Guo et al. fluid temperature comparison

3. Wall Temperature Comparison

Figure 47 illustrates the inner and outer wall surface temperatures predicted by TRACE and the corresponding experimental data. The TRACE solver did not predict a significant temperature gradient within the pipe wall. The TRACE wall temperature was lower than the experimental data during the steady-state portion of the study. During the pressure transient, the TRACE wall temperature did not increase at the rate or the extent seen in the experimental data.

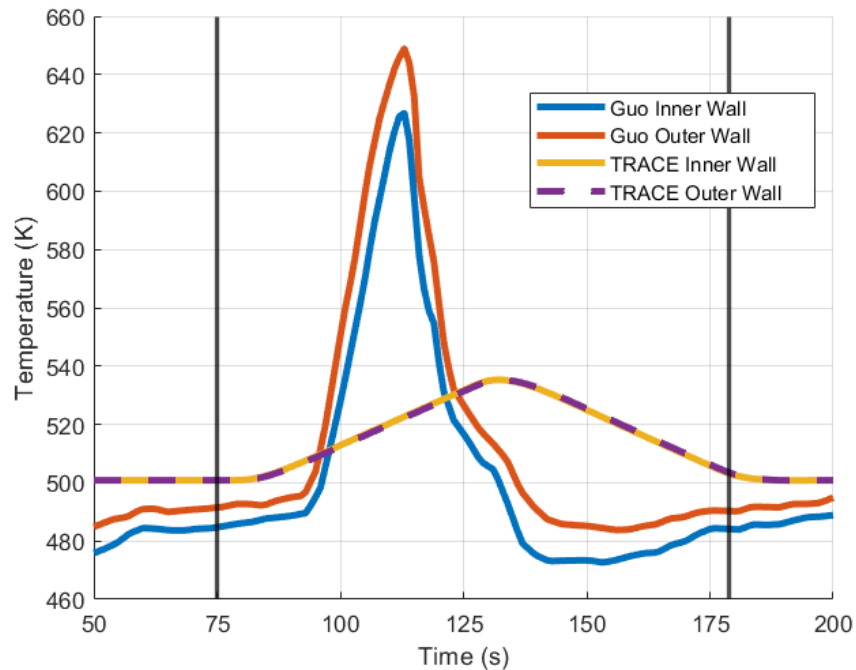


Figure 47. TRACE initial model and Guo et al. wall temperature comparison

Because the vapor temperature was much higher in the TRACE simulation, this lower wall superheat was likely due to a deficiency in heat transfer between the fluid and the wall in TRACE. Removing roughness from the second TRACE run resulted in an 18 K decrease in inner and outer wall temperatures over the entire simulation. Figure 48 shows excellent agreement between the steady-state regions' experimental and simulated wall temperatures. The transient wall temperature behavior was unchanged in the second TRACE run.

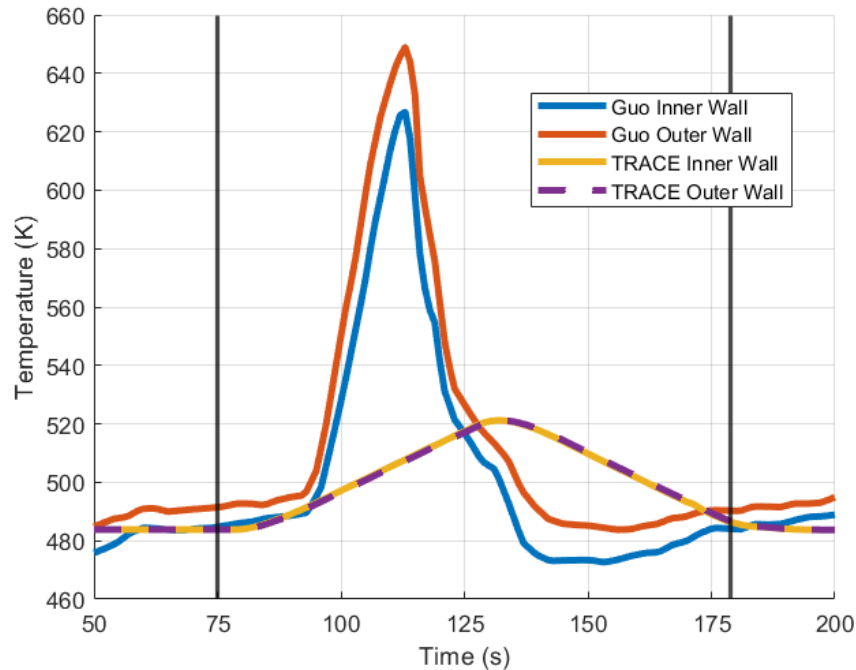


Figure 48. TRACE smooth pipe model and Guo et al. wall temperature comparison

The change in fill flow quality from 100% to 99.96% did not produce any noticeable effect in the simulated wall temperatures.

4. Heat Transfer Coefficient Comparison

Figure 49 shows that the TRACE heat transfer coefficient was much lower than in experimental data, suggesting that the disparity between the fluid and wall temperatures was due to underpredicted heat transfer. Additionally, the heat transfer coefficient was nearly constant over the entire transient.

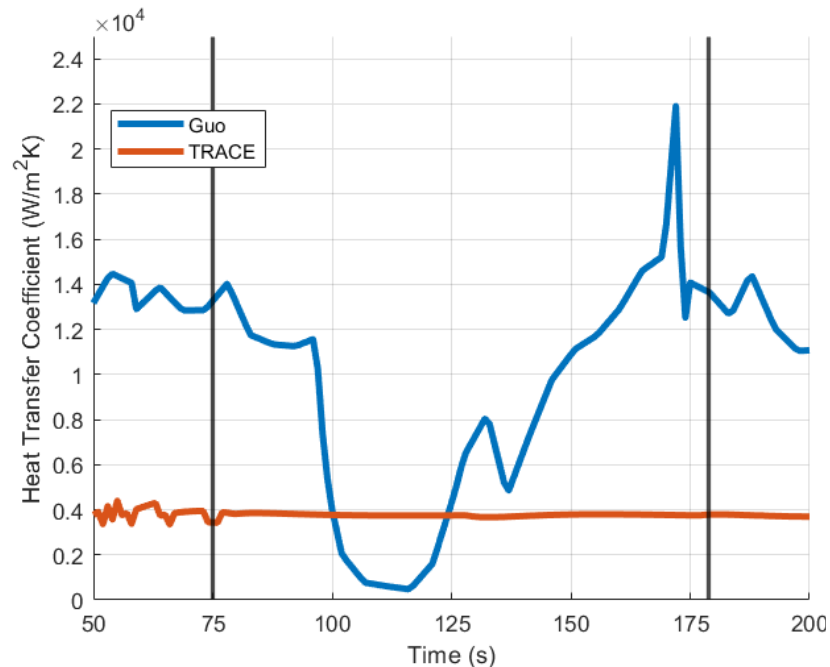


Figure 49. TRACE initial model and Guo et al. heat transfer coefficient comparison

The maximum timestep size was reduced from 1E-2 to 1E-4 seconds, and the allotted problem time was doubled to eliminate unsteady behavior before the transient. This change did not result in a significant change in the heat transfer coefficient. Figure 50 shows the heat transfer coefficient with the wall roughness removed from the TRACE model. In a physical system, the heat transfer coefficient would be lower for a smoother pipe wall due to the reduced turbulence and mixing near the wall. While the heat transfer coefficient value did not change significantly in this case, the trends seen before and after the transient resembled those in the experimental data. The slight downward trend before the transient and the spike towards the end of the transient were seen in both datasets. However, this was likely coincidental. Due to a small temperature difference between the pipe wall and the vapor, the heat transfer coefficient was intermittently undefined prior to the transient.

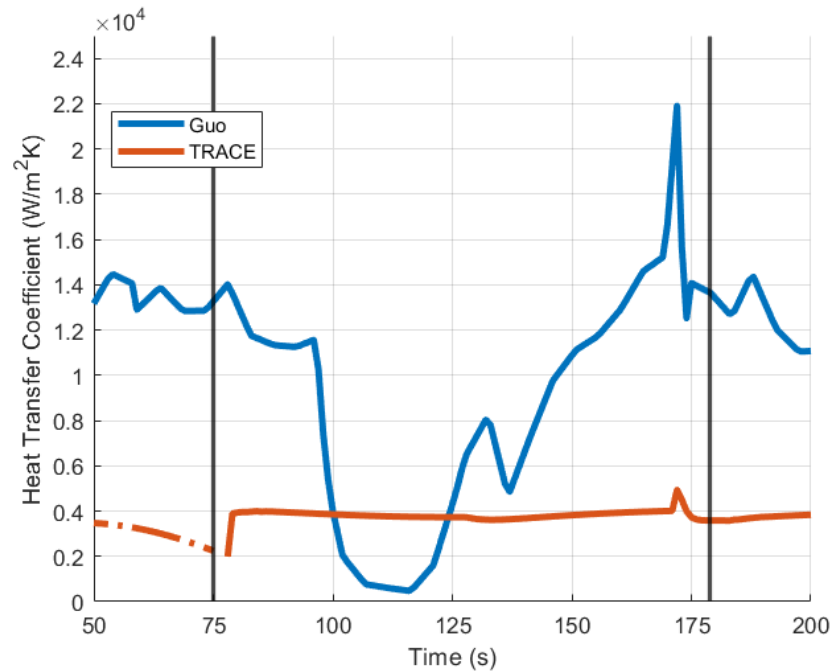


Figure 50. TRACE smooth pipe model and Guo et al. heat transfer coefficient comparison

As shown in Figure 51, the fill flow quality change caused the heat transfer coefficient to be undefined until just after the beginning of the transient. The trends in Figure 50 were not seen in this case, where the TRACE heat transfer coefficient was nearly constant for the defined region.

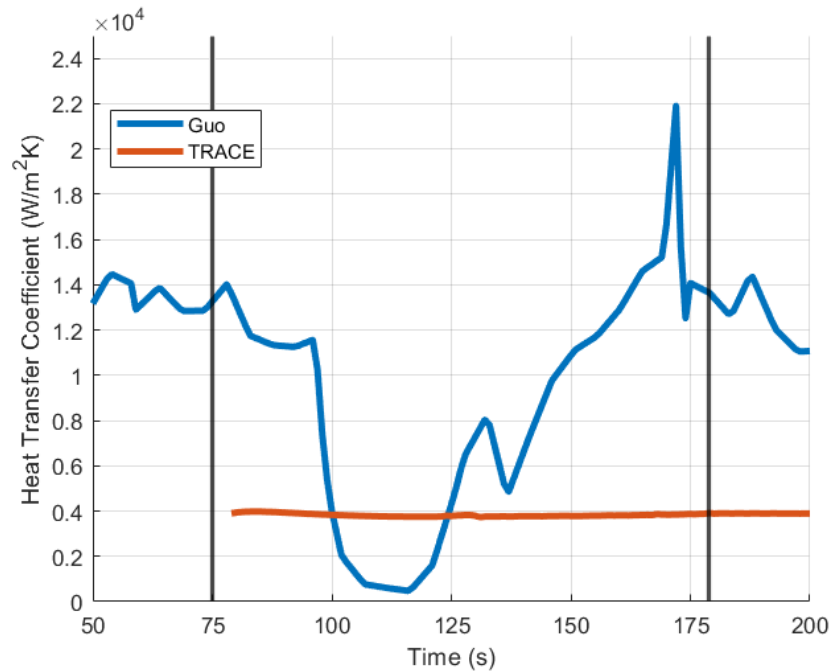


Figure 51. TRACE smooth 99.96% flow quality pipe model and Guo et al. heat transfer coefficient comparison

5. Mass Flux Comparison

The mass flux, shown in Figure 52, was constant over the course of the transient for each of the three different TRACE runs. The Guo et al. experimental setup relied on many components, such as the surge tanks, preheaters, and pumps, to control the steady-state and transient flow characteristics. The flow parameters in TRACE were implemented instantaneously and ideally, resulting in a non-physical, constant mass flux throughout the transient.

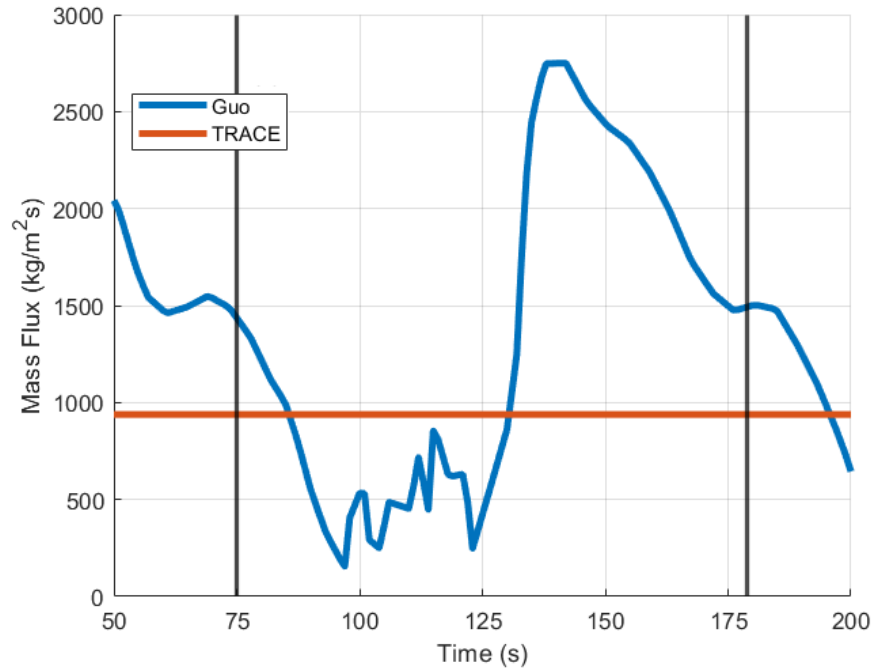


Figure 52. TRACE and Guo et al. mass flux comparison

6. Vapor Mass Fraction Comparison

As expected from previous simulations, the TRACE solver could not model minute changes for high vapor mass fraction flows. Figure 53 shows that the cell vapor mass fraction was static for the test section length and the pressure transient in the initial TRACE model.

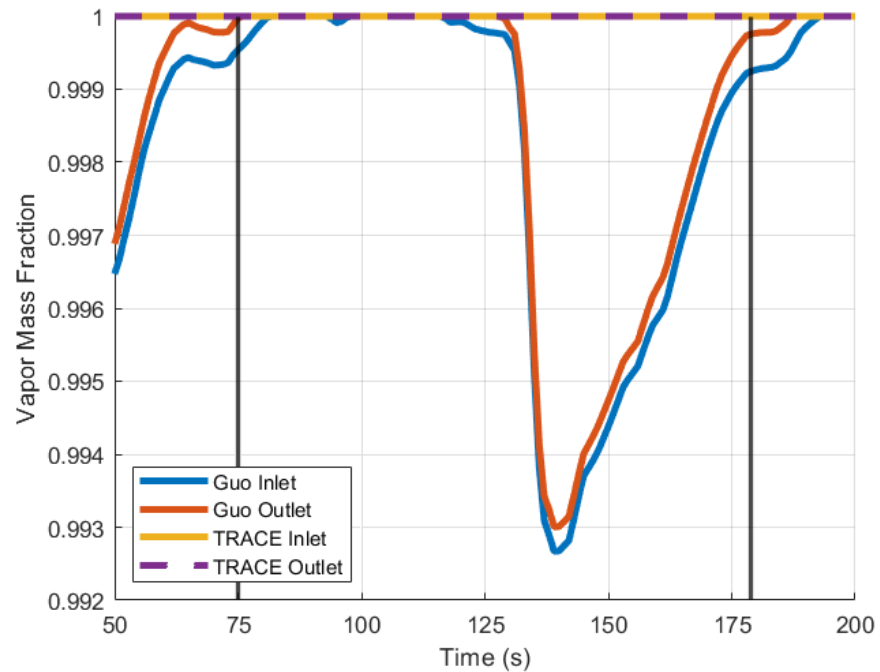


Figure 53. TRACE initial model and Guo et al. vapor mass fraction comparison

As shown in Figure 54, removing wall roughness from the TRACE model resulted in slight unsteadiness in the vapor mass fraction at the test section outlet, while the inlet vapor mass fraction was unaffected.

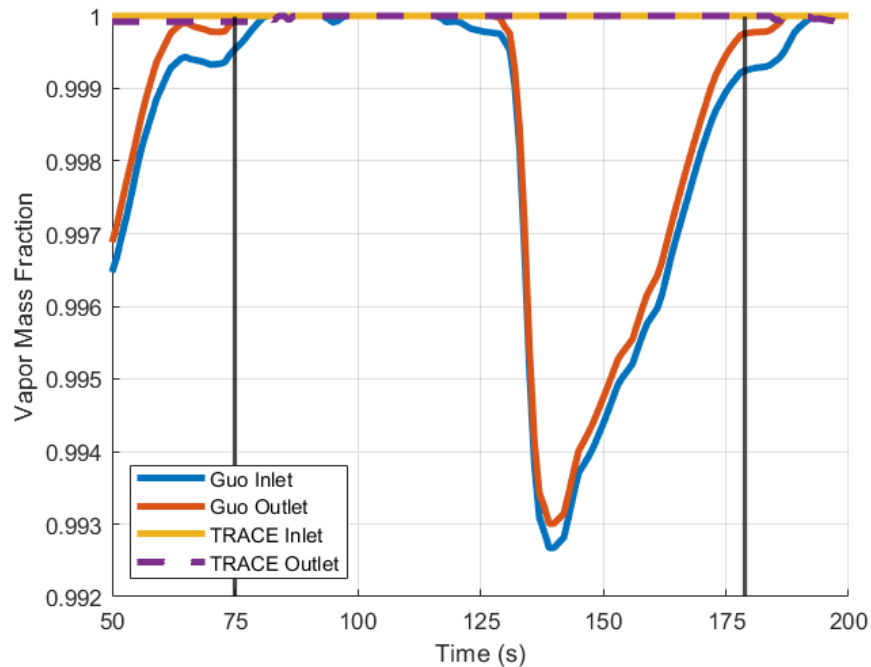


Figure 54. TRACE smooth pipe model and Guo et al. vapor mass fraction comparison

The change in fill flow quality from 100% to 99.96% caused the vapor mass fraction at the outlet to change over the transient, as illustrated in Figure 55. While the vapor mass fraction was inversely related to pressure in the experimental data, the TRACE vapor mass fraction rose and fell with pressure transient. This change in fill flow quality also resulted in an unsteady vapor mass fraction calculation after the beginning of the pressure transient.

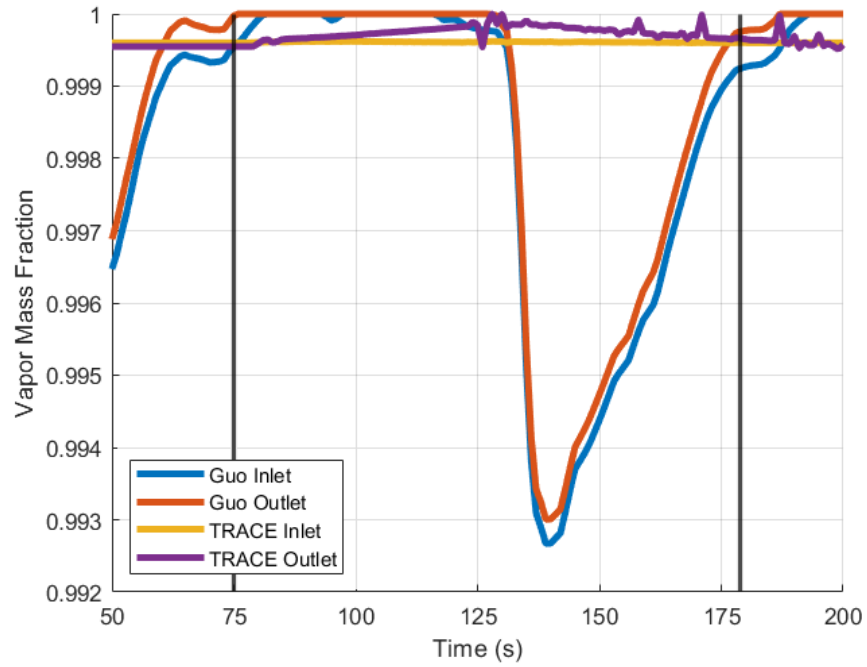


Figure 55. TRACE smooth 99.96% flow quality pipe model and Guo et al. vapor mass fraction comparison

VIII. CONCLUSIONS AND RECOMMENDATIONS

The U.S. Navy uses pressurized water nuclear reactors to power its aircraft carriers and submarines. Design optimization of the secondary loop is vital to improve efficiency and steam quality and minimize losses under transient conditions. CFD codes can predict steam behavior and allow for iteration of future designs. The collection of liquid water in these systems causes efficiency losses and damage to pipe and turbine components. Transient flows, which result from variations in reactor operation, cause changes in pressure and flow quality, increase the moisture in the piping systems. This study aimed to model high-quality transient two-phase steam flows in a piping system. Two transient parameters were of interest: pressure and inlet flow quality. TRACE is a powerful tool for modeling the performance of large-scale thermo-fluid systems, such as pressurized water reactors, under various conditions. However, as the code developers have stated, “TRACE makes no attempt to capture, in detail, the fluid dynamics in a pipe” [15]. In light of these limitations, this work also sought to interrogate the limits of TRACE’s ability to produce reasonable results for the flows relevant to this effort.

An examination of TRACE’s steady-state modeling for a range of inlet flow qualities suggested that the code could not reasonably simulate moisture distribution within the piping system for inlet qualities above 98%. As with previous work analyzing TRACE performance, this study observed that the TRACE solver tended to overpredict the pressure differential across the piping system. For transient pressure cases, these observations were considered, and the inlet flow quality was tested at 98% rather than 99%. These simulations yielded reasonable results. However, the boundary conditions at the fill and break in the TRACE model resulted in forced, non-physical temperatures and unstable heat transfer at the inlet and outlet of the pipe during transient portions of the simulation. Attempts to model transient cases for inlet flow quality were unsuccessful. Using the available options in the TRACE input model, transient fill flow quality could not be modeled at a constant input pressure without overconstraining the problem or forcing the flow velocity to zero.

The case study comparison of TRACE to open literature data reinforced the observation that TRACE is limited in modeling high-quality flows. Slight changes in model

input parameters resulted in large instabilities in the solution and affected parameters that should not have changed. The TRACE estimations of pressure and temperature for transient cases in this study were reasonable and the computational resources required were minimal compared to codes such as Ansys CFX and Fluent. Because TRACE was designed to model heat transfer between the liquid coolant and fuel rods within the reactor compartment, the heat transfer coefficient correlations beyond the critical heat point were under developed. As a result, TRACE predicted a constant, underpredicted heat transfer coefficient at the pipe wall for the comparison case with experimental data. The lack of wall heat transfer relative to the experimental data led to a high fluid temperature, lower wall temperature, and a lack of change in the vapor quality during the transient.

While the TRACE solver produced reasonable steady-state results and had very low computational requirements for all cases, the results for high-quality moisture distribution and wall heat transfer were poor. TRACE displayed a limited ability to model heat transfer coefficients and moisture distribution for high-quality, two-phase transient steam flows within the piping system. Future work with TRACE could be improved by refining heat transfer correlations for the pipe component and high-quality flow regimes.

APPENDIX A. MATLAB SCRIPT FOR VAPOR MASS FRACTION TO VOLUME FRACTION CONVERSION

```

%% Conversion From Vapor Mass Fraction to Gas Volume Fraction

% ***** CHANGE FROM '2022' TO 'garth' for personal/lab computer*****

% Tidy up!
close all
clear
clc
addpath ['C:\Users\2022\OneDrive - Naval Postgraduate School\Stonhill...
        ' Thesis\Open Literature Data\Guo Two Phase Steam\XSteam_Matlab_v2.6']

%% VF to X

% Define Variables
VF = linspace(0.1,1,1000000); % volume fraction vector
r  = 0.25/2;                 % pipe cell radius - m
L  = 0.25;                   % pipe cell length - m
A  = pi*r^2;                 % pipe cell cross section area - m^2
P  = 2.639*10;               % fluid pressure - bar
V_tot = L*A;                 % total cell volume - m^3
T  = 505.89;                 % Fluid temperature - K

%% Calculate vapor mass fraction from vapor volume fraction array

X = NaN(1,1000000);         % preallocate memory
for i = 1:1000000
    V_v = VF(i)*V_tot;       % vapor volume - m^3
    V_l = (1-VF(i))*V_tot;   % liquid volume - m^3
    rho_v = 1/(XSteam('vV_p',P)); % vapor density - kg/m^3
    rho_l = 1/(XSteam('vL_p',P)); % liquid density - kg/m^3
    m_v = V_v*rho_v;         % vapor mass - kg
    m_l = V_l*rho_l;         % liquid mass - kg
    m_tot = m_l + m_v;       % total mass - kg
    X(i) = m_v/m_tot;        % vapor mass fraction

% Conditional statements to isolate relevant vapor mass fractions
if X(i) > 0.75 && X(i) < 0.751
    VFforX75 = VF(i);
end

if X(i) > 0.70 && X(i) < 0.701
    VFforX70 = VF(i);
end

if X(i) > 0.80 && X(i) < 0.801
    VFforX80 = VF(i);
end

```

```

if X(i) > 0.83 && X(i) < 0.831
    VFforX83 = VF(i);
end

if X(i) > 0.84 && X(i) < 0.841
    VFforX84 = VF(i);
end

if X(i) > 0.85 && X(i) < 0.851
    VFforX85 = VF(i);
end

if X(i) > 0.90 && X(i) < 0.901
    VFforX90 = VF(i);
end

if X(i) > 0.95 && X(i) < 0.951
    VFforX95 = VF(i);
end

if X(i) > 0.96 && X(i) < 0.961
    VFforX96 = VF(i);
end

if X(i) > 0.97 && X(i) < 0.971
    VFforX97 = VF(i);
end

if X(i) > 0.98 && X(i) < 0.981
    VFforX98 = VF(i);
end

if X(i) > 0.99 && X(i) < 0.991
    VFforX99 = VF(i);
end

if X(i) > 0.9996 && X(i) < 0.99961
    VFforX9996 = VF(i);
end

end

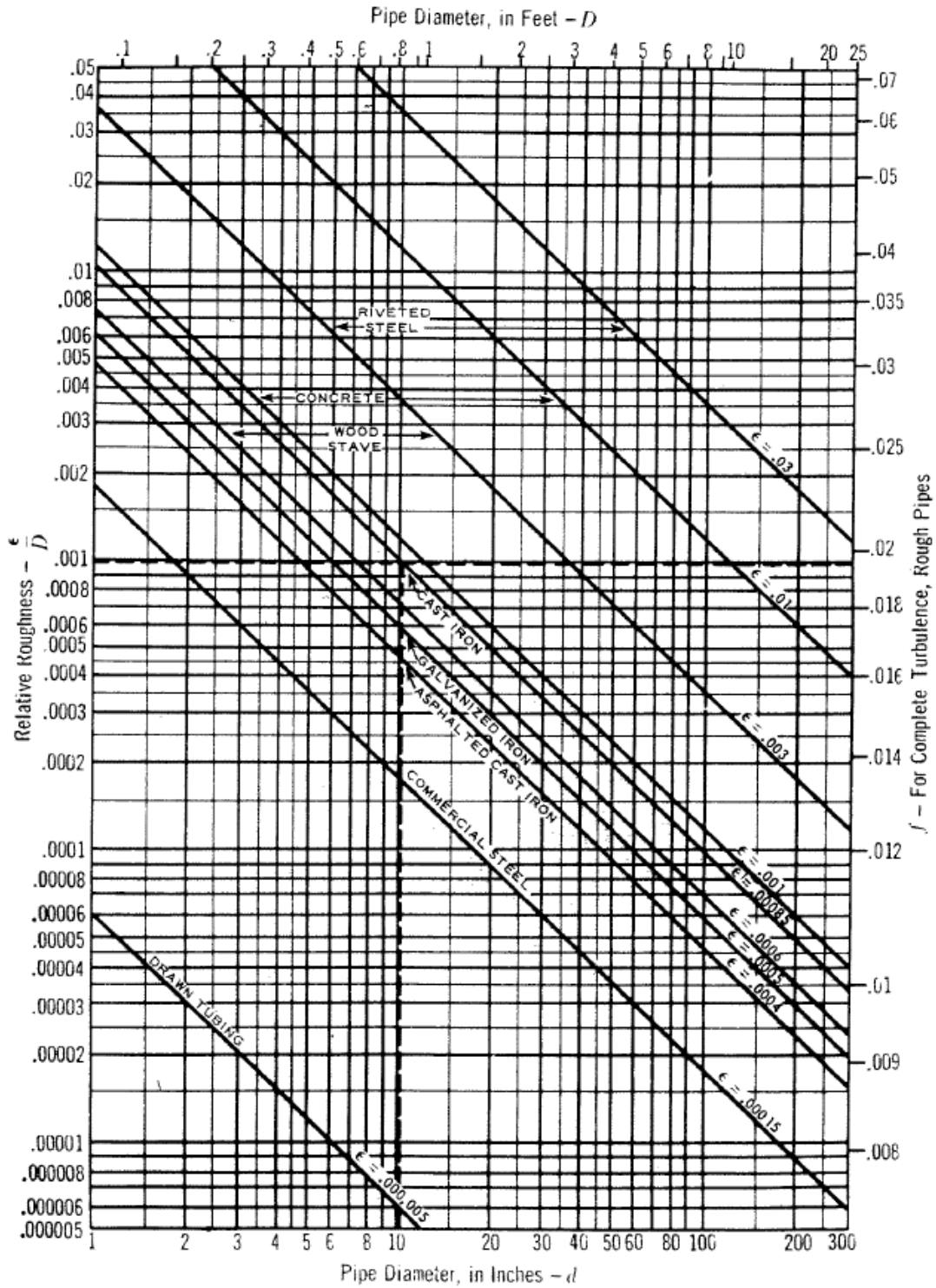
```

APPENDIX B. STEADY-STATE PHASE MASS DISTRIBUTION FOR VARYING INLET QUALITY

Inlet Flow Quality (%)	Pipe Liquid Mass (kg)	Pipe Vapor Mass (kg)	Pipe Fluid Mass (kg)
85	3.0124	14.6541	17.6665
90	1.8564	14.6542	16.5106
95	0.8315	14.6513	15.4828
96	0.6432	14.6502	15.2934
97	0.4589	14.6491	15.108
98	0.2878	14.6396	14.9274
99	0.1575	14.6238	14.7813

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. MOODY DIAGRAM FOR ROUGHNESS IN FULLY TURBULENT PIPE FLOW



THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. PIPE BEND RESISTANCE COEFFICIENT DOCUMENTATION

the formation of the so-called vortex pair which, superimposed on the main stream parallel to the channel axis, gives the streamlines a helical shape (Figure 6-3).

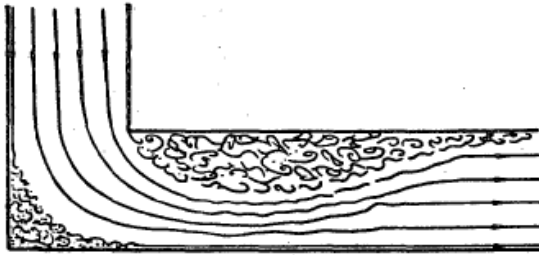


FIGURE 6-2. Stream pattern in a 90° elbow.

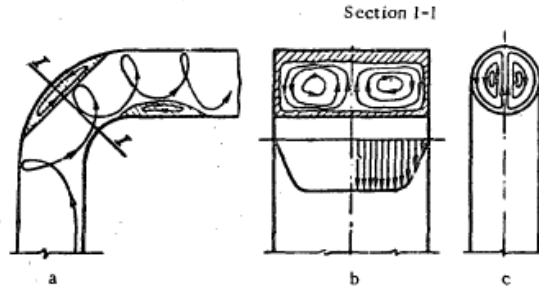


FIGURE 6-3. Vortex pair in an elbow:

a — longitudinal section; b — cross section of rectangular conduit;
c — cross section of a circular pipe.

4. The pressure losses in curved pipes are mainly due to the formation of eddies at the inner wall. This eddy formation also determines the pattern of velocity distribution beyond the bend.

The magnitude of the resistance coefficient in curved pipes varies as a function of the factors determining the turbulent intensity, the Reynolds number $Re = \frac{w_0 D_h}{\nu}$, the relative roughness of the walls $\bar{\Delta} = \frac{\Delta}{D_h}$, the inlet conditions, etc. It is also a function of the pipe shape, the angle of bend δ° , the relative radius of curvature $\frac{R_0}{D_h}$, the side ratio of the cross section $\frac{a_2}{b_0}$, the ratio of the inlet to exit areas $\frac{F_3}{F_0}$, etc.

5. The total resistance coefficient of bends and elbows is determined as the sum of the coefficient of local resistance of the bend ζ_l and the friction coefficient ζ_{fr} :

$$\zeta = \frac{\Delta H}{\frac{\gamma w_0^2}{2g}} = \zeta_l + \zeta_{fr}, \quad (6-1)$$

where ζ_l is determined from the data given in this section; $\zeta_{fr} = \lambda \frac{l}{D_h}$ is calculated as ζ for straight stretches, with λ determined from diagrams 2-1 to 2-5 as a function of Re and the relative roughness $\bar{\Delta} = \frac{\Delta}{D_h}$; l = length of the bend or elbow, measured along the axis, so that:

$$\frac{l}{D_h} = \pi \frac{\delta^\circ}{180^\circ} \cdot \frac{R_0}{D_h} = 0.0175 \frac{R_0}{D_h} \delta^\circ. \quad (6-2)$$

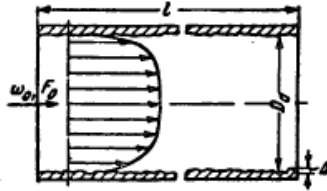
Thus,

$$\zeta_{fr} = 0.0175 \lambda \frac{R_0}{D_h} \delta^\circ. \quad (6-3)$$

Conduit with uniform wall roughness.
Friction coefficient at $Re > 2000$

Section II
Diagram 2-3

$$D_h = \frac{4F_0}{\Pi_0}; \Pi_0 = \text{perimeter}$$



1. Circular and rectangular sections ($\frac{a_0}{b_0} \approx 0.5 + 2.0$):

$$\lambda = \frac{\Delta H}{\frac{\gamma w_0^2 l}{2g D_h}} = \frac{1}{[a_1 + b_1 \lg(Re \sqrt{\lambda}) + c_1 \lg \Delta]^2}$$

is determined from curve a and Table 2-2 (p. 69); the values of a_1 , b_1 and c_1 are given below:

$\Delta Re \sqrt{\lambda}$	a_1	b_1	c_1
3.6-10	-0.800	2.000	0
10-20	0.068	1.130	-0.870
20-40	1.538	0.000	-2.000
40-191.2	2.471	-0.588	-2.588
>191.2	1.138	0	-2.000

2. Annular section:

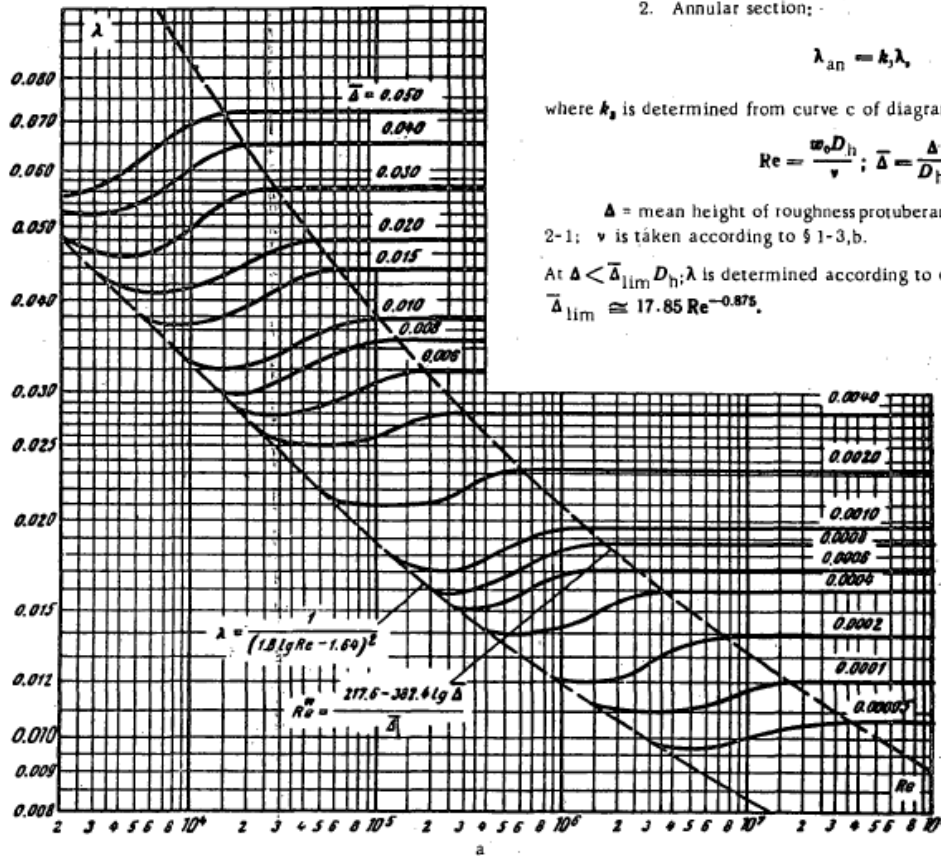
$$\lambda_{an} = k_s \lambda_s$$

where k_s is determined from curve c of diagram 2-1;

$$Re = \frac{w_0 D_h}{\nu}; \Delta = \frac{\Delta}{D_h};$$

Δ = mean height of roughness protuberances according to Table 2-1; ν is taken according to § 1-3, b.

At $\Delta < \Delta_{lim} D_h$; λ is determined according to diagram 2-2, where $\Delta_{lim} \approx 17.85 Re^{-0.875}$.



Conduit with uniform wall roughness. Friction coefficient
at $Re > 2000$ (continuation of diagram 2-3)

Section II
Table 2-2

Values of λ

Re $\lambda = \frac{f}{D_{11}}$	$2 \cdot 10^3$	$4 \cdot 10^3$	$6 \cdot 10^3$	10^4	$2 \cdot 10^4$	$4 \cdot 10^4$	$6 \cdot 10^4$	10^5	$2 \cdot 10^5$
0.05	0.056	0.060	0.063	0.069	0.072	0.072	0.072	0.072	0.072
0.04	0.053	0.053	0.055	0.060	0.065	0.065	0.065	0.065	0.065
0.03	0.048	0.046	0.046	0.050	0.056	0.057	0.057	0.057	0.057
0.02	0.048	0.042	0.041	0.042	0.044	0.048	0.049	0.049	0.049
0.015	0.048	0.042	0.038	0.037	0.039	0.042	0.044	0.044	0.044
0.010	0.048	0.042	0.038	0.033	0.032	0.035	0.036	0.038	0.038
0.008	0.048	0.042	0.038	0.033	0.030	0.032	0.033	0.035	0.035
0.006	0.048	0.042	0.038	0.033	0.028	0.028	0.029	0.030	0.032
0.004	0.048	0.042	0.038	0.033	0.027	0.025	0.025	0.026	0.028
0.002	0.048	0.042	0.038	0.033	0.027	0.023	0.021	0.021	0.021
0.001	0.048	0.042	0.038	0.033	0.027	0.023	0.021	0.018	0.017
0.0008	0.048	0.042	0.038	0.033	0.027	0.023	0.021	0.018	0.016
0.0006	0.048	0.042	0.038	0.033	0.027	0.023	0.021	0.018	0.016
0.0004	0.048	0.042	0.038	0.033	0.027	0.023	0.021	0.018	0.016
0.0002	0.048	0.042	0.038	0.033	0.027	0.023	0.021	0.018	0.016
0.0001	0.048	0.042	0.038	0.033	0.027	0.023	0.021	0.018	0.016
0.00005	0.048	0.042	0.038	0.033	0.027	0.023	0.021	0.018	0.016

Values of λ

Re $\lambda = \frac{f}{D_{11}}$	$4 \cdot 10^5$	$6 \cdot 10^5$	10^6	$2 \cdot 10^6$	$4 \cdot 10^6$	$6 \cdot 10^6$	10^7	$2 \cdot 10^7$	$> 10^8$
0.05	0.072	0.072	0.072	0.072	0.072	0.072	0.072	0.072	0.072
0.04	0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065
0.03	0.057	0.057	0.057	0.057	0.057	0.057	0.057	0.057	0.057
0.02	0.049	0.049	0.049	0.049	0.049	0.049	0.049	0.049	0.049
0.015	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044
0.010	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038
0.008	0.035	0.035	0.035	0.035	0.035	0.035	0.035	0.035	0.035
0.006	0.032	0.032	0.032	0.032	0.032	0.032	0.032	0.032	0.032
0.004	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028
0.002	0.022	0.023	0.023	0.023	0.023	0.023	0.023	0.023	0.023
0.001	0.018	0.018	0.020	0.020	0.020	0.020	0.020	0.020	0.020
0.0008	0.016	0.017	0.018	0.019	0.019	0.019	0.019	0.019	0.019
0.0006	0.015	0.016	0.017	0.017	0.017	0.017	0.017	0.017	0.017
0.0004	0.014	0.014	0.014	0.015	0.016	0.016	0.016	0.016	0.016
0.0002	0.014	0.013	0.012	0.012	0.013	0.014	0.014	0.014	0.014
0.0001	0.014	0.013	0.012	0.011	0.011	0.011	0.012	0.012	0.012
0.00005	0.014	0.013	0.012	0.011	0.010	0.010	0.010	0.010	0.011

16. The coefficient of local resistance of smooth bends is calculated by the following formula, proposed by Abramovich /6-1/* :

$$\zeta_1 = \frac{\Delta H}{\frac{v_0^2}{2g}} = A_1 B_1 C_1, \quad (6-4)$$

where A_1 is a coefficient allowing for the influence of the bend angle δ° ; B_1 is a coefficient allowing for the influence of the relative radius of curvature of the bend $\frac{R_0}{D_h}$; C_1 is a coefficient allowing for the influence of the side ratio of the bend cross section $\frac{a_c}{b_0}$.

The value of A_1 is found from the following table, established by Nekrasov /6-11/ :

$$\left. \begin{array}{l} \text{at } \delta = 90^\circ \quad A_1 = 1.0; \\ \text{at } \delta < 70^\circ \quad A_1 = 0.9 \sin \delta; \\ \text{at } \delta > 100^\circ \quad A_1 = 0.7 + 0.35 \frac{\delta}{90^\circ} \end{array} \right\} \quad (6-5)$$

or from graph a of diagrams 6-1 or 6-2.

The value of B_1 can be calculated by the following approximate formulas:

$$\left. \begin{array}{l} \text{at } \frac{R_0}{D_h} < 1.0 \quad B_1 = \frac{0.21}{\left(\frac{R_0}{D_h}\right)^{2.5}}; \\ \text{at } \frac{R_0}{D_h} > 1.0 \quad B_1 = \frac{0.21}{\sqrt{\frac{R_0}{D_h}}} \end{array} \right\} \quad (6-6)$$

or from graph b of diagrams 6-1 or 6-2.

The value of C_1 is determined from graph c of diagrams 6-1 or 6-2.

APPENDIX E. MATLAB FUNCTIONS FOR IMPORTING TRACE DATA WITHOUT USE OF INTERFACE

```
function [P,P_drop_rate,n_timestep] = TRACE_Pressure_Function(Pa,Pb,Pc)

% This function parses pressure data from TRACE and returns a pressure
% vector and a vector for the pressure drop rate

Pa = importdata(Pa);
Pb = importdata(Pb);
Pc = importdata(Pc);
t = Pa.data(:,1);
n_timestep = length(t);
Pa = Pa.data(:,2:length(Pa.data(1,:)));
Pb = Pb.data(:,2:length(Pb.data(1,:)));
Pc = Pc.data(:,2:length(Pc.data(1,:)));
P = cat(2,Pa,Pb,Pc);
diffP = diff(P');
P_drop_rate = diffP';
end

function [templiqliq,tempgas,tempwalli,tempwallo,tempsat,qwalli,qwallo] =...
TRACE_Wall_Function(templiqla,templiqlb,templiqlc,tempgasa,tempgasb,...
tempgasc,tempwallia,tempwallib,tempwallic,tempwalloa,tempwallob,...
tempwalloc,tempsata,tempsatb,tempsatc,qwallia,qwallib,qwallic,...
qwalloa,qwallob,qwalloc)

% This function parses pipe wall data from TRACE and returns
% arrays for inner and outer wall temperature along the pipe as well as
% heat flux on the inner and outer surfaces of the wall along the pipe

templiqla = importdata(templiqla);
templiqlb = importdata(templiqlb);
templiqlc = importdata(templiqlc);
templiqla = templiqla.data(:,2:length(templiqla.data(1,:)));
templiqlb = templiqlb.data(:,2:length(templiqlb.data(1,:)));
templiqlc = templiqlc.data(:,2:length(templiqlc.data(1,:)));
templiqliq = cat(2,templiqla,templiqlb,templiqlc);

tempgasa = importdata(tempgasa);
tempgasb = importdata(tempgasb);
tempgasc = importdata(tempgasc);
tempgasa = tempgasa.data(:,2:length(tempgasa.data(1,:)));
tempgasb = tempgasb.data(:,2:length(tempgasb.data(1,:)));
tempgasc = tempgasc.data(:,2:length(tempgasc.data(1,:)));
tempgas = cat(2,tempgasa,tempgasb,tempgasc);

tempwallia = importdata(tempwallia);
tempwallib = importdata(tempwallib);
```

```

tempwallic = importdata(tempwallic);
tempwallia = tempwallia.data(:,2:length(tempwallia.data(1,:)));
tempwallib = tempwallib.data(:,2:length(tempwallib.data(1,:)));
tempwallic = tempwallic.data(:,2:length(tempwallic.data(1,:)));
tempwalli = cat(2,tempwallia,tempwallib,tempwallic);

tempwalloa = importdata(tempwalloa);
tempwallob = importdata(tempwallob);
tempwalloc = importdata(tempwalloc);
tempwalloa = tempwalloa.data(:,2:length(tempwalloa.data(1,:)));
tempwallob = tempwallob.data(:,2:length(tempwallob.data(1,:)));
tempwalloc = tempwalloc.data(:,2:length(tempwalloc.data(1,:)));
tempwallo = cat(2,tempwalloa,tempwallob,tempwalloc);

tempsata = importdata(tempsata);
tempsatb = importdata(tempsatb);
tempsatc = importdata(tempsatc);
tempsata = tempsata.data(:,2:length(tempsata.data(1,:)));
tempsatb = tempsatb.data(:,2:length(tempsatb.data(1,:)));
tempsatc = tempsatc.data(:,2:length(tempsatc.data(1,:)));
tempsat = cat(2,tempsata,tempsatb,tempsatc);

qwallia = importdata(qwallia);
qwallib = importdata(qwallib);
qwallic = importdata(qwallic);
qwallia = qwallia.data(:,2:length(qwallia.data(1,:)));
qwallib = qwallib.data(:,2:length(qwallib.data(1,:)));
qwallic = qwallic.data(:,2:length(qwallic.data(1,:)));
qwalli = cat(2,qwallia,qwallib,qwallic);

qwalloa = importdata(qwalloa);
qwallob = importdata(qwallob);
qwalloc = importdata(qwalloc);
qwalloa = qwalloa.data(:,2:length(qwalloa.data(1,:)));
qwallob = qwallob.data(:,2:length(qwallob.data(1,:)));
qwalloc = qwalloc.data(:,2:length(qwalloc.data(1,:)));
qwallo = cat(2,qwalloa,qwallob,qwalloc);
end

function [mdotgas,mdottot,FQ,FQ_drop_rate,n_timestep] =...
TRACE_Flow_Quality_Function(mdotgasa,mdotgasb,mdotgasc,...
mdottota,mdottotb,mdottotc)

% This function parses mass flow rate data from TRACE and returns a flow
% quality vector and a vector for the flow quality change rate

mdotgasa = importdata(mdotgasa);
mdotgasb = importdata(mdotgasb);
mdotgasc = importdata(mdotgasc);
t = mdotgasa.data(:,1);
n_timestep = length(t);

```

```

mdotgasa = mdotgasa.data(:,2:size(mdotgasa.data,2));
mdotgasb = mdotgasb.data(:,2:size(mdotgasb.data,2));
mdotgasc = mdotgasc.data(:,2:size(mdotgasc.data,2));
mdotgas = cat(2,mdotgasa,mdotgasb,mdotgasc);

mdottota = importdata(mdottota);
mdottotb = importdata(mdottotb);
mdottotc = importdata(mdottotc);
mdottota = mdottota.data(:,2:size(mdottota.data,2));
mdottotb = mdottotb.data(:,2:size(mdottotb.data,2));
mdottotc = mdottotc.data(:,2:size(mdottotc.data,2));
mdottot = cat(2,mdottota,mdottotb,mdottotc);

FQ = mdotgas./mdottot;
diffFQ = diff(FQ');
FQ_drop_rate = diffFQ';
end

```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F. MATLAB SCRIPT FOR STEADY-STATE TRACE DATA PROCESSING

```
%% TRACE Steady-State Bent Pipe Data Processor

% Tidy Up!
close all
clear
clc

% ***** CHANGE THE FOLDER OF THE XTV FILE YOU WISH TO ANALYZE*****
% ***** CHANGE FROM '2022' TO 'garth' for personal/lab computer*****

% Tidy Up!
close all
clear
clc

% Add TRACE-MATLAB Interface to path
javaaddpath('C:\Users\2022\snap\lib\cafean.jar')
javaaddpath('C:\Users\2022\snap\lib\matlab_acs.jar')
javaaddpath('C:\Users\2022\snap\plugins\extdata.jar')
javaaddpath('C:\Users\2022\snap\plugins\TRACE.jar')
import com.cafean.matlab.*
TRACEINT = MatlabTraceIntf; % handle for MATLAB-TRACE interface

%% Open XTV file for 99% inlet quality and generate file ID

TRACEINT.openFile(['C:\Users\2022\OneDrive - Naval Postgraduate' ...
' School\Stonhill Thesis\TRACE Data\TRACE Runs\04JANnum1\Base_Job\'...
'Base_Job.xtv']));

% Change from British units (default) to SI units.
TRACEINT.setUseSIunits( true );

% Number of timesteps from TRACE XTV file
NumTimeSteps99 = TRACEINT.getNumTimeSteps(0);

% Number of Cells in simulation
NumCells99 = 92;

% Preallocate memory
[mdotgas99,mdottot99,rho_v99,rho_l99,P99,templiq99,...
tempgas99,tempsat99,VF99] = ...
deal(NaN(NumTimeSteps99,NumCells99*2));

% Load data

for i = 1:NumCells99

if i < 10
```

```

j = '0';
else
j = '';
end

k = num2str(i);
m = i-1;

mdotgas99(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('rvmf-10A',j,k)});
mdottot99(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('rmvm-10A',j,k)});
rho_v99(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('rovn-10A',j,k)});
rho_l99(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('roln-10A',j,k)});
P99(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('pn-10A',j,k)});
templiq99(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tln-10A',j,k)});
tempgas99(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tvn-10A',j,k)});
tempsat99(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tssn-10A',j,k)});
VF99(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('alpn-10A',j,k)});

end

% delete timestep columns
mdotgas99(:,1:2:end) = [];
mdottot99(:,1:2:end) = [];
rho_v99(:,1:2:end) = [];
rho_l99(:,1:2:end) = [];
P99(:,1:2:end) = [];
tempgas99(:,1:2:end) = [];
templiq99(:,1:2:end) = [];
tempsat99(:,1:2:end) = [];
VF99(:,1:2:end) = [];

% Calculated Values

% all change rates are with respect to pipe length, not time
Cells99 = 1:NumCells99;
PipeLength99 = Cells99*0.25; % Length of Pipe- m
r99 = 0.125; % pipe inner radius -m
l99 = 0.25; % cell length - m
cell_vol99 = pi*(r99^2)*l99; % cell volume - m^3
FQ99 = mdotgas99./mdottot99; % cell flow quality
mdotliq99 = mdottot99-mdotgas99; % liquid mass flow rate - kg/s
diffFQ99 = diff(FQ99');
FQ_drop_rate99 = diffFQ99'; % cell flow quality change rate
V_v99 = VF99.*cell_vol99; % cell vapor volume - m^3
V_l99 = (1.-VF99).*cell_vol99; % cell liquid volume - m^3
m_v99 = V_v99.*rho_v99; % cell vapor mass - kg
m_l99 = V_l99.*rho_l99; % cell liquid mass - kg
massfrac99 = m_v99./(m_v99+m_l99); % cell vapor mass fracion
diffmassfrac99 = diff(massfrac99');
massfrac_drop_rate99 = diffmassfrac99'; % vapor mass fraction change rate
diffP99 = diff(P99');
P_drop_rate99 = diffP99'; % cell pressure change rate - MPa/m
m_v_pipe99 = sum(m_v99,2); % total mass of vapor in pipe - kg
m_l_pipe99 = sum(m_l99,2); % total mass of vapor in pipe - kg

```

```

%% Open XTV file for 98% inlet quality and generate file ID

TRACEINT.openFile(['C:\Users\2022\OneDrive - Naval Postgraduate' ...
' School\Stonhill Thesis\TRACE Data\TRACE Runs\04JANnum2\Base_Job\'...
'Base_Job.xtv']);

%% Change from British units (default) to SI units.
TRACEINT.setUseSIunits( true );

% Number of timesteps from TRACE XTV file
NumTimeSteps98 = TRACEINT.getNumTimeSteps(1);

% Number of Cells in simulation
NumCells98 = 92;

% Preallocate memory
[mdotgas98,mdottot98,rho_v98,rho_l98,P98,templiq98,...
tempgas98,tempsat98,VF98] = ...
deal(NaN(NumTimeSteps98,NumCells98*2));

% Load data

for i = 1:NumCells98

if i < 10
j = '0';
else
j = '';
end

k = num2str(i);
m = i-1;

mdotgas98(:,i+m:i+1+m) = TRACEINT.getData(1,{strcat('rvmf-10A',j,k)});
mdottot98(:,i+m:i+1+m) = TRACEINT.getData(1,{strcat('rmvm-10A',j,k)});
rho_v98(:,i+m:i+1+m) = TRACEINT.getData(1,{strcat('rovn-10A',j,k)});
rho_l98(:,i+m:i+1+m) = TRACEINT.getData(1,{strcat('roln-10A',j,k)});
P98(:,i+m:i+1+m) = TRACEINT.getData(1,{strcat('pn-10A',j,k)});
templiq98(:,i+m:i+1+m) = TRACEINT.getData(1,{strcat('tln-10A',j,k)});
tempgas98(:,i+m:i+1+m) = TRACEINT.getData(1,{strcat('tvn-10A',j,k)});
tempsat98(:,i+m:i+1+m) = TRACEINT.getData(1,{strcat('tssn-10A',j,k)});
VF98(:,i+m:i+1+m) = TRACEINT.getData(1,{strcat('alpn-10A',j,k)});

end

% delete timestep columns
mdotgas98(:,1:2:end) = [];
mdottot98(:,1:2:end) = [];
rho_v98(:,1:2:end) = [];
rho_l98(:,1:2:end) = [];
P98(:,1:2:end) = [];
tempgas98(:,1:2:end) = [];

```

```

templiq98(:,1:2:end) = [];
tempsat98(:,1:2:end) = [];
VF98(:,1:2:end) = [];

% Calculated Values

% all change rates are with respect to pipe length, not time
Cells98 = 1:NumCells98;
PipeLength98 = Cells98*0.25; % Length of Pipe- m
r98 = 0.125; % pipe inner radius -m
l98 = 0.25; % cell length - m
cell_vol98 = pi*(r98^2)*l98; % cell volume - m^3
FQ98 = mdotgas98./mdottot98; % cell flow quality
mdotliq98 = mdottot98-mdotgas98; % liquid mass flow rate - kg/s
diffFQ98 = diff(FQ98');
FQ_drop_rate98 = diffFQ98'; % cell flow quality change rate
V_v98 = VF98.*cell_vol98; % cell vapor volume - m^3
V_l98 = (1.-VF98).*cell_vol98; % cell liquid volume - m^3
m_v98 = V_v98.*rho_v98; % cell vapor mass - kg
m_l98 = V_l98.*rho_l98; % cell liquid mass - kg
massfrac98 = m_v98./(m_v98+m_l98); % cell vapor mass fracion
diffmassfrac98 = diff(massfrac98');
massfrac_drop_rate98 = diffmassfrac98'; % vapor mass fraction change rate
diffP98 = diff(P98');
P_drop_rate98 = diffP98'; % cell pressure change rate - MPa/m
m_v_pipe98 = sum(m_v98,2); % total mass of vapor in pipe - kg
m_l_pipe98 = sum(m_l98,2); % total mass of vapor in pipe - kg

%% Open XTV file for 97% inlet quality and generate file ID

TRACEINT.openFile(['C:\Users\2022\OneDrive - Naval Postgraduate' ...
' School\Stonhill Thesis\TRACE Data\TRACE Runs\04JANum3\Base_Job\'...
'Base_Job.txtv']);

% Change from British units (default) to SI units.
TRACEINT.setUseSIunits( true );

% Number of timesteps from TRACE XTV file
NumTimeSteps97 = TRACEINT.getNumTimeSteps(2);

% Number of Cells in simulation
NumCells97 = 92;

% Preallocate memory
[mdotgas97,mdottot97,rho_v97,rho_l97,P97,templiq97,...
tempgas97,tempsat97,VF97] = ...
deal(NaN(NumTimeSteps97,NumCells97*2));

% Load data

for i = 1:NumCells97

if i < 10

```

```

j = '0';
else
j = '';
end

k = num2str(i);
m = i-1;

mdotgas97(:,i+m:i+1+m) = TRACEINT.getData(2,{strcat('rvmf-10A',j,k)});
mdottot97(:,i+m:i+1+m) = TRACEINT.getData(2,{strcat('rmvm-10A',j,k)});
rho_v97(:,i+m:i+1+m) = TRACEINT.getData(2,{strcat('rovn-10A',j,k)});
rho_l97(:,i+m:i+1+m) = TRACEINT.getData(2,{strcat('roln-10A',j,k)});
P97(:,i+m:i+1+m) = TRACEINT.getData(2,{strcat('pn-10A',j,k)});
templiq97(:,i+m:i+1+m) = TRACEINT.getData(2,{strcat('tln-10A',j,k)});
tempgas97(:,i+m:i+1+m) = TRACEINT.getData(2,{strcat('tvn-10A',j,k)});
tempsat97(:,i+m:i+1+m) = TRACEINT.getData(2,{strcat('tssn-10A',j,k)});
VF97(:,i+m:i+1+m) = TRACEINT.getData(2,{strcat('alpn-10A',j,k)});

end

% delete timestep columns
mdotgas97(:,1:2:end) = [];
mdottot97(:,1:2:end) = [];
rho_v97(:,1:2:end) = [];
rho_l97(:,1:2:end) = [];
P97(:,1:2:end) = [];
tempgas97(:,1:2:end) = [];
templiq97(:,1:2:end) = [];
tempsat97(:,1:2:end) = [];
VF97(:,1:2:end) = [];

% Calculated Values

% all change rates are with respect to pipe length, not time
Cells97 = 1:NumCells97;
PipeLength97 = Cells97*0.25; % Length of Pipe- m
r97 = 0.125; % pipe inner radius -m
l97 = 0.25; % cell length - m
cell_vol97 = pi*(r97^2)*l97; % cell volume - m^3
FQ97 = mdotgas97./mdottot97; % cell flow quality
mdotliq97 = mdottot97-mdotgas97; % liquid mass flow rate - kg/s
diffFQ97 = diff(FQ97');
FQ_drop_rate97 = diffFQ97'; % cell flow quality change rate
V_v97 = VF97.*cell_vol97; % cell vapor volume - m^3
V_l97 = (1.-VF97).*cell_vol97; % cell liquid volume - m^3
m_v97 = V_v97.*rho_v97; % cell vapor mass - kg
m_l97 = V_l97.*rho_l97; % cell liquid mass - kg
massfrac97 = m_v97./(m_v97+m_l97); % cell vapor mass fracion
diffmassfrac97 = diff(massfrac97');
massfrac_drop_rate97 = diffmassfrac97'; % vapor mass fraction change rate
diffP97 = diff(P97');
P_drop_rate97 = diffP97'; % cell pressure change rate - MPa/m
m_v_pipe97 = sum(m_v97,2); % total mass of vapor in pipe - kg
m_l_pipe97 = sum(m_l97,2); % total mass of vapor in pipe - kg

```

```

%% Open XTV file for 96% inlet quality and generate file ID

TRACEINT.openFile(['C:\Users\2022\OneDrive - Naval Postgraduate' ...
' School\Stonhill Thesis\TRACE Data\TRACE Runs\04JANnum4\Base_Job\'...
'Base_Job.xtv']);

% Change from British units (default) to SI units.
TRACEINT.setUseSIunits( true );

% Number of timesteps from TRACE XTV file
NumTimeSteps96 = TRACEINT.getNumTimeSteps(3);

% Number of Cells in simulation
NumCells96 = 92;

% Preallocate memory
[mdotgas96,mdottot96,rho_v96,rho_l96,P96,templiq96,...
tempgas96,tempsat96,VF96] = ...
deal(NaN(NumTimeSteps96,NumCells96*2));

% Load data

for i = 1:NumCells96

    if i < 10
        j = '0';
    else
        j = '';
    end

    k = num2str(i);
    m = i-1;

    mdotgas96(:,i+m:i+1+m) = TRACEINT.getData(3,{strcat('rvmf-10A',j,k)});
    mdottot96(:,i+m:i+1+m) = TRACEINT.getData(3,{strcat('rmvm-10A',j,k)});
    rho_v96(:,i+m:i+1+m) = TRACEINT.getData(3,{strcat('rovn-10A',j,k)});
    rho_l96(:,i+m:i+1+m) = TRACEINT.getData(3,{strcat('roln-10A',j,k)});
    P96(:,i+m:i+1+m) = TRACEINT.getData(3,{strcat('pn-10A',j,k)});
    templiq96(:,i+m:i+1+m) = TRACEINT.getData(3,{strcat('tln-10A',j,k)});
    tempgas96(:,i+m:i+1+m) = TRACEINT.getData(3,{strcat('tvn-10A',j,k)});
    tempsat96(:,i+m:i+1+m) = TRACEINT.getData(3,{strcat('tssn-10A',j,k)});
    VF96(:,i+m:i+1+m) = TRACEINT.getData(3,{strcat('alpn-10A',j,k)});
end

% delete timestep columns
mdotgas96(:,1:2:end) = [];
mdottot96(:,1:2:end) = [];
rho_v96(:,1:2:end) = [];
rho_l96(:,1:2:end) = [];
P96(:,1:2:end) = [];
tempgas96(:,1:2:end) = [];

```

```

templiq96(:,1:2:end) = [];
tempsat96(:,1:2:end) = [];
VF96(:,1:2:end) = [];

% Calculated Values

% all change rates are with respect to pipe length, not time
Cells96 = 1:NumCells96;
PipeLength96 = Cells96*0.25; % Length of Pipe- m
r96 = 0.125; % pipe inner radius -m
l96 = 0.25; % cell length - m
cell_vol96 = pi*(r96^2)*l96; % cell volume - m^3
FQ96 = mdotgas96./mdottot96; % cell flow quality
mdotliq96 = mdottot96-mdotgas96; % liquid mass flow rate - kg/s
diffFQ96 = diff(FQ96');
FQ_drop_rate96 = diffFQ96'; % cell flow quality change rate
V_v96 = VF96.*cell_vol96; % cell vapor volume - m^3
V_l96 = (1.-VF96).*cell_vol96; % cell liquid volume - m^3
m_v96 = V_v96.*rho_v96; % cell vapor mass - kg
m_l96 = V_l96.*rho_l96; % cell liquid mass - kg
massfrac96 = m_v96./(m_v96+m_l96); % cell vapor mass fraction
diffmassfrac96 = diff(massfrac96');
massfrac_drop_rate96 = diffmassfrac96'; % vapor mass fraction change rate
diffP96 = diff(P96');
P_drop_rate96 = diffP96'; % cell pressure change rate - MPa/m
m_v_pipe96 = sum(m_v96,2); % total mass of vapor in pipe - kg
m_l_pipe96 = sum(m_l96,2); % total mass of liquid in pipe - kg

%% Open XTV file for 95% inlet quality and generate file ID

TRACEINT.openFile(['C:\Users\2022\OneDrive - Naval Postgraduate' ...
'School\Stonhill Thesis\TRACE Data\TRACE Runs\04JANum5\Base_Job\'...
'Base_Job.xtv']);

% Change from British units (default) to SI units.
TRACEINT.setUseSIunits( true );

% Number of timesteps from TRACE XTV file
NumTimeSteps95 = TRACEINT.getNumTimeSteps(4);

% Number of Cells in simulation
NumCells95 = 92;

% Preallocate memory
[mdotgas95,mdottot95,rho_v95,rho_l95,P95,templiq95,...
tempgas95,tempsat95,VF95] = ...
deal(NaN(NumTimeSteps95,NumCells95*2));

% Load data

for i = 1:NumCells95

```

```

if i < 10
j = '0';
else
j = '';
end

k = num2str(i);
m = i-1;

mdotgas95(:,i+m:i+1+m) = TRACEINT.getData(4,{strcat('rvmf-10A',j,k)});
mdottot95(:,i+m:i+1+m) = TRACEINT.getData(4,{strcat('rmvm-10A',j,k)});
rho_v95(:,i+m:i+1+m) = TRACEINT.getData(4,{strcat('rovn-10A',j,k)});
rho_l95(:,i+m:i+1+m) = TRACEINT.getData(4,{strcat('roln-10A',j,k)});
P95(:,i+m:i+1+m) = TRACEINT.getData(4,{strcat('pn-10A',j,k)});
templiq95(:,i+m:i+1+m) = TRACEINT.getData(4,{strcat('tln-10A',j,k)});
tempgas95(:,i+m:i+1+m) = TRACEINT.getData(4,{strcat('tvn-10A',j,k)});
tempsat95(:,i+m:i+1+m) = TRACEINT.getData(4,{strcat('tssn-10A',j,k)});
VF95(:,i+m:i+1+m) = TRACEINT.getData(4,{strcat('alpn-10A',j,k)});

end

% delete timestep columns
mdotgas95(:,1:2:end) = [];
mdottot95(:,1:2:end) = [];
rho_v95(:,1:2:end) = [];
rho_l95(:,1:2:end) = [];
P95(:,1:2:end) = [];
tempgas95(:,1:2:end) = [];
templiq95(:,1:2:end) = [];
tempsat95(:,1:2:end) = [];
VF95(:,1:2:end) = [];

% Calculated Values

% all change rates are with respect to pipe length, not time
Cells95 = 1:NumCells95;
PipeLength95 = Cells95*0.25; % Length of Pipe- m
r95 = 0.125; % pipe inner radius -m
l95 = 0.25; % cell length - m
cell_vol95 = pi*(r95^2)*l95; % cell volume - m^3
FQ95 = mdotgas95./mdottot95; % cell flow quality
mdotliq95 = mdottot95-mdotgas95; % liquid mass flow rate - kg/s
diffFQ95 = diff(FQ95');
FQ_drop_rate95 = diffFQ95'; % cell flow quality change rate
V_v95 = VF95.*cell_vol95; % cell vapor volume - m^3
V_l95 = (1.-VF95).*cell_vol95; % cell liquid volume - m^3
m_v95 = V_v95.*rho_v95; % cell vapor mass - kg
m_l95 = V_l95.*rho_l95; % cell liquid mass - kg
massfrac95 = m_v95./(m_v95+m_l95); % cell vapor mass fracion
diffmassfrac95 = diff(massfrac95');
massfrac_drop_rate95 = diffmassfrac95'; % vapor mass fraction change rate
diffP95 = diff(P95');
P_drop_rate95 = diffP95'; % cell pressure change rate - MPa/m
m_v_pipe95 = sum(m_v95,2); % total mass of vapor in pipe - kg

```

```

m_l_pipe95 = sum(m_l95,2); % total mass of vapor in pipe - kg

%% Open XTV file for 90% inlet quality and generate file ID

TRACEINT.openFile(['C:\Users\2022\OneDrive - Naval Postgraduate' ...
' School\Stonhill Thesis\TRACE Data\TRACE Runs\04JANnum6\Base_Job\'...
'Base_Job.xtv']);

% Change from British units (default) to SI units.
TRACEINT.setUseSIunits( true );

% Number of timesteps from TRACE XTV file
NumTimeSteps90 = TRACEINT.getNumTimeSteps(5);

% Number of Cells in simulation
NumCells90 = 92;

% Preallocate memory
[mdotgas90,mdottot90,rho_v90,rho_l90,P90,templiq90,...
tempgas90,tempsat90,VF90] = ...
deal(NaN(NumTimeSteps90,NumCells90*2));

% Load data

for i = 1:NumCells90

if i < 10
j = '0';
else
j = '';
end

k = num2str(i);
m = i-1;

mdotgas90(:,i+m:i+1+m) = TRACEINT.getData(5,{strcat('rvmf-10A',j,k)});
mdottot90(:,i+m:i+1+m) = TRACEINT.getData(5,{strcat('rmvm-10A',j,k)});
rho_v90(:,i+m:i+1+m) = TRACEINT.getData(5,{strcat('rovn-10A',j,k)});
rho_l90(:,i+m:i+1+m) = TRACEINT.getData(5,{strcat('roln-10A',j,k)});
P90(:,i+m:i+1+m) = TRACEINT.getData(5,{strcat('pn-10A',j,k)});
templiq90(:,i+m:i+1+m) = TRACEINT.getData(5,{strcat('tln-10A',j,k)});
tempgas90(:,i+m:i+1+m) = TRACEINT.getData(5,{strcat('tvn-10A',j,k)});
tempsat90(:,i+m:i+1+m) = TRACEINT.getData(5,{strcat('tssn-10A',j,k)});
VF90(:,i+m:i+1+m) = TRACEINT.getData(5,{strcat('alpn-10A',j,k)});

end

% delete timestep columns
mdotgas90(:,1:2:end) = [];
mdottot90(:,1:2:end) = [];
rho_v90(:,1:2:end) = [];
rho_l90(:,1:2:end) = [];
P90(:,1:2:end) = [];

```

```

tempgas90(:,1:2:end) = [];
templiq90(:,1:2:end) = [];
tempsat90(:,1:2:end) = [];
VF90(:,1:2:end) = [];

% Calculated Values

% all change rates are with respect to pipe length, not time
Cells90 = 1:NumCells90;
PipeLength90 = Cells90*0.25; % Length of Pipe- m
r90 = 0.125; % pipe inner radius -m
l90 = 0.25; % cell length - m
cell_vol90 = pi*(r90^2)*l90; % cell volume - m^3
FQ90 = mdotgas90./mdottot90; % cell flow quality
mdotliq90 = mdottot90-mdotgas90; % liquid mass flow rate - kg/s
diffFQ90 = diff(FQ90');
FQ_drop_rate90 = diffFQ90'; % cell flow quality change rate
V_v90 = VF90.*cell_vol90; % cell vapor volume - m^3
V_l90 = (1.-VF90).*cell_vol90; % cell liquid volume - m^3
m_v90 = V_v90.*rho_v90; % cell vapor mass - kg
m_l90 = V_l90.*rho_l90; % cell liquid mass - kg
massfrac90 = m_v90./(m_v90+m_l90); % cell vapor mass fraction
diffmassfrac90 = diff(massfrac90');
massfrac_drop_rate90 = diffmassfrac90'; % vapor mass fraction change rate
diffP90 = diff(P90');
P_drop_rate90 = diffP90'; % cell pressure change rate - MPa/m
m_v_pipe90 = sum(m_v90,2); % total mass of vapor in pipe - kg
m_l_pipe90 = sum(m_l90,2); % total mass of liquid in pipe - kg

%% Open XTV file for 85% inlet quality and generate file ID

TRACEINT.openFile(['C:\Users\2022\OneDrive - Naval Postgraduate' ...
' School\Stonhill Thesis\TRACE Data\TRACE Runs\04JANum7\Base_Job\'...
'Base_Job.xtv']);

% Change from British units (default) to SI units.
TRACEINT.setUseSIunits( true );

% Number of timesteps from TRACE XTV file
NumTimeSteps85 = TRACEINT.getNumTimeSteps(6);

% Number of Cells in simulation
NumCells85 = 92;

% Preallocate memory
[mdotgas85,mdottot85,rho_v85,rho_l85,P85,templiq85,...
tempgas85,tempsat85,VF85] = ...
deal(NaN(NumTimeSteps85,NumCells85*2));

% Load data

for i = 1:NumCells85

```

```

if i < 10
j = '0';
else
j = '';
end

k = num2str(i);
m = i-1;

mdotgas85(:,i+m:i+1+m) = TRACEINT.getData(6,{strcat('rvmf-10A',j,k)});
mdottot85(:,i+m:i+1+m) = TRACEINT.getData(6,{strcat('rmvm-10A',j,k)});
rho_v85(:,i+m:i+1+m) = TRACEINT.getData(6,{strcat('rovn-10A',j,k)});
rho_l85(:,i+m:i+1+m) = TRACEINT.getData(6,{strcat('roln-10A',j,k)});
P85(:,i+m:i+1+m) = TRACEINT.getData(6,{strcat('pn-10A',j,k)});
templiq85(:,i+m:i+1+m) = TRACEINT.getData(6,{strcat('tln-10A',j,k)});
tempgas85(:,i+m:i+1+m) = TRACEINT.getData(6,{strcat('tvn-10A',j,k)});
tempsat85(:,i+m:i+1+m) = TRACEINT.getData(6,{strcat('tssn-10A',j,k)});
VF85(:,i+m:i+1+m) = TRACEINT.getData(6,{strcat('alpn-10A',j,k)});

end

% delete timestep columns
mdotgas85(:,1:2:end) = [];
mdottot85(:,1:2:end) = [];
rho_v85(:,1:2:end) = [];
rho_l85(:,1:2:end) = [];
P85(:,1:2:end) = [];
tempgas85(:,1:2:end) = [];
templiq85(:,1:2:end) = [];
tempsat85(:,1:2:end) = [];
VF85(:,1:2:end) = [];

% Calculated Values

% all change rates are with respect to pipe length, not time
Cells85 = 1:NumCells85;
PipeLength85 = Cells85*0.25; % Length of Pipe- m
r85 = 0.125; % pipe inner radius -m
l85 = 0.25; % cell length - m
cell_vol85 = pi*(r85^2)*l85; % cell volume - m^3
FQ85 = mdotgas85./mdottot85; % cell flow quality
mdotliq85 = mdottot85-mdotgas85; % liquid mass flow rate - kg/s
diffFQ85 = diff(FQ85');
FQ_drop_rate85 = diffFQ85'; % cell flow quality change rate
V_v85 = VF85.*cell_vol85; % cell vapor volume - m^3
V_l85 = (1.-VF85).*cell_vol85; % cell liquid volume - m^3
m_v85 = V_v85.*rho_v85; % cell vapor mass - kg
m_l85 = V_l85.*rho_l85; % cell liquid mass - kg
massfrac85 = m_v85./(m_v85+m_l85); % cell vapor mass fracion
diffmassfrac85 = diff(massfrac85');
massfrac_drop_rate85 = diffmassfrac85'; % vapor mass fraction change rate
diffP85 = diff(P85');
P_drop_rate85 = diffP85'; % cell pressure change rate - MPa/m
m_v_pipe85 = sum(m_v85,2); % total mass of vapor in pipe - kg

```

```
m_l_pipe85 = sum(m_l85,2); % total mass of vapor in pipe - kg
```

```
%% Plot Data
```

```
% Pressure
```

```
figure
hold on
grid on
plot(Cells85*0.25,P85(end,:)/10^6,'LineWidth',3)
plot(Cells90*0.25,P90(end,:)/10^6,'LineWidth',3)
plot(Cells95*0.25,P95(end,:)/10^6,'LineWidth',3)
plot(Cells96*0.25,P96(end,:)/10^6,'LineWidth',3)
plot(Cells97*0.25,P97(end,:)/10^6,'LineWidth',3)
plot(Cells98*0.25,P98(end,:)/10^6,'LineWidth',3)
plot(Cells99*0.25,P99(end,:)/10^6,'LineWidth',3)
xline(2,'k','LineWidth',1)
xline(4,'k','LineWidth',1)
xline(10,'k','LineWidth',1)
xline(13,'k','LineWidth',1)
xline(21,'k','LineWidth',1)
set(gca,'xtick',0:1:23)
xlim([0,NumCells99*0.25]);
xlabel('Pipe Length (m)');
ylabel('Pressure (MPa)');
legend('85%','90%','95%','96%','97%','98%','99%','Pipe Bend',...
'Location','Northeast','NumColumns',4)
```

```
% Pressure Drop Rate
```

```
figure
hold on
grid on
plot(Cells85(2:NumCells85)*0.25,P_drop_rate85(end,:)/10^3,'LineWidth',3);
plot(Cells90(2:NumCells90)*0.25,P_drop_rate90(end,:)/10^3,'LineWidth',3);
plot(Cells95(2:NumCells95)*0.25,P_drop_rate95(end,:)/10^3,'LineWidth',3);
plot(Cells96(2:NumCells96)*0.25,P_drop_rate96(end,:)/10^3,'LineWidth',3);
plot(Cells97(2:NumCells97)*0.25,P_drop_rate97(end,:)/10^3,'LineWidth',3);
plot(Cells98(2:NumCells98)*0.25,P_drop_rate98(end,:)/10^3,'LineWidth',3);
plot(Cells99(2:NumCells99)*0.25,P_drop_rate99(end,:)/10^3,'LineWidth',3);
xline(2,'k','LineWidth',1)
xline(4,'k','LineWidth',1)
xline(10,'k','LineWidth',1)
xline(13,'k','LineWidth',1)
xline(21,'k','LineWidth',1)
set(gca,'xtick',0:1:23)
xlim([0,NumCells99*0.25]);
ylim([-3,0.5]);
xlabel('Pipe Length (m)');
ylabel('Pressure Change Rate (kPa/m)');
legend('85%','90%','95%','96%','97%','98%','99%','Pipe Bend',...
'Location','Northeast','NumColumns',4)
```

```

% Flow Quality
figure
hold on
grid on
plot(Cells85*185,FQ85(end,:), 'LineWidth',3)
plot(Cells90*190,FQ90(end,:), 'LineWidth',3)
plot(Cells95*195,FQ95(end,:), 'LineWidth',3)
plot(Cells96*196,FQ96(end,:), 'LineWidth',3)
plot(Cells97*197,FQ97(end,:), 'LineWidth',3)
plot(Cells98*198,FQ98(end,:), 'LineWidth',3)
plot(Cells99*199,FQ99(end,:), 'LineWidth',3)
xline(2,'k','LineWidth',1)
xline(4,'k','LineWidth',1)
xline(10,'k','LineWidth',1)
xline(13,'k','LineWidth',1)
xline(21,'k','LineWidth',1)
axis([0 NumCells99*0.25 0.84 1]);
xlabel('Pipe Length (m)');
ylabel('Flow Quality');
set(gca,'xtick',0:1:23)
set(gca,'ytick',0:0.02:1)
xlim([0,NumCells99*0.25]);
ylim([0.8,1]);
legend('85%', '90%', '95%', '96%', '97%', '98%', '99%', 'Pipe Bend', ...
'Location', 'Southeast', 'NumColumns',4);

% Vapor Temperature
figure
hold on
grid on
plot(Cells85*185,tempgas85(end,:), 'LineWidth',3)
plot(Cells90*190,tempgas90(end,:), 'LineWidth',3)
plot(Cells95*195,tempgas95(end,:), 'LineWidth',3)
plot(Cells96*196,tempgas96(end,:), 'LineWidth',3)
plot(Cells97*197,tempgas97(end,:), 'LineWidth',3)
plot(Cells98*198,tempgas98(end,:), 'LineWidth',3)
plot(Cells99*199,tempgas99(end,:), 'LineWidth',3)
plot(Cells85*185,tempsat85(end,:), 'k.', 'LineWidth',3)
xline(2,'k','LineWidth',1)
xline(4,'k','LineWidth',1)
xline(10,'k','LineWidth',1)
xline(13,'k','LineWidth',1)
xline(21,'k','LineWidth',1)
set(gca,'xtick',0:1:23);
set(gca,'ytick',480:1:510);
grid on
xlim([0,NumCells99*0.25]);
ylim([499,509]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
legend('85%', '90%', '95%', '96%', '97%', '98%', '99%', 'Saturation', ...
'Pipe Bend', 'Location', 'Northeast', 'NumColumns',4)

```

```

% Liquid Temperature
figure
hold on
grid on
plot(Cells85*185,templiq85(end,:), 'LineWidth',3)
plot(Cells90*190,templiq90(end,:), 'LineWidth',3)
plot(Cells95*195,templiq95(end,:), 'LineWidth',3)
plot(Cells96*196,templiq96(end,:), 'LineWidth',3)
plot(Cells97*197,templiq97(end,:), 'LineWidth',3)
plot(Cells98*198,templiq98(end,:), 'LineWidth',3)
plot(Cells99*199,templiq99(end,:), 'LineWidth',3)
xline(2, 'k', 'LineWidth',1)
xline(4, 'k', 'LineWidth',1)
xline(10, 'k', 'LineWidth',1)
xline(13, 'k', 'LineWidth',1)
xline(21, 'k', 'LineWidth',1)
set(gca, 'xtick', 0:1:23);
set(gca, 'ytick', 480:0.1:510);
grid on
xlim([0, NumCells99*0.25]);
ylim([499.7, 500.4]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
legend('85%', '90%', '95%', '96%', '97%', '98%', '99%', ...
'Pipe Bend', 'Location', 'Northeast', 'NumColumns', 4)

```

```

% Cell Vapor Mass Fraction
figure
hold on
grid on
plot(Cells85*0.25, massfrac85(end,:), 'LineWidth',5)
plot(Cells90*0.25, massfrac90(end,:), 'LineWidth',5)
plot(Cells95*0.25, massfrac95(end,:), 'LineWidth',5)
plot(Cells96*0.25, massfrac96(end,:), 'LineWidth',5)
plot(Cells97*0.25, massfrac97(end,:), 'LineWidth',5)
plot(Cells98*0.25, massfrac98(end,:), 'LineWidth',5)
plot(Cells99*0.25, massfrac99(end,:), 'LineWidth',5)
xline(2, 'k', 'LineWidth',1)
xline(4, 'k', 'LineWidth',1)
xline(10, 'k', 'LineWidth',1)
xline(13, 'k', 'LineWidth',1)
xline(21, 'k', 'LineWidth',1)
xlim([0, NumCells99*0.25]);
ylim([0.77, 1]);
xlabel('Pipe Length (m)');
ylabel('Cell Vapor Mass Fraction');
grid on
set(gca, 'xtick', 0:1:23)
set(gca, 'ytick', 0:0.02:1)
legend('85%', '90%', '95%', '96%', '97%', '98%', '99%', 'Pipe Bend', ...
'Location', 'Southeast', 'NumColumns', 4);

```

```

% Total Pipe Mass

r = 0.125; % pipe inner radius -m
l = 0.25; % cell length - m
cell_vol = pi*(r^2)*l;
qualities = [0.99 0.98 0.97 0.96 0.95 0.90 0.85]; % tested inlet qualities

figure
x = categorical({'99%', '98%', '97%', '96%', '95%', '90%', '85%'});
y = [m_l_pipe99(end) m_v_pipe99(end); m_l_pipe98(end) m_v_pipe98(end); ...
m_l_pipe97(end) m_v_pipe97(end); m_l_pipe96(end) m_v_pipe96(end); ...
m_l_pipe95(end) m_v_pipe95(end); m_l_pipe90(end) m_v_pipe90(end); ...
m_l_pipe85(end) m_v_pipe85(end)];
bar(x,y,'stacked');
grid on
legend('liquid', 'vapor', 'Orientation', 'Horizontal')
xlabel('Inlet Flow Quality');
ylabel('Total Fluid Mass in Pipe (kg)');

```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G. MATLAB SCRIPT FOR GUO ET AL. OPEN LITERATURE DATA PROCESSING

```

%% Stonhill Data Processor for Guo et al. Data

% ***** CHANGE FROM '2022' TO 'garth' for personal/lab computer*****

% Tidy Up!
close all;
clear;
clc;

addpath ['C:\Users\garth\OneDrive - Naval Postgraduate School\Stonhill...
'Thesis\Open Literature Data\Guo Two Phase Steam\XSteam_Matlab_v2.6']

%% Import data digitized from Guo et al. 2nd PDO Fig. 12

Temp_Fluid_i      = importdata('Guo_fluidtemp_a.csv');      % deg C
Temp_Inner_Wall_i = importdata('Guo_innerwalltemp_a.csv'); % deg C
Temp_Outer_Wall_i = importdata('Guo_outerwalltemp_a.csv'); % deg C
Pressure_Inlet_i  = importdata('Guo_inletpressure_a.csv');  % MPa
Pressure_Outlet_i = importdata('Guo_outletpressure_a.csv'); % MPa
Pressure_Drop_i   = importdata('Guo_pressuredrop_a.csv');   % kPa
Mass_Flux_Rate_i  = importdata('Guo_massflowrate_a.csv');   % kg/m2s
h_i              = importdata('Guo_Heat_Transfer_Coefficient_a.csv'); % W/m2s

%% Parse data

% measurement data - digitized from Guo, et. al
Temp_Fluid      = Temp_Fluid_i(:,2)+273.15;      % Fluid temp - K
Temp_Inner_Wall = Temp_Inner_Wall_i(:,2)+273.15; % Inner wall temp - K
Temp_Outer_Wall = Temp_Outer_Wall_i(:,2)+273.15; % Outer wall temp - K
Pressure_Inlet  = Pressure_Inlet_i(:,2);         % Outlet pressure - MPa
Pressure_Outlet = Pressure_Outlet_i(:,2);         % Inlet pressure - MPa
Pressure_Drop   = Pressure_Drop_i(:,2)./1000;    % Pressure drop - MPa
Mass_Flux_Rate  = Mass_Flux_Rate_i(:,2);         % Mass flux rate - kg/m2s
Heat_Trans_Coeff = h_i(:,2); % Fluid/wall heat transfer coefficient - W/m2K

% Correspondng time data - digitized from Guo, et. al
Temp_Fluid_t      = Temp_Fluid_i(:,1);          % Fluid temperature - s
Temp_Inner_Wall_t = Temp_Inner_Wall_i(:,1);      % Inner wall temperature - s
Temp_Outer_Wall_t = Temp_Outer_Wall_i(:,1);      % Outer wall temperature - s
Pressure_Inlet_t   = Pressure_Inlet_i(:,1);      % Outlet pressure - s
Pressure_Outlet_t  = Pressure_Outlet_i(:,1);      % Inlet pressure - s
Pressure_Drop_t    = Pressure_Drop_i(:,1);       % Pressure drop - s
Mass_Flux_Rate_t   = Mass_Flux_Rate_i(:,1);      % Mass flux rate - s
Heat_Trans_Coeff_t = h_i(:,1); % Fluid/wall heat transfer coefficient - s

%% Interpolate data sets to the same length

data_length = 1000; % set data set length

```

```

% largest time value
longest_time = max([max(Temp_Inner_Wall_t);max(Temp_Outer_Wall_t);...
max(Temp_Fluid_t);max(Mass_Flux_Rate_t);max(Pressure_Inlet_t);...
max(Pressure_Outlet_t);max(Pressure_Drop_t)]);

time_interp      =...
linspace(0, longest_time, data_length);           % s

Temp_Fluid_interp      =...
interp1(Temp_Fluid_t, Temp_Fluid, time_interp);    % K

Temp_Inner_Wall_interp =...
interp1(Temp_Inner_Wall_t, Temp_Inner_Wall, time_interp); % K

Temp_Outer_Wall_interp =...
interp1(Temp_Outer_Wall_t, Temp_Outer_Wall, time_interp); % K

Pressure_Inlet_interp  =...
interp1(Pressure_Inlet_t, Pressure_Inlet, time_interp); % MPa

Pressure_Outlet_interp =...
interp1(Pressure_Outlet_t, Pressure_Outlet, time_interp); % MPa

Pressure_Drop_interp   =...
interp1(Pressure_Drop_t, Pressure_Drop, time_interp);    % MPa

Mass_Flux_Rate_interp  =...
interp1(Mass_Flux_Rate_t, Mass_Flux_Rate, time_interp);  % kg/m2s

Heat_Trans_Coeff_interp =...
interp1(Heat_Trans_Coeff_t, Heat_Trans_Coeff, time_interp); % W/m2K

%% Calculate inlet enthalpy and void fraction

% Pre-allocate memory
Enthalpy_Inlet_TV = NaN(1,1000);
X_Inlet_TV = NaN(1,1000);

% Inlet steam table calculations - all enthalpies that are commented out
% will result in a quality calculation of 1, 0, or NaN. Quality was only
% successfully estimated using enthalpy (calculated from fluid saturation
% temperature) and pressure

for i=1:length(time_interp)

% Enthalpy_Inlet_PV(i) = XSteam...
% ('hV_p',Pressure_Inlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% sat vapor enthalpy as a function of pressure - kJ/kg

% Enthalpy_Inlet_PL(i) = XSteam...
% ('hL_p',Pressure_Inlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% sat liquid enthalpy as a function of pressure - kJ/kg

```

```

Enthalpy_Inlet_TV(i) = XSteam...
('hV_T',Temp_Fluid_interp(i)-273.15);
% sat vapor enthalpy as a function of temperature - kJ/kg

% Enthalpy_Inlet_TL(i) = XSteam...
% ('hL_T',Pressure_Inlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% sat liquid enthalpy as a function of temperature - kJ/kg

% X_Inlet_PV(i) = XSteam...
% ('x_ph',Pressure_Inlet_interp(i).*10,Enthalpy_Inlet_PV(i));
% - vapor mass fraction from sat vapor pressure based enthalpy and pressure

% X_Inlet_PL(i) = XSteam...
% ('x_ph',Pressure_Inlet_interp(i).*10,Enthalpy_Inlet_PL(i));
% vapor mass fraction from sat vapor pressure based enthalpy and pressure

X_Inlet_TV(i) = XSteam...
('x_ph',Pressure_Inlet_interp(i).*10,Enthalpy_Inlet_TV(i));
% vapor mass fraction from sat vapor temp based enthalpy and pressure

% X_Inlet_TL(i) = XSteam...
% ('x_ph',Pressure_Inlet_interp(i).*10,Enthalpy_Inlet_TL(i));
% vapor mass fraction from sat vapor temp based enthalpy and pressure

end

%% Calculate outlet enthalpy and void fraction

% preallocate memory
Enthalpy_Outlet_TV = NaN(1,1000);
X_Outlet_TV = NaN(1,1000);

% % Outlet steam table calculations - all enthalpies that are commented out
% % will result in a quality calculation of 1, 0, or NaN. Quality was only
% % succesfully estimated using enthalpy (calculated from fluid saturation
% % temperature) and pressure

for i=1:length(time_interp)

% Enthalpy_Outlet_PV(i) = XSteam...
% ('hV_p',Pressure_Outlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% % sat vapor enthalpy as a function of pressure - kJ/kg

% Enthalpy_Outlet_PL(i) = XSteam...
% ('hL_p',Pressure_Outlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% % sat liquid enthalpy as a function of pressure - kJ/kg

Enthalpy_Outlet_TV(i) = XSteam...
('hV_T',Temp_Fluid_interp(i)-273.15);
% % sat vapor enthalpy as a function of temperature - kJ/kg

% Enthalpy_Outlet_TL(i) = XSteam...
% ('hL_T',Pressure_Outlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% % sat liquid enthalpy as a function of temp - kJ/kg

```

```

% X_Outlet_PV(i)      = XSteam...
% ('x_ph',Pressure_Outlet_interp(i).*10,Enthalpy_Outlet_PV(i));
% % vapor mass fraction from sat vapor pressure based enthalpy and pressure

% X_Outlet_PL(i)      = XSteam...
% ('x_ph',Pressure_Outlet_interp(i).*10,Enthalpy_Outlet_PL(i));
% % vapor mass fraction from sat vapor pressure based enthalpy and pressure

X_Outlet_TV(i)      = XSteam...
('x_ph',Pressure_Outlet_interp(i).*10,Enthalpy_Outlet_TV(i));
% % vapor mass fraction from sat vapor temp based enthalpy and pressure

% X_Outlet_TL(i)      = XSteam...
% ('x_ph',Pressure_Outlet_interp(i).*10,Enthalpy_Outlet_TL(i));
% % vapor mass fraction from sat vapor temp based enthalpy and pressure

end

%% Plot interpolated and calculated Data

figure(1)
subplot(6,1,1)
hold on
plot(time_interp,Temp_Fluid_interp,'LineWidth',3);
plot(time_interp,Temp_Inner_Wall_interp,'LineWidth',3);
plot(time_interp,Temp_Outer_Wall_interp,'LineWidth',3);
legend('Fluid','Inner Wall','Outer Wall');
title('Temperature')
ylabel('(K)')

subplot(6,1,2)
hold on
plot(time_interp,Pressure_Inlet_interp,'LineWidth',3);
plot(time_interp,Pressure_Outlet_interp,'LineWidth',3);
legend('Inlet','Outlet');
title('Pressure')
ylabel('(MPa)')

subplot(6,1,3)
hold on
plot(time_interp,Mass_Flux_Rate_interp,'LineWidth',3);
title('Mass Flux Rate')
ylabel('(kg/m2s)')

subplot(6,1,4)
hold on
plot(time_interp,Heat_Trans_Coeff_interp,'LineWidth',3);
title('Wall Heat Transfer Coefficient')
ylabel('(W/m2K)')

subplot(6,1,5)
hold on
plot(time_interp,Enthalpy_Inlet_TV,'LineWidth',3);

```

```

plot(time_interp,Enthalpy_Outlet_TV,'LineWidth',3);
title('Calculated Enthalpy')
legend('Inlet','Outlet')
ylabel('kJ/kg')

subplot(6,1,6)
hold on
xlabel('time (s)')
plot(time_interp,X_Inlet_TV,'LineWidth',3);
plot(time_interp,X_Outlet_TV,'LineWidth',3);
legend('Inlet','Outlet')
title('Calculated Vapor Mass Fraction')
sgtitle('Guo Experimental Data for Two-Phase Pressure Oscillations')

%% Index Steady State Data Inputs (75 seconds by inspection)

time_SS = time_interp(253);
Temp_Fluid_SS = Temp_Fluid_interp(253);
Temp_Inner_Wall_SS = Temp_Inner_Wall_interp(253);
Temp_Outer_Wall_SS = Temp_Outer_Wall_interp(253);
Pressure_Delta_SS = Pressure_Inlet_interp(253)-Pressure_Outlet_interp(253);
Pressure_Outlet_SS = XSteam('psat_T',Temp_Fluid_SS-273.15)*0.1;
Pressure_Inlet_SS = Pressure_Outlet_SS + Pressure_Delta_SS;
Mass_Flux_Rate_SS = Mass_Flux_Rate_interp(253);
Heat_Trans_Coeff_SS = Heat_Trans_Coeff_interp(253);
Enthalpy_Inlet_TV_SS = Enthalpy_Inlet_TV(253);
Enthalpy_Outlet_TV_SS = Enthalpy_Outlet_TV(253);
X_Inlet_TV_SS = X_Inlet_TV(253);
X_Outlet_TV_SS = X_Outlet_TV(253);

```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX H. MATLAB SCRIPT FOR TRACE INTERFACING AND TRANSIENT CASE GRAPH AND VIDEO GENERATION

```
%% TRACE Interface Data Processor Pressure and Vapor Mass Fraction Transient

% ***** CHANGE THE FOLDER OF THE XTV FILE YOU WISH TO ANALYZE*****
% ***** CHANGE FROM '2022' TO 'garth' for personal/lab computer*****

% Tidy Up!
close all;
clear;
clc;

% Add TRACE-MATLAB Interface to path
javaaddpath('C:\Users\garth\snap\lib\cafean.jar')
javaaddpath('C:\Users\garth\snap\lib\matlab_acs.jar')
javaaddpath('C:\Users\garth\snap\plugins\extdata.jar')
javaaddpath('C:\Users\garth\snap\plugins\TRACE.jar')
import com.cafean.matlab.*
TRACEINT = MatlabTraceIntf; % handle for MATLAB-TRACE interface

% Open XTV file and generate file ID - Change if needed
TRACEINT.openFile(['C:\Users\garth\OneDrive - Naval Postgraduate' ...
' School\Stonhill Thesis\TRACE Data\TRACE Runs\27FEBnum2\Base_Job\'...
'Base_Job.xtv']);

% Change from British units (default) to SI units.
TRACEINT.setUseSIunits( true );

% Number of timesteps from TRACE XTV file
NumTimeSteps = TRACEINT.getNumTimeSteps(0);

% Number of cells in simulation
NumCells = 92;

% Preallocate memory
[mdotgas,mdottot,rho_v,rho_l,P,qwalli,qwallo,templiq,tempgas,tempsat,...
tempwalli,tempwallo,VF] = deal(NaN(NumTimeSteps,NumCells*2));

%% Load data

for i = 1:NumCells

    if i < 10
        j = '0';
    else
        j = '';
    end

    k = num2str(i);
    m = i-1;
```

```

mdotgas(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('rvmf-10A',j,k)});
mdottot(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('rmvm-10A',j,k)});
rho_v(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('rovn-10A',j,k)});
rho_l(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('roln-10A',j,k)});
P(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('pn-10A',j,k)});
qwalli(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('qppi-10001A',j,k)});
qwallo(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('qppo-10001A',j,k)});
templiq(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tln-10A',j,k)});
tempgas(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tvn-10A',j,k)});
tempsat(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tssn-10A',j,k)});
tempwalli(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tsurfi-10001A',j,k)});
tempwallo(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tsurfo-10001A',j,k)});
VF(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('alpn-10A',j,k)});
end

```

% delete timestep columns

```

mdotgas(:,1:2:end) = [];
mdottot(:,1:2:end) = [];
rho_v(:,1:2:end) = [];
rho_l(:,1:2:end) = [];
P(:,1:2:end) = [];
qwalli(:,1:2:end) = [];
qwallo(:,1:2:end) = [];
tempgas(:,1:2:end) = [];
templiq(:,1:2:end) = [];
tempsat(:,1:2:end) = [];
tempwalli(:,1:2:end) = [];
tempwallo(:,1:2:end) = [];
VF(:,1:2:end) = [];

```

% Calculated Values

% all change rates are with respect to pipe length, not time

```

Cells = 1:NumCells; % Cell array
PipeLength = Cells*0.25; % Length of Pipe- m
r = 0.125; % pipe inner radius -m
l = 0.25; % cell length - m
cell_vol = pi*(r^2)*l; % cell volume - m^3
FQ = mdotgas./mdottot; % flow quality
mdotliq = mdottot-mdotgas; % liquid mass flow rate - kg/s
diffFQ = diff(FQ');
FQ_drop_rate = diffFQ'; % flow quality change rate
V_v = VF.*cell_vol; % vapor volume - m^3
V_l = (1.-VF).*cell_vol; % liquid volume - m^3
m_v = V_v.*rho_v; % vapor mass - kg
m_l = V_l.*rho_l; % liquid mass - kg
massfrac = m_v./(m_v+m_l); % vapor mass fraction
diffmassfrac = diff(massfrac');
massfrac_drop_rate = diffmassfrac'; % vapor mass fraction change rate
diffP = diff(P');
P_drop_rate = diffP'; % pressure change rate - MPa/m
h = qwalli./(tempwalli-tempgas); % heat transfer coefficient - W/m^2K
m_v_pipe = sum(m_v,2); % total mass of vapor in pipe - kg

```

```

m_l_pipe          = sum(m_l,2);          % total mass of vapor in pipe - kg

%% Plot Data

% Uncomment and run each section individually for each plot and video
% Uncomment video frame saving code if saving data

%% Pressure

i = 1:NumTimeSteps; %timestep;

figure('units','normalized','outerposition',[0 0 1 1]) % maximize figure
hold on
T = tiledlayout(2,1,'TileSpacing','compact');

nexttile
hold on
A = plot(PipeLength,P(i(1),:)/10^6,'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
set(gca,'ytick',2.1:0.1:2.8);
grid on
xlim([0,NumCells*0.25]);
ylim([2.1,2.8]);
ylabel('Pressure (Mpa)','FontSize',17);
legend('','Pipe Bend','FontSize',15);

nexttile
hold on
B = plot(PipeLength(2:end),P_drop_rate(i(1),:)/10^6,'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
grid on
xlim([0,NumCells*0.25]);
ylim([-0.005,0.005]);
ylabel('Pressure Change Rate (MPa/m)','FontSize',17);
xlabel('Pipe Length (m)','FontSize',15);
legend('','Pipe Bend','FontSize',15);

A.YDataSource = 'Ay'; % Pressure handle
B.YDataSource = 'By'; % Pressure change rate handle

Title = {'27FEB2_Pressure1.mp4','27FEB2_Pressure2.mp4',...
'27FEB2_Pressure3.mp4','27FEB2_Pressure4.mp4'};
Frame_Rate = [3,30,3,30];

```

```

for k = 1:4

if k == 1
range = 1:10;
end
if k == 2
range = 11:299;
end
if k == 3
range = 300:360;
end
if k == 4
range = 361:667;
end

Video = VideoWriter(char(Title(k)), 'MPEG-4');
Video.FrameRate = Frame_Rate(k);
open(Video);

for j = range

Ay = P(j,:)/10^6;
By = P_drop_rate(j,:)/10^6;
title(T,sprintf('Time Elapsed: %.0f seconds', j),'FontSize',17)

refreshdata
drawnow

Frame = getframe(gcf) ;
writeVideo(Video, Frame)

end

close(Video);

end

%% Temperature

i = 1:NumTimeSteps; %timestep;

figure('units','normalized','outerposition',[0 0 1 1]) % maximize figure
hold on
A = plot(PipeLength,tempgas(i(1),:),'LineWidth',3);
B = plot(PipeLength,tempqliq(i(1),:),'LineWidth',3);
C = plot(PipeLength,tempwalli(i(1),:),'LineWidth',3);
D = plot(PipeLength,tempwallo(i(1),:),'LineWidth',3);
E = plot(PipeLength,tempsat(i(1),:),'k.','LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);

```

```

xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
set(gca,'ytick',490:1:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)','FontSize',15);
ylabel('Temperature (K)','FontSize',15);
time_legend = legend('Liquid','Vapor','Inner Wall','Outer Wall',...
'Saturation','Pipe Bend','','','Location','Northeast','FontSize',...
17,'Orientation','horizontal');

A.YDataSource = 'Ay'; % Liquid temperature handle
B.YDataSource = 'By'; % Vapor temperature handle
C.YDataSource = 'Cy'; % Inner wall temperature handle
D.YDataSource = 'Dy'; % Outer wall temperature handle
E.YDataSource = 'Ey'; % Saturation temperature handle

Title = {'27FEB2_Temp1.mp4','27FEB2_Temp2.mp4','27FEB2_Temp3.mp4',...
'27FEB2_Temp4.mp4'};
Frame_Rate = [3,30,3,30];

for k = 1:4

if k == 1
range = 1:10;
end
if k == 2
range = 11:299;
end
if k == 3
range = 300:360;
end
if k == 4
range = 361:667;
end

Video = VideoWriter(char(Title(k)),'MPEG-4');
Video.FrameRate = Frame_Rate(k);
open(Video);

for j = range

Ay = tempgas(j,:);
By = templiq(j,:);
Cy = tempwalli(j,:);
Dy = tempwallo(j,:);
Ey = tempsat(j,:);
title(time_legend,sprintf('Time Elapsed: %.0f seconds', j))
refreshdata
drawnow

Frame = getframe(gcf) ;
writeVideo(Video, Frame)

```

```

end

close(Video);

end

%% Mass Flow Rate

i = 1:NumTimeSteps; %timestep;

figure('units','normalized','outerposition',[0 0 1 1]) % maximize figure
hold on
T = tiledlayout(3,1);

nexttile
hold on
A = plot(PipeLength,mdotliq(i(1),:),'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
set(gca,'ytick',0:0.05:5);
grid on
xlim([0,NumCells*0.25]);
ylim([0.35,0.7]);
ylabel(nexttile(1),'Liquid','FontSize',15);

nexttile
hold on
B = plot(PipeLength,mdotgas(i(1),:),'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
set(gca,'ytick',25.2:0.05:25.5);
grid on
xlim([0,NumCells*0.25]);
ylim([25.2,25.5]);
ylabel(nexttile(2),'Vapor','FontSize',15);

nexttile
hold on
C = plot(PipeLength,mdottot(i(1),:),'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);

```

```

set(gca,'xtick',0:1:23);
set(gca,'ytick',24:0.05:29);
grid on
xlim([0,NumCells*0.25]);
ylim([25.7,26]);
xlabel(T,'Pipe Length (m)','FontSize',15);
ylabel(nexttile(3),'Total','FontSize',15);
ylabel(T,'Mass Flow Rate (kg/s)','FontSize',17);

A.YDataSource = 'Ay'; % Liquid mass flow handle
B.YDataSource = 'By'; % Vapor mass flow handle
C.YDataSource = 'Cy'; % Total mass flow handle

Title = {'27FEB2_md0t1.mp4','27FEB2_md0t2.mp4','27FEB2_md0t3.mp4',...
'27FEB2_md0t4.mp4'};
Frame_Rate = [3,30,3,30];

for k = 1:4

if k == 1
range = 1:10;
end
if k == 2
range = 11:299;
end
if k == 3
range = 300:360;
end
if k == 4
range = 361:667;
end

Video = VideoWriter(char(Title(k)),'MPEG-4');
Video.FrameRate = Frame_Rate(k);
open(Video);

for j = range

Ay = mdotliq(j,:);
By = mdotgas(j,:);
Cy = mdottot(j,:);
title(T,sprintf('Time Elapsed: %.0f seconds', j),'FontSize',17)

refreshdata
drawnow

Frame = getframe(gcf) ;
writeVideo(Video, Frame)

end

close(Video);

end

```

```

%% Flow Quality

i = 1:NumTimeSteps; %timestep;

figure('units','normalized','outerposition',[0 0 1 1]) % maximize figure
hold on
T = tiledlayout(2,1,'TileSpacing','compact');

nexttile
hold on
A = plot(PipeLength,FQ(i(1),:),'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
set(gca,'ytick',0.95:0.001:1);
grid on
xlim([0,NumCells*0.25]);
ylim([0.975,0.985]);
ylabel('Flow Quality','FontSize',17);
legend('','Pipe Bend','FontSize',15);

nexttile
hold on
B = plot(PipeLength(2:end),FQ_drop_rate(i(1),:),'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
grid on
xlim([0,NumCells*0.25]);
ylim([-0.001,0.001]);
ylabel('Flow Quality Change Rate (/m)','FontSize',17);
xlabel('Pipe Length (m)','FontSize',15);
legend('','Pipe Bend','FontSize',15);

A.YDataSource = 'Ay'; % Flow Quality handle
B.YDataSource = 'By'; % Flow Quality change rate handle

Title = {'27FEB2_FQ1.mp4','27FEB2_FQ2.mp4','27FEB2_FQ3.mp4',...
'27FEB2_FQ4.mp4'};
Frame_Rate = [3,30,3,30];

for k = 1:4

if k == 1
range = 1:10;
end
if k == 2
range = 11:299;

```

```

end
if k == 3
range = 300:360;
end
if k == 4
range = 361:667;
end

Video = VideoWriter(char(Title(k)), 'MPEG-4');
Video.FrameRate = Frame_Rate(k);
open(Video);

for j = range

Ay = FQ(j,:);
By = FQ_drop_rate(j,:);
title(T,sprintf('Time Elapsed: %.0f seconds', j), 'FontSize', 17)

refreshdata
drawnow

Frame = getframe(gcf) ;
writeVideo(Video, Frame)

end

close(Video);

end

%% Vapor Mass Fraction

i = 1:NumTimeSteps; %timestep;

figure('units','normalized','outerposition',[0 0 1 1]) % maximize figure
hold on

hold on
A = plot(PipeLength, massfrac(i(1),:), 'LineWidth', 3);
xline(2, 'k', 'LineWidth', 1);
xline(4, 'k', 'LineWidth', 1);
xline(10, 'k', 'LineWidth', 1);
xline(13, 'k', 'LineWidth', 1);
xline(21, 'k', 'LineWidth', 1);
set(gca, 'xtick', 0:1:23);
set(gca, 'ytick', 0.955:0.005:1);
grid on
xlim([0, NumCells*0.25]);
ylim([0.955, 1]);
ylabel('Vapor Mass Fraction', 'FontSize', 15);
xlabel('Pipe Length (m)', 'FontSize', 15);
legend('', 'Pipe Bend', 'FontSize', 15);

```

```

A.YDataSource = 'Ay'; % Vapor Mass Fraction handle

Title =
{'27FEB2_massfrac1.mp4','27FEB2_massfrac2.mp4','27FEB2_massfrac3.mp4',...
'27FEB2_massfrac4.mp4'};
Frame_Rate = [3,30,3,30];

for k = 1:4

    if k == 1
        range = 1:10;
    end
    if k == 2
        range = 11:299;
    end
    if k == 3
        range = 300:360;
    end
    if k == 4
        range = 361:667;
    end

    Video = VideoWriter(char(Title(k)),'MPEG-4');
    Video.FrameRate = Frame_Rate(k);
    open(Video);

    for j = range

        Ay = massfrac(j,:);

        title(sprintf('Time Elapsed: %.0f seconds', j),'FontSize',17)

        refreshdata
        drawnow

        Frame = getframe(gcf) ;
        writeVideo(Video, Frame)

    end

    close(Video);

end

%% Heat Flux and Heat Transfer Coefficient

i = 1:NumTimeSteps; %timestep;

figure('units','normalized','outerposition',[0 0 1 1]) % maximize figure
hold on
T = tiledlayout(2,1,'TileSpacing','compact');

nexttile

```

```

hold on
A = plot(PipeLength,-qwalli(i(1),:),'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
set(gca,'ytick',-15000:2000:15000);
grid on
xlim([0,NumCells*0.25]);
ylim([-15000,15000]);
ylabel('Inner Wall Heat Flux (W/m^2)','FontSize',15);
legend('','Pipe Bend','FontSize',15);

nexttile
hold on
B = plot(PipeLength,h(i(1),:),'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
grid on
xlim([0,NumCells*0.25]);
ylim([-3000,3000]);
ylabel('Heat Transfer Coefficient (W/m^2 K)','FontSize',15);
xlabel('Pipe Length (m)','FontSize',15);
legend('','Pipe Bend','FontSize',15);

A.YDataSource = 'Ay'; % Heat Flux handle
B.YDataSource = 'By'; % Heat transfer coefficient handle

Title = {'27FEB2_qwalli1.mp4','27FEB2_qwalli2.mp4','27FEB2_qwalli3.mp4',...
'27FEB2_qwalli4.mp4'};
Frame_Rate = [3,30,3,30];

for k = 1:4

if k == 1
range = 1:10;
end
if k == 2
range = 11:299;
end
if k == 3
range = 300:360;
end
if k == 4
range = 361:667;
end

Video = VideoWriter(char(Title(k)),'MPEG-4');

```

```

Video.FrameRate = Frame_Rate(k);
open(Video);

for j = range

Ay = -qwalli(j,:);
By = h(j,:);
title(T,sprintf('Time Elapsed: %.0f seconds', j),'FontSize',17);

refreshdata
drawnow

Frame = getframe(gcf) ;
writeVideo(Video, Frame)

end

close(Video);

end

%% Total Pipe Mass over time

figure('units','normalized','outerposition',[0 0 1 1]) % maximize figure
hold on
T = tiledlayout(3,1,'TileSpacing','compact');

nexttile
hold on
xline(300,'k','LineWidth',1);
xline(360,'k','LineWidth',1);
plot(1:length(m_l_pipe),m_l_pipe,'LineWidth',3)
grid on
set(gca,'xtick',0:25:666)
xlim([0,666]);
ylabel('Total Pipe Liquid Mass (kg)');
title(T,'Total Pipe Fluid Mass for Varying Inlet Quality')
legend('transient initiation','transient termination')

nexttile
hold on
xline(300,'k','LineWidth',1);
xline(360,'k','LineWidth',1);
plot(1:length(m_v_pipe),m_v_pipe,'LineWidth',3)
grid on
set(gca,'xtick',0:25:666)
xlim([0,666]);
ylabel('Total Pipe Vapor Mass (kg)');

nexttile
hold on
xline(300,'k','LineWidth',1);
xline(360,'k','LineWidth',1);

```

```

plot(1:length(m_v_pipe),m_v_pipe+m_l_pipe,'LineWidth',3)
grid on
set(gca,'xtick',0:25:666)
xlim([0,666]);
xlabel('Time (s)');
ylabel('Total Pipe Fluid Mass (kg)');

%% Define Custom Colors for Static Transient Plots

Gray = [.7 .7 .7];
length1 = 6;
length2 = 6;
red = [0.8500, 0.3250, 0.0980];
pink = [255, 192, 203]/255;
color1 = [linspace(red(1),pink(1),length1)', linspace(red(2),pink(2),...
    length1)', linspace(red(3),pink(3),length1)'];
darkblue = [0, 0.4470, 0.7410];
lightblue = [189, 252 255]/255;
color2 = [linspace(darkblue(1),lightblue(1),length2)',...
    linspace(darkblue(2),lightblue(2),length2)', linspace(darkblue(3),...
    lightblue(3),length2)'];

% SS colors
SSblue = [0, 113.9850, 188.9550];
SSred = [ 216.7500, 82.8750, 24.9900];
SSyellow = [0.9290, 0.6940, 0.1250];
SSpurple = [0.4940, 0.1840, 0.5560];

% line types
upline = '-';
dnlne = '-.';

%% Plot Data

% Pressure SS and Transients No video

figure('units','normalized','outerposition',[0 0 0.6 0.85]);
hold on
plot(PipeLength,P(600,:)/10^6,upline,'Color','k','LineWidth',3);
plot(PipeLength,P(605,:)/10^6,upline,'Color',color1(6,:), 'LineWidth',3);
plot(PipeLength,P(610,:)/10^6,upline,'Color',color1(5,:), 'LineWidth',3);
plot(PipeLength,P(615,:)/10^6,upline,'Color',color1(4,:), 'LineWidth',3);
plot(PipeLength,P(620,:)/10^6,upline,'Color',color1(3,:), 'LineWidth',3);
plot(PipeLength,P(625,:)/10^6,upline,'Color',color1(2,:), 'LineWidth',3);
plot(PipeLength,P(630,:)/10^6,upline,'Color',color1(1,:), 'LineWidth',3);
plot(PipeLength,P(635,:)/10^6,dnlne,'Color',color2(1,:), 'LineWidth',3);
plot(PipeLength,P(640,:)/10^6,dnlne,'Color',color2(2,:), 'LineWidth',3);
plot(PipeLength,P(645,:)/10^6,dnlne,'Color',color2(3,:), 'LineWidth',3);

```

```

plot(PipeLength,P(650,:)/10^6,dnline,'Color',color2(4,:), 'LineWidth',3);
plot(PipeLength,P(655,:)/10^6,dnline,'Color',color2(5,:), 'LineWidth',3);
plot(PipeLength,P(660,:)/10^6,dnline,'Color',color2(6,:), 'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',2.1:0.05:2.8);
grid on
xlim([0,NumCells*0.25]);
ylim([2.2,2.7]);
ylabel('Pressure (MPa)');
xlabel('Pipe Length (m)');
legend('SS','605 s','610 s','615 s','620 s','625 s','630 s','635 s',...
    '640 s','645 s','650 s','655 s','660 s','Pipe Bend','NumColumns',8);

```

%% Temperature SS and Transients No video

% Transient drop

```

figure('units','normalized','outerposition',[0 0 0.6 0.85]);
tiledlayout(2,2,'TileSpacing','compact');
nexttile
hold on
plot(PipeLength(1:3:end),tempgas(600,1:3:end),dnline,'LineWidth',3);
plot(PipeLength(3:4:end),templiq(600,3:4:end),dnline,'LineWidth',3);
plot(PipeLength(2:3:end),tempwalli(600,2:3:end),dnline,'LineWidth',3);
plot(PipeLength(1:5:end),tempwallo(600,1:5:end),dnline,'LineWidth',3);
plot(PipeLength(1:4:end),tempsat(600,1:4:end),'k*','MarkerSize',7);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25,509,'Steady-State:', 'HorizontalAlignment','left',...
    'fontweight','bold','FontSize',8.5);
text(13.25,508,'600 s and 660 s', 'HorizontalAlignment','left',...
    'fontweight','bold','FontSize',8.5);
leg = legend('SS Vapor','SS Liquid','SS Inner Wall','SS Outer Wall',...
    'SS Saturation','Pipe Bend','', 'NumColumns',3);
leg.Layout.Tile = 'South';

nexttile

```

```

hold on
plot(PipeLength,tempgas(605,:), 'LineWidth',3);
plot(PipeLength,templiq(605,:), 'LineWidth',3);
plot(PipeLength,tempwalli(605,:), 'LineWidth',3);
plot(PipeLength,tempwallo(605,:), 'LineWidth',3);
plot(PipeLength(1:3:end),tempsat(605,1:3:end), 'k.', 'MarkerSize',10);
xline(2, 'k', 'LineWidth',1);
xline(4, 'k', 'LineWidth',1);
xline(10, 'k', 'LineWidth',1);
xline(13, 'k', 'LineWidth',1);
xline(21, 'k', 'LineWidth',1);
set(gca, 'xtick', 1:2:23);
set(gca, 'ytick', 490:2:510);
grid on
xlim([0, NumCells*0.25]);
ylim([490, 510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25, 509, 'Time: 605 s', 'HorizontalAlignment', 'left', 'fontweight', ...
    'bold', 'FontSize', 8.5);

```

```

nexttile
hold on
plot(PipeLength,tempgas(610,:), 'LineWidth',3);
plot(PipeLength,templiq(610,:), 'LineWidth',3);
plot(PipeLength,tempwalli(610,:), 'LineWidth',3);
plot(PipeLength,tempwallo(610,:), 'LineWidth',3);
plot(PipeLength(1:3:end),tempsat(610,1:3:end), 'k.', 'MarkerSize',10);
xline(2, 'k', 'LineWidth',1);
xline(4, 'k', 'LineWidth',1);
xline(10, 'k', 'LineWidth',1);
xline(13, 'k', 'LineWidth',1);
xline(21, 'k', 'LineWidth',1);
set(gca, 'xtick', 1:2:23);
set(gca, 'ytick', 490:2:510);
grid on
xlim([0, NumCells*0.25]);
ylim([490, 510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25, 509, 'Time: 610 s', 'HorizontalAlignment', 'left', 'fontweight', ...
    'bold', 'FontSize', 8.5);

```

```

nexttile
hold on
plot(PipeLength,tempgas(615,:), 'LineWidth',3);
plot(PipeLength,templiq(615,:), 'LineWidth',3);
plot(PipeLength,tempwalli(615,:), 'LineWidth',3);
plot(PipeLength,tempwallo(615,:), 'LineWidth',3);
plot(PipeLength(1:3:end),tempsat(615,1:3:end), 'k.', 'MarkerSize',10);
xline(2, 'k', 'LineWidth',1);
xline(4, 'k', 'LineWidth',1);
xline(10, 'k', 'LineWidth',1);

```

```

xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
leg = legend('Vapor','Liquid','Inner Wall','Outer Wall',...
    'Saturation','Pipe Bend','','','NumColumns',3);
leg.Layout.Tile = 'South';
text(13.25,509,'Time: 615 s','HorizontalAlignment','left','fontweight',...
    'bold','FontSize',8.5);

figure('units','normalized','outerposition',[0 0 0.6 0.85]);
tiledlayout(2,2,'TileSpacing','compact');
nexttile
hold on
plot(PipeLength(1:3:end),tempgas(600,1:3:end),dline,'LineWidth',3);
plot(PipeLength(3:4:end),templiq(600,3:4:end),dline,'LineWidth',3);
plot(PipeLength(2:3:end),tempwalli(600,2:3:end),dline,'LineWidth',3);
plot(PipeLength(1:5:end),tempwallo(600,1:5:end),dline,'LineWidth',3);
plot(PipeLength(1:4:end),tempsat(600,1:4:end),'k*','MarkerSize',7);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25,509,'Steady-State:', 'HorizontalAlignment','left',...
    'fontweight','bold','FontSize',8.5);
text(13.25,508,'600 s and 660 s', 'HorizontalAlignment','left',...
    'fontweight','bold','FontSize',8.5);
leg = legend('SS Vapor','SS Liquid','SS Inner Wall','SS Outer Wall',...
    'SS Saturation','Pipe Bend','','','NumColumns',3);
leg.Layout.Tile = 'South';

nexttile
hold on
plot(PipeLength,tempgas(620,:), 'LineWidth',3);
plot(PipeLength,templiq(620,:), 'LineWidth',3);
plot(PipeLength,tempwalli(620,:), 'LineWidth',3);
plot(PipeLength,tempwallo(620,:), 'LineWidth',3);
plot(PipeLength(1:3:end),tempsat(620,1:3:end),'k.','MarkerSize',10);
xline(2,'k','LineWidth',1);

```

```

xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25,509,'Time: 620 s','HorizontalAlignment','left','fontweight',...
     'bold','FontSize',8.5);

```

```

nexttile
hold on
plot(PipeLength,tempgas(625,:), 'LineWidth',3);
plot(PipeLength,templiq(625,:), 'LineWidth',3);
plot(PipeLength,tempwalli(625,:), 'LineWidth',3);
plot(PipeLength,tempwallo(625,:), 'LineWidth',3);
plot(PipeLength(1:3:end),tempsat(625,1:3:end), 'k.', 'MarkerSize',10);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25,509,'Time: 625 s','HorizontalAlignment','left','fontweight',...
     'bold','FontSize',8.5);

```

```

nexttile
hold on
plot(PipeLength,tempgas(630,:), 'LineWidth',3);
plot(PipeLength,templiq(630,:), 'LineWidth',3);
plot(PipeLength,tempwalli(630,:), 'LineWidth',3);
plot(PipeLength,tempwallo(630,:), 'LineWidth',3);
plot(PipeLength(1:3:end),tempsat(630,1:3:end), 'k.', 'MarkerSize',10);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);

```

```

xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
leg = legend('Vapor','Liquid','Inner Wall','Outer Wall',...
    'Saturation','Pipe Bend','','','NumColumns',3);
leg.Layout.Tile = 'South';
text(13.25,509,'Time: 630 s','HorizontalAlignment','left','fontweight',...
    'bold','FontSize',8.5);

```

% Transient Recovery

```

figure('units','normalized','outerposition',[0 0 0.6 0.85]);
tiledlayout(2,2,'TileSpacing','compact');
nexttile
hold on
plot(PipeLength(1:3:end),tempgas(600,1:3:end),dline,'LineWidth',3);
plot(PipeLength(3:4:end),templiq(600,3:4:end),dline,'LineWidth',3);
plot(PipeLength(2:3:end),tempwalli(600,2:3:end),dline,'LineWidth',3);
plot(PipeLength(1:5:end),tempwallo(600,1:5:end),dline,'LineWidth',3);
plot(PipeLength(1:4:end),tempsat(600,1:4:end),'k*','MarkerSize',7);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25,509,'Steady-State:', 'HorizontalAlignment','left',...
    'fontweight','bold','FontSize',8.5);
text(13.25,508,'600 s and 660 s','HorizontalAlignment','left',...
    'fontweight','bold','FontSize',8.5);
leg = legend('SS Vapor','SS Liquid','SS Inner Wall','SS Outer Wall',...
    'SS Saturation','Pipe Bend','','','NumColumns',3);
leg.Layout.Tile = 'South';

nexttile
hold on
plot(PipeLength,tempgas(635,:), 'LineWidth',3);
plot(PipeLength,templiq(635,:), 'LineWidth',3);
plot(PipeLength,tempwalli(635,:), 'LineWidth',3);
plot(PipeLength,tempwallo(635,:), 'LineWidth',3);
plot(PipeLength(1:3:end),tempsat(635,1:3:end),'k.','MarkerSize',10);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);

```

```

set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25,509,'Time: 635 s','HorizontalAlignment','left','fontweight',...
     'bold','FontSize',8.5);

```

```

nexttile
hold on
plot(PipeLength,tempgas(640,:), 'LineWidth',3);
plot(PipeLength,templiq(640,:), 'LineWidth',3);
plot(PipeLength,tempwalli(640,:), 'LineWidth',3);
plot(PipeLength,tempwallo(640,:), 'LineWidth',3);
plot(PipeLength(1:3:end),tempsat(640,1:3:end), 'k.', 'MarkerSize',10);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25,509,'Time: 640 s','HorizontalAlignment','left','fontweight',...
     'bold','FontSize',8.5);

```

```

nexttile
hold on
plot(PipeLength,tempgas(645,:), 'LineWidth',3);
plot(PipeLength,templiq(645,:), 'LineWidth',3);
plot(PipeLength,tempwalli(645,:), 'LineWidth',3);
plot(PipeLength,tempwallo(645,:), 'LineWidth',3);
plot(PipeLength(1:3:end),tempsat(645,1:3:end), 'k.', 'MarkerSize',10);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
leg = legend('Vapor','Liquid','Inner Wall','Outer Wall',...
             'Saturation','Pipe Bend','','','NumColumns',3);
leg.Layout.Tile = 'South';

```

```

text(13.25,509,'Time: 645 s','HorizontalAlignment','left','fontweight',...
     'bold','FontSize',8.5);

figure('units','normalized','outerposition',[0 0 0.6 0.85]);
tiledlayout(2,2,'TileSpacing','compact');
nexttile
hold on
plot(PipeLength(1:3:end),tempgas(600,1:3:end),dline,'LineWidth',3);
plot(PipeLength(3:4:end),templiq(600,3:4:end),dline,'LineWidth',3);
plot(PipeLength(2:3:end),tempwalli(600,2:3:end),dline,'LineWidth',3);
plot(PipeLength(1:5:end),tempwallo(600,1:5:end),dline,'LineWidth',3);
plot(PipeLength(1:4:end),tempsat(600,1:4:end),'k*','MarkerSize',7);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25,509,'Steady-State:','HorizontalAlignment','left',...
     'fontweight','bold','FontSize',8.5);
text(13.25,508,'600 s and 660 s','HorizontalAlignment','left',...
     'fontweight','bold','FontSize',8.5);
leg = legend('SS Vapor','SS Liquid','SS Inner Wall','SS Outer Wall',...
             'SS Saturation','Pipe Bend','','','NumColumns',3);
leg.Layout.Tile = 'South';

nexttile
hold on
plot(PipeLength,tempgas(650,:), 'LineWidth',3);
plot(PipeLength,templiq(650,:), 'LineWidth',3);
plot(PipeLength,tempwalli(650,:), 'LineWidth',3);
plot(PipeLength,tempwallo(650,:), 'LineWidth',3);
plot(PipeLength(1:3:end),tempsat(650,1:3:end), 'k.', 'MarkerSize',10);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',1:2:23);
set(gca,'ytick',490:2:510);
grid on
xlim([0,NumCells*0.25]);
ylim([490,510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25,509,'Time: 650 s','HorizontalAlignment','left','fontweight',...
     'bold','FontSize',8.5);

```

```

nexttile
hold on
plot(PipeLength, tempgas(655,:), 'LineWidth', 3);
plot(PipeLength, tempqliq(655,:), 'LineWidth', 3);
plot(PipeLength, tempwalli(655,:), 'LineWidth', 3);
plot(PipeLength, tempwallo(655,:), 'LineWidth', 3);
plot(PipeLength(1:3:end), tempsat(655, 1:3:end), 'k.', 'MarkerSize', 10);
xline(2, 'k', 'LineWidth', 1);
xline(4, 'k', 'LineWidth', 1);
xline(10, 'k', 'LineWidth', 1);
xline(13, 'k', 'LineWidth', 1);
xline(21, 'k', 'LineWidth', 1);
set(gca, 'xtick', 1:2:23);
set(gca, 'ytick', 490:2:510);
grid on
xlim([0, NumCells*0.25]);
ylim([490, 510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
text(13.25, 509, 'Time: 655 s', 'HorizontalAlignment', 'left', 'fontweight', ...
    'bold', 'FontSize', 8.5);

nexttile
hold on
plot(PipeLength, tempgas(660,:), 'LineWidth', 3);
plot(PipeLength, tempqliq(660,:), 'LineWidth', 3);
plot(PipeLength, tempwalli(660,:), 'LineWidth', 3);
plot(PipeLength, tempwallo(660,:), 'LineWidth', 3);
plot(PipeLength(1:3:end), tempsat(660, 1:3:end), 'k.', 'MarkerSize', 10);
xline(2, 'k', 'LineWidth', 1);
xline(4, 'k', 'LineWidth', 1);
xline(10, 'k', 'LineWidth', 1);
xline(13, 'k', 'LineWidth', 1);
xline(21, 'k', 'LineWidth', 1);
set(gca, 'xtick', 1:2:23);
set(gca, 'ytick', 490:2:510);
grid on
xlim([0, NumCells*0.25]);
ylim([490, 510]);
xlabel('Pipe Length (m)');
ylabel('Temperature (K)');
leg = legend('Vapor', 'Liquid', 'Inner Wall', 'Outer Wall', ...
    'Saturation', 'Pipe Bend', '', '', 'NumColumns', 3);
leg.Layout.Tile = 'South';
text(13.25, 509, 'Time: 660 s', 'HorizontalAlignment', 'left', 'fontweight', ...
    'bold', 'FontSize', 8.5);

%% Mass Flow Rate Transients no video

figure('units', 'normalized', 'outerposition', [0 0 0.6 0.85]);

```

```

hold on
plot(PipeLength,mdotliq(600,:), 'Color','k','LineWidth',4);
plot(PipeLength,mdotliq(605,:),upline,'Color',color1(6,:), 'LineWidth',3);
plot(PipeLength,mdotliq(610,:),upline,'Color',color1(5,:), 'LineWidth',3);
plot(PipeLength,mdotliq(615,:),upline,'Color',color1(4,:), 'LineWidth',3);
plot(PipeLength,mdotliq(620,:),upline,'Color',color1(3,:), 'LineWidth',3);
plot(PipeLength,mdotliq(625,:),upline,'Color',color1(2,:), 'LineWidth',3);
plot(PipeLength,mdotliq(630,:),upline,'Color',color1(1,:), 'LineWidth',3);
plot(PipeLength,mdotliq(635,:),dnlne,'Color',color2(1,:), 'LineWidth',3);
plot(PipeLength,mdotliq(640,:),dnlne,'Color',color2(2,:), 'LineWidth',2);
plot(PipeLength,mdotliq(645,:),dnlne,'Color',color2(3,:), 'LineWidth',2);
plot(PipeLength,mdotliq(650,:),dnlne,'Color',color2(4,:), 'LineWidth',2);
plot(PipeLength,mdotliq(655,:),dnlne,'Color',color2(5,:), 'LineWidth',2);
plot(PipeLength,mdotliq(660,:),dnlne,'Color',color2(6,:), 'LineWidth',2);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
set(gca,'ytick',0.15:0.05:0.75);
grid on
xlim([0,NumCells*0.25]);
ylim([0.42,0.60]);
xlabel('Pipe Length (m)');
ylabel('Liquid Mass Flow Rate (kg/s)');
legend('SS','605 s','610 s','615 s','620 s','625 s','630 s','635 s',...
    '640 s','645 s','650 s','655 s','660 s','Pipe Bend','NumColumns',8);

```

```

figure('units','normalized','outerposition',[0 0 0.6 0.85]);
hold on
plot(PipeLength,mdotgas(600,:), 'Color','k','LineWidth',4);
plot(PipeLength,mdotgas(605,:),upline,'Color',color1(6,:), 'LineWidth',3);
plot(PipeLength,mdotgas(610,:),upline,'Color',color1(5,:), 'LineWidth',3);
plot(PipeLength,mdotgas(615,:),upline,'Color',color1(4,:), 'LineWidth',3);
plot(PipeLength,mdotgas(620,:),upline,'Color',color1(3,:), 'LineWidth',3);
plot(PipeLength,mdotgas(625,:),upline,'Color',color1(2,:), 'LineWidth',3);
plot(PipeLength,mdotgas(630,:),upline,'Color',color1(1,:), 'LineWidth',3);
plot(PipeLength,mdotgas(635,:),dnlne,'Color',color2(1,:), 'LineWidth',3);
plot(PipeLength,mdotgas(640,:),dnlne,'Color',color2(2,:), 'LineWidth',3);
plot(PipeLength,mdotgas(645,:),dnlne,'Color',color2(3,:), 'LineWidth',3);
plot(PipeLength,mdotgas(650,:),dnlne,'Color',color2(4,:), 'LineWidth',3);
plot(PipeLength,mdotgas(655,:),dnlne,'Color',color2(5,:), 'LineWidth',3);
plot(PipeLength,mdotgas(660,:),dnlne,'Color',color2(6,:), 'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
set(gca,'ytick',25.2:0.05:25.5);
grid on
xlim([0,NumCells*0.25]);

```

```

ylim([25.2,25.5]);
xlabel('Pipe Length (m)');
ylabel('Vapor Mass Flow Rate (kg/s)');
legend('SS','605 s','610 s','615 s','620 s','625 s','630 s','635 s',...
      '640 s','645 s','650 s','655 s','660 s','Pipe Bend','NumColumns',8);

```

%% Flow Quality

```

figure('units','normalized','outerposition',[0 0 0.6 0.85]);
hold on
upline = '-';
dnlne = '-.';
plot(PipeLength,FQ(600,:), 'Color','k','LineWidth',3);
plot(PipeLength,FQ(605,:),upline, 'Color',color1(6,:), 'LineWidth',3);
plot(PipeLength,FQ(610,:),upline, 'Color',color1(5,:), 'LineWidth',3);
plot(PipeLength,FQ(615,:),upline, 'Color',color1(4,:), 'LineWidth',3);
plot(PipeLength,FQ(620,:),upline, 'Color',color1(3,:), 'LineWidth',3);
plot(PipeLength,FQ(625,:),upline, 'Color',color1(2,:), 'LineWidth',3);
plot(PipeLength,FQ(630,:),upline, 'Color',color1(1,:), 'LineWidth',3);
plot(PipeLength,FQ(635,:),dnlne, 'Color',color2(1,:), 'LineWidth',3);
plot(PipeLength,FQ(640,:),dnlne, 'Color',color2(2,:), 'LineWidth',3);
plot(PipeLength,FQ(645,:),dnlne, 'Color',color2(3,:), 'LineWidth',3);
plot(PipeLength,FQ(650,:),dnlne, 'Color',color2(4,:), 'LineWidth',3);
plot(PipeLength,FQ(655,:),dnlne, 'Color',color2(5,:), 'LineWidth',3);
plot(PipeLength,FQ(660,:),dnlne, 'Color',color2(6,:), 'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
set(gca,'ytick',0.96:0.001:1);
grid on
xlim([0,NumCells*0.25]);
ylim([0.977,0.984]);
ylabel('Flow Quality');
legend('SS','605 s','610 s','615 s','620 s','625 s','630 s','635 s',...
      '640 s','645 s','650 s','655 s','660 s','Pipe Bend','NumColumns',8);

```

%% Vapor Mass Fraction

```

figure('units','normalized','outerposition',[0 0 0.6 0.85]);
hold on
plot(PipeLength,massfrac(605,:),upline, 'Color',color1(6,:), 'LineWidth',3);
plot(PipeLength,massfrac(610,:),upline, 'Color',color1(5,:), 'LineWidth',3);
plot(PipeLength,massfrac(615,:),upline, 'Color',color1(4,:), 'LineWidth',3);
plot(PipeLength,massfrac(620,:),upline, 'Color',color1(3,:), 'LineWidth',3);
plot(PipeLength,massfrac(625,:),upline, 'Color',color1(2,:), 'LineWidth',3);
plot(PipeLength,massfrac(630,:),upline, 'Color',color1(1,:), 'LineWidth',3);
plot(PipeLength,massfrac(635,:),dnlne, 'Color',color2(1,:), 'LineWidth',2);

```

```

plot(PipeLength, massfrac(640,:), dline, 'Color', color2(2,:), 'LineWidth', 2);
plot(PipeLength, massfrac(645,:), dline, 'Color', color2(3,:), 'LineWidth', 2);
plot(PipeLength, massfrac(650,:), dline, 'Color', color2(4,:), 'LineWidth', 2);
plot(PipeLength, massfrac(655,:), dline, 'Color', color2(5,:), 'LineWidth', 2);
plot(PipeLength, massfrac(660,:), dline, 'Color', color2(6,:), 'LineWidth', 2);
plot(PipeLength, massfrac(600,:), 'Color', 'k', 'LineWidth', 3);
xline(2, 'k', 'LineWidth', 1);
xline(4, 'k', 'LineWidth', 1);
xline(10, 'k', 'LineWidth', 1);
xline(13, 'k', 'LineWidth', 1);
xline(21, 'k', 'LineWidth', 1);
set(gca, 'xtick', 0:1:23);
set(gca, 'ytick', 0.955:0.001:1);
grid on
xlim([0, NumCells*0.25]);
ylim([0.9705, 0.985]);
ylabel('Vapor Mass Fraction');
xlabel('Pipe Length (m)');
legend('605 s', '610 s', '615 s', '620 s', '625 s', '630 s', '635 s', '640 s', ...
    '645 s', '650 s', '655 s', '660 s', 'SS', 'Pipe Bend', 'NumColumns', 8);

%% Heat Flux

figure('units', 'normalized', 'outerposition', [0 0 0.6 0.85]);
hold on
plot(PipeLength, -qwalli(600,:), 'Color', 'k', 'LineWidth', 3);
plot(PipeLength, -qwalli(605,:), upline, 'Color', color1(6,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(610,:), upline, 'Color', color1(5,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(615,:), upline, 'Color', color1(4,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(620,:), upline, 'Color', color1(3,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(625,:), upline, 'Color', color1(2,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(630,:), upline, 'Color', color1(1,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(635,:), dline, 'Color', color2(1,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(640,:), dline, 'Color', color2(2,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(645,:), dline, 'Color', color2(3,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(650,:), dline, 'Color', color2(4,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(655,:), dline, 'Color', color2(5,:), 'LineWidth', 3);
plot(PipeLength, -qwalli(660,:), dline, 'Color', color2(6,:), 'LineWidth', 3);
xline(2, 'k', 'LineWidth', 1);
xline(4, 'k', 'LineWidth', 1);
xline(10, 'k', 'LineWidth', 1);
xline(13, 'k', 'LineWidth', 1);
xline(21, 'k', 'LineWidth', 1);
set(gca, 'xtick', 0:1:23);
set(gca, 'ytick', -14000:2000:14000);
grid on
xlim([0, NumCells*0.25]);
ylim([-1.3E4, 1E4]);
ylabel('Inner Wall Heat Flux (W/m^2)');
legend('SS', '605 s', '610 s', '615 s', '620 s', '625 s', '630 s', '635 s', ...
    '640 s', '645 s', '650 s', '655 s', '660 s', 'Pipe Bend', 'NumColumns', 8);

```

```
%% Heat Tranfer Coefficient
```

```
figure('units','normalized','outerposition',[0 0 0.6 0.85]);
hold on
plot(PipeLength,h(600,:), 'Color','k','LineWidth',3);
plot(PipeLength,h(605,:),upline,'Color',color1(6,:), 'LineWidth',3);
plot(PipeLength,h(610,:),upline,'Color',color1(5,:), 'LineWidth',3);
plot(PipeLength,h(615,:),upline,'Color',color1(4,:), 'LineWidth',3);
plot(PipeLength,h(620,:),upline,'Color',color1(3,:), 'LineWidth',3);
plot(PipeLength,h(625,:),upline,'Color',color1(2,:), 'LineWidth',3);
plot(PipeLength,h(630,:),upline,'Color',color1(1,:), 'LineWidth',3);
plot(PipeLength,h(635,:),dnlne,'Color',color2(1,:), 'LineWidth',3);
plot(PipeLength,h(640,:),dnlne,'Color',color2(2,:), 'LineWidth',3);
plot(PipeLength,h(645,:),dnlne,'Color',color2(3,:), 'LineWidth',3);
plot(PipeLength,h(650,:),dnlne,'Color',color2(4,:), 'LineWidth',3);
plot(PipeLength,h(655,:),dnlne,'Color',color2(5,:), 'LineWidth',3);
plot(PipeLength,h(660,:),dnlne,'Color',color2(6,:), 'LineWidth',3);
xline(2,'k','LineWidth',1);
xline(4,'k','LineWidth',1);
xline(10,'k','LineWidth',1);
xline(13,'k','LineWidth',1);
xline(21,'k','LineWidth',1);
set(gca,'xtick',0:1:23);
grid on
xlim([0,NumCells*0.25]);
ylim([-6000,6000]);
ylabel('Heat Transfer Coefficient (W/m^2 K)');
xlabel('Pipe Length (m)');
legend('SS','605 s','610 s','615 s','620 s','625 s','630 s','635 s',...
    '640 s','645 s','650 s','655 s','660 s','Pipe Bend','NumColumns',8);
```

```
%% Total Pipe Mass over time
```

```
figure('units','normalized','outerposition',[0 0 0.6 0.85]);
hold on
T = tiledlayout(3,1,'TileSpacing','compact');

nexttile
hold on
xline(600,'k','LineWidth',1);
xline(660,'k','LineWidth',1);
plot(1:length(m_l_pipe),m_l_pipe,'LineWidth',3)
grid on
set(gca,'xtick',0:5:1260)
set(gca,'ytick',0:0.01:1)
xlim([550,710]);
ylim([0.27 0.32])
ylabel('Total Pipe Liquid Mass (kg)');
legend('Transient Initiation','Transient Termination')
```

```

nexttile
hold on
xline(600,'k','LineWidth',1);
xline(660,'k','LineWidth',1);
plot(1:length(m_v_pipe),m_v_pipe,'LineWidth',3)
grid on
set(gca,'ytick',0:0.5:16)
set(gca,'xtick',0:5:1260)
xlim([550,710]);
ylabel('Total Pipe Vapor Mass (kg)');

nexttile
hold on
xline(600,'k','LineWidth',1);
xline(660,'k','LineWidth',1);
plot(1:length(m_v_pipe),m_v_pipe+m_l_pipe,'LineWidth',3)
grid on
set(gca,'xtick',0:5:1260)
set(gca,'ytick',0:0.5:16)
xlim([550,710]);
xlabel('Time (s)');
ylabel('Total Pipe Fluid Mass (kg)');

```

APPENDIX I. MATLAB SCRIPT FOR LITERATURE COMPARISON WITH TRACE DATA

```

%% TRACE Interface Data Processor 03APR1 Guo Comparison

% ***** CHANGE THE FOLDER OF THE XTV FILE YOU WISH TO ANALYZE*****
% ***** CHANGE FROM '2022' TO 'garth' for personal/lab computer*****

% Tidy Up!
close all
clear
clc

%% Experimental Data Initializing

addpath ['C:\Users\2022\OneDrive - Naval Postgraduate School\Stonhill...'
        'Thesis\Open Literature Data\Guo Two Phase Steam\XSteam_Matlab_v2.6']

% Import data digitized from Guo, et. al. 2nd PDO Fig. 12

Temp_Fluid_i      = importdata('Guo_fluidtemp_a.csv');      % deg C
Temp_Inner_Wall_i = importdata('Guo_innerwalltemp_a.csv'); % deg C
Temp_Outer_Wall_i = importdata('Guo_outerwalltemp_a.csv'); % deg C
Pressure_Inlet_i  = importdata('Guo_inletpressure_a.csv');  % MPa
Pressure_Outlet_i = importdata('Guo_outletpressure_a.csv'); % MPa
Pressure_Drop_i   = importdata('Guo_pressuredrop_a.csv');   % kPa
Mass_Flux_Rate_i  = importdata('Guo_massflowrate_a.csv');   % kg/m2s
h_i              = importdata('Guo_Heat_Transfer_Coefficient_a.csv'); % W/m2s

% Parse data

% measurement data - digitized from Guo, et. al
Temp_Fluid      = Temp_Fluid_i(:,2)+273.15; % Fluid temp - K
Temp_Inner_Wall = Temp_Inner_Wall_i(:,2)+273.15; % Inner wall temp - K
Temp_Outer_Wall = Temp_Outer_Wall_i(:,2)+273.15; % Outer wall temp - K
Pressure_Inlet  = Pressure_Inlet_i(:,2); % Outlet pressure - MPa
Pressure_Outlet = Pressure_Outlet_i(:,2); % Inlet pressure - MPa
Pressure_Drop   = Pressure_Drop_i(:,2)/1000; % Pressure drop - MPa
Mass_Flux_Rate  = Mass_Flux_Rate_i(:,2); % Mass flux rate - kg/m2s
h              = h_i(:,2); % Fluid/wall heat transfer coefficient - W/m2K

% corresponding time scale data - digitized from Guo, et. al
Temp_Fluid_t      = Temp_Fluid_i(:,1); % Fluid temp - s
Temp_Inner_Wall_t = Temp_Inner_Wall_i(:,1); % Inner wall temp - s
Temp_Outer_Wall_t = Temp_Outer_Wall_i(:,1); % Outer wall temp - s
Pressure_Inlet_t   = Pressure_Inlet_i(:,1); % Outlet pressure - s
Pressure_Outlet_t  = Pressure_Outlet_i(:,1); % Inlet pressure - s
Pressure_Drop_t    = Pressure_Drop_i(:,1); % Pressure drop - s
Mass_Flux_Rate_t   = Mass_Flux_Rate_i(:,1); % Mass flux rate - s
h_t               = h_i(:,1); % Fluid/wall heat transfer coefficient - s

```

```

% Interpolate data sets to the same length

% set data set length
data_length = 300;

% largest time value
longest_time = max([max(Temp_Inner_Wall_t);max(Temp_Outer_Wall_t);...
    max(Temp_Fluid_t);max(Mass_Flux_Rate_t);max(Pressure_Inlet_t);...
    max(Pressure_Outlet_t);max(Pressure_Drop_t)]);

time_interp = ...
    linspace(0, longest_time, data_length); % s

Temp_Fluid_interp = ...
    interp1(Temp_Fluid_t, Temp_Fluid, time_interp); % K

Temp_Inner_Wall_interp = ...
    interp1(Temp_Inner_Wall_t, Temp_Inner_Wall, time_interp); % K

Temp_Outer_Wall_interp = ...
    interp1(Temp_Outer_Wall_t, Temp_Outer_Wall, time_interp); % K

Pressure_Inlet_interp = ...
    interp1(Pressure_Inlet_t, Pressure_Inlet, time_interp); % MPa

Pressure_Outlet_interp = ...
    interp1(Pressure_Outlet_t, Pressure_Outlet, time_interp); % MPa

Pressure_Drop_interp = ...
    interp1(Pressure_Drop_t, Pressure_Drop, time_interp); % MPa

Mass_Flux_Rate_interp = ...
    interp1(Mass_Flux_Rate_t, Mass_Flux_Rate, time_interp); % kg/m2s

Heat_Trans_Coeff_interp = ...
    interp1(h_t, h, time_interp); % W/m2K

%% Calculate inlet enthalpy and void fraction

% preallocate memory
Enthalpy_Inlet_TV = NaN(1,data_length);
X_Inlet_TV = NaN(1,data_length);

% Inlet steam table calculations - all enthalpies that are commented out
% will result in a quality calculation of 1, 0, or NaN. Quality was only
% succesfully estimated using enthalpy (calculated from fluid saturation
% temp) and pressure

for i=1:length(time_interp)

% Enthalpy_Inlet_PV(i) = ...
% XSteam('hV_p',Pressure_Inlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% % sat vapor enthalpy as a function of pressure - kJ/kg

```

```

% Enthalpy_Inlet_PL(i) = XSteam...
% ('hL_p',Pressure_Inlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% % sat liquid enthalpy as a function of pressure - kJ/kg

Enthalpy_Inlet_TV(i) = XSteam...
    ('hV_T',Temp_Fluid_interp(i)-273.15);
% % sat vapor enthalpy as a function of temp - kJ/kg

% Enthalpy_Inlet_TL(i) = XSteam...
% ('hL_T',Pressure_Inlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% % sat liquid enthalpy as a function of temp - kJ/kg

% X_Inlet_PV(i)      = XSteam...
% ('x_ph',Pressure_Inlet_interp(i).*10,Enthalpy_Inlet_PV(i));
% % vapor mass fraction from sat vapor pressure based enthalpy and pressure

% X_Inlet_PL(i)      = XSteam...
% ('x_ph',Pressure_Inlet_interp(i).*10,Enthalpy_Inlet_PL(i));
% % vapor mass fraction from sat vapor pressure based enthalpy and pressure

X_Inlet_TV(i)        = XSteam...
    ('x_ph',Pressure_Inlet_interp(i).*10,Enthalpy_Inlet_TV(i));
% % vapor mass fraction from sat vapor temp based enthalpy and pressure

% X_Inlet_TL(i)      = XSteam...
% ('x_ph',Pressure_Inlet_interp(i).*10,Enthalpy_Inlet_TL(i));
% % vapor mass fraction from sat vapor temp based enthalpy and pressure

end

% Calculate outlet enthalpy and void fraction

% preallocate memory
Enthalpy_Outlet_TV = NaN(1,data_length);
X_Outlet_TV        = NaN(1,data_length);

% Outlet steam table calculations - all enthalpies that are commented out
% will result in a quality calculation of 1, 0, or NaN. Quality was only
% succesfully estimated using enthalpy (calculated from fluid saturation
% temp) and pressure

for i=1:length(time_interp)
% Enthalpy_Outlet_PV(i) = XSteam...
% ('hV_p',Pressure_Outlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% % sat vapor enthalpy as a function of pressure - kJ/kg

% Enthalpy_Outlet_PL(i) = XSteam...
% ('hL_p',Pressure_Outlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% % sat liquid enthalpy as a function of pressure - kJ/kg

Enthalpy_Outlet_TV(i) = XSteam...
    ('hV_T',Temp_Fluid_interp(i)-273.15);
% % sat vapor enthalpy as a function of temp - kJ/kg

```

```

% Enthalpy_Outlet_TL(i) = XSteam...
% ('hL_T',Pressure_Outlet_interp(i).*10,Temp_Fluid_interp(i)-273.15);
% sat liquid enthalpy as a function of temp - kJ/kg
% X_Outlet_PV(i) = XSteam...
% ('x_ph',Pressure_Outlet_interp(i).*10,Enthalpy_Outlet_PV(i));
% % vapor mass fraction from sat vapor pressure based enthalpy and pressure

% X_Outlet_PL(i) = XSteam...
% ('x_ph',Pressure_Outlet_interp(i).*10,Enthalpy_Outlet_PL(i));
% % vapor mass fraction from sat vapor pressure based enthalpy and pressure

X_Outlet_TV(i) = XSteam...
('x_ph',Pressure_Outlet_interp(i).*10,Enthalpy_Outlet_TV(i));
% % vapor mass fraction from sat vapor temp based enthalpy and pressure

% X_Outlet_TL(i) = XSteam...
% ('x_ph',Pressure_Outlet_interp(i).*10,Enthalpy_Outlet_TL(i));
% % vapor mass fraction from sat vapor temp based enthalpy and pressure

end

%% Trace Initializing

% Add TRACE-MATLAB Interface to path
javaaddpath('C:\Users\2022\snap\lib\cafean.jar')
javaaddpath('C:\Users\2022\snap\lib\matlab_acs.jar')
javaaddpath('C:\Users\2022\snap\plugins\extdata.jar')
javaaddpath('C:\Users\2022\snap\plugins\TRACE.jar')
import com.cafean.matlab.*
TRACEINT = MatlabTraceIntf; % handle for MATLAB-TRACE interface

% Open XTV file and generate file ID - Change if needed
TRACEINT.openFile(['C:\Users\2022\OneDrive - Naval Postgraduate' ...
    ' School\Stonhill Thesis\TRACE Data\TRACE Runs\03APRnum3\Base_Job\'...
    'Base_Job.xtv']);

% Change from British units (default) to SI units.
TRACEINT.setUseSIunits( true );

% Number of timesteps from TRACE XTV file
NumTimeSteps = TRACEINT.getNumTimeSteps(0);

% Number of cells in simulation
NumCells = 92;

% Preallocate memory
[mdotgas,mdottot,rho_v,rho_l,P,qwalli,qwallo,templi,tempgas,tempsat,...
    tempwalli,tempwallo,VF] = deal(NaN(NumTimeSteps,NumCells*2));

% Load data

for i = 1:NumCells

```

```

if i < 10
    j = '0';
else
    j = '';
end

k = num2str(i);
m = i-1;

mdotgas(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('rvmf-10A',j,k)});
mdottot(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('rmvm-10A',j,k)});
rho_v(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('rovn-10A',j,k)});
rho_l(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('roln-10A',j,k)});
P(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('pn-10A',j,k)});
qwalli(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('qppi-10001A',j,k)});
qwallo(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('qppo-10001A',j,k)});
templiq(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tln-10A',j,k)});
tempgas(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tvn-10A',j,k)});
tempsat(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tssn-10A',j,k)});
tempwalli(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tsurfi-10001A',j,k)});
tempwallo(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('tsurfo-10001A',j,k)});
VF(:,i+m:i+1+m) = TRACEINT.getData(0,{strcat('alpn-10A',j,k)});

end

% delete timestep columns
mdotgas(:,1:2:end) = [];
mdottot(:,1:2:end) = [];
rho_v(:,1:2:end) = [];
rho_l(:,1:2:end) = [];
P(:,1:2:end) = [];
qwalli(:,1:2:end) = [];
qwallo(:,1:2:end) = [];
tempgas(:,1:2:end) = [];
templiq(:,1:2:end) = [];
tempsat(:,1:2:end) = [];
tempwalli(:,1:2:end) = [];
tempwallo(:,1:2:end) = [];
VF(:,1:2:end) = [];

% Calculated Values

% all change rates are with respect to pipe length, not time
Cells = 1:NumCells;
l = 6.448/92; % cell length - m
PipeLength = Cells*l; % Length of Pipe - m
r = 15/2000; % pipe inner radius - m
cell_area = pi*(r^2); % cell area - m^2
cell_vol = cell_area*l; % cell volume - m^3
FQ = mdotgas./mdottot; % cell flow quality
mdotliq = mdottot-mdotgas; % liquid mass flow rate - kg/m
diffFQ = diff(FQ');
FQ_drop_rate = diffFQ'; % cell flow quality change rate

```

```

V_v = VF.*cell_vol; % vapor volume - m^3
V_l = (1.-VF).*cell_vol; % liquid volume - m^3
m_v = V_v.*rho_v; % vapor mass - kg
m_l = V_l.*rho_l; % liquid mass - kg
massfrac = m_v./(m_v+m_l); % vapor mass fraction
diffmassfrac = diff(massfrac');
massfrac_drop_rate = diffmassfrac'; % vapor mass fraction change rate
diffP = diff(P');
P_drop_rate = diffP'; % pressure change rate - MPa/m
h = qwalli./(tempwalli-tempgas); % heat transfer coefficient - W/m^2K
m_v_pipe = sum(m_v,2); % total mass of vapor in pipe - kg
m_l_pipe = sum(m_l,2); % total mass of vapor in pipe - kg

%% Isolate single transient and prepare TRACE data for comparison

P_i = P(:,4)/(10^6);
P_o = P(:,91)/(10^6);
tempwalli_new = mean(tempwalli,2);
tempwallo_new = mean(tempwallo,2);
templiq_new = mean(templiq,2);
tempgas_new = mean(tempgas,2);
mflux = (mdotgas(:,4)+mdotliq(:,4))/cell_area; % mass flux - kg/m^2s
h_new = h(:,30);
massfrac_i = massfrac(:,11);
massfrac_o = massfrac(:,91);

% Plot Data
close all
t = (50:200);

% Pressure Comparison
figure
hold on
grid on
set(gca,'xtick',50:25:200);
xline(75,'k','LineWidth',2)
xline(179,'k','LineWidth',2)
xlim([50,200]);
ylim([1.6,3.4]);
plot(t,Pressure_Inlet_interp(50:200),'LineWidth',3);
plot(t,Pressure_Outlet_interp(50:200),'LineWidth',3);
plot(t,P_i(255:405),'LineWidth',3);
plot(t,P_o(255:405),'LineWidth',3);
legend('Transient Initiation','Transient Termination','Guo Inlet',...
'Guo Outlet','TRACE Inlet','TRACE Outlet');
xlabel('Time (s)')
ylabel('Pressure (MPa)')

% Fluid Temperature Comparison
figure
hold on

```

```

grid on
set(gca,'xtick',50:25:200);
set(gca,'ytick',450:20:1050);
xline(75,'k','LineWidth',2)
xline(179,'k','LineWidth',2)
xlim([50,200]);
ylim([460,560]);
plot(t,Temp_Fluid_interp(50:200),'LineWidth',3);
plot(t,temp_liq_new(255:405),'LineWidth',3);
plot(t,temp_gas_new(255:405),'-','LineWidth',3);
legend('Transient Initiation','Transient Termination','Guo Fluid',...
    'TRACE Liquid','TRACE Vapor');
xlabel('Time (s)')
ylabel('Temperature (K)')

% Wall Temperature Comparison
figure
hold on
grid on
set(gca,'xtick',50:25:200);
set(gca,'ytick',460:20:1000);
xline(75,'k','LineWidth',2)
xline(179,'k','LineWidth',2)
xlim([50,200]);
plot(t,Temp_Inner_Wall_interp(50:200),'LineWidth',3);
plot(t,Temp_Outer_Wall_interp(50:200),'LineWidth',3);
plot(t,temp_walli_new(255:405),'-','LineWidth',3);
plot(t,temp_wallo_new(255:405),'--','LineWidth',3);
legend('Transient Initiation','Transient Termination','Guo Inner Wall',...
    'Guo Outer Wall',...
    'TRACE Inner Wall','TRACE Outer Wall');
xlabel('Time (s)')
ylabel('Temperature (K)')

% Mass Flux Comparison
figure
hold on
grid on
set(gca,'xtick',50:25:200);
set(gca,'ytick',0:500:10000);
xline(75,'k','LineWidth',2)
xline(179,'k','LineWidth',2)
xlim([50,200]);
plot(t,Mass_Flux_Rate_interp(50:200),'LineWidth',3);
plot(t,mflux(255:405),'LineWidth',3);
legend('Transient Initiation','Transient Termination','Guo','TRACE',...
    'Location','northwest');
xlabel('Time (s)')
ylabel('Mass Flux (kg/m^2s)')

% Heat Transfer Coefficient Comparison

```

```

figure
hold on
grid on
set(gca,'xtick',50:25:200);
set(gca,'ytick',0:2000:30000);
xline(75,'k','LineWidth',2)
xline(179,'k','LineWidth',2)
xlim([50,200]);
plot(t,Heat_Trans_Coeff_interp(50:200),'LineWidth',3);
plot(t,h_new(255:405),'LineWidth',3);
legend('Transient Initiation','Transient Termination','Guo','TRACE',...
'Location','northwest');
xlabel('Time (s)')
ylabel('Heat Transfer Coefficient (W/m^2K)')

% Vapor Mass Fraction Comparison
figure
hold on
grid on
xline(75,'k','LineWidth',2)
xline(179,'k','LineWidth',2)
xlabel('time (s)')
set(gca,'xtick',50:25:200);
plot(t,X_Inlet_TV(50:200),'LineWidth',3);
plot(t,X_Outlet_TV(50:200),'LineWidth',3);
plot(t,massfrac_i(255:405),'LineWidth',3);
plot(t,massfrac_o(255:405),'-','LineWidth',3);
legend('Transient Initiation','Transient Termination','Guo Inlet',...
'Guo Outlet','TRACE Inlet','TRACE Outlet','Location','southwest')
xlabel('Time (s)')
ylabel('Vapor Mass Fraction')

```

APPENDIX J. TRACE INPUTS FOR STEADY STATE MODEL

TRACE Model Report

Title: TRACE Steady-State Model
Date: Tue May 02 22:47:59 PDT 2023
Version Info: TRACE V 5.0 Patch 5
SNAP Version 4.1.0
TRACE Plug-in Version 4.5.2
Java Version 11.0.12

Component		Count
Hydraulic Components	Breaks	1
	Fills	1
	Pipes	1
	Total:	3
Control Systems	Signal Variables	1
	Total:	1

Hydraulic Statistics

Total Fluid Volume 1.1290 m³
hydro. system 1 1.1290 m³

Total Fluid Length 23.0 m
hydro. system 1 23.0 m

Overall Elevation Change 4.9080 m

Hydraulic System Statistics

Hydraulic System	Minimum Location	Minimum Elevation (m)	Maximum Location	Maximum Elevation (m)	Elevation Change (m)
Hydro. System 1	Pipe 10 Cell 85	0.0	Pipe 10 Cell 1	4.9080	4.9080

Model Validation Tests

10 tests performed, 10 passes, 0 failures

Loop Check

Test Description: Verifies that hydraulic components in a loop have a net elevation change of 0.0.

Error Tolerance: 0.001m

Test Result: Succeeded with no Issues.

Noncondensable Partial Pressure Test

Test Description: When the NOAIR namelist variable is set to one, every PA value must be set to 0.0. This

test examines every component in the model to determine if it has a PA value > 0.0 .

Test Result: Succeeded with no Issues.

Junction Flow Area Test

Test Description: Verifies that every junction connected to two cells has a flow area that does not exceed $1.1 \times$ the average flow area of the adjacent cells.

Test Result: Succeeded with no Issues.

Abrupt Area Change Test

Test Description: Verifies that adjacent cells flow areas are within a fraction of each other. The junction between them may also be included in this check.

Include Edges between Adjacent Volumes: false

Flow Area Ratio: 2.0

Test Result Succeeded with no Issues.

Edge Hydro-Diameter Test

Test Description: Compares the hydraulic diameter specified for edges inside 1D hydraulic components with the calculated hydraulic diameter of the volumes on either side. An error is reported if the edge is not within the tolerance of one of the calculated diameters.

Tolerance (Calculated Diameter vs Actual): 1.02

Test Result: Succeeded with no Issues.

PUMPFRIQC Consistency Test

Test Description: Verifies that power provided by enabling the PUMPFRIQC namelist option does not conflict with power added using a FLPOW component.

Test Result: Succeeded with no Issues.

3D Hydraulic Connection Test

Test Description: This test verifies the consistency of Hydraulic connections to the vessel. The component flow direction is compared target vessel cell face to ensure a consistent flow. This also verifies that the vessel cells have a non-zero volume fraction.

Tolerance(Connection GRAV vs Expected GRAV): $1.0E-4$

Test Result: Succeeded with no Issues.

Adjacent Cell Volume Consistency Test

Test Description: Verifies adjacent cell volumes do not exceed a user specified ratio.

Include Crossflows: true

Include 3D Volumes: true

Volume Ratio: 10.0

Test Result: Succeeded with no Issues.

Spawned Component Number Test

Test Description: Spawned Component Number Test

Single Kinetics Model Test

Test Description: Verifies that only a single power component will use the kinetics model.

Test Result: Succeeded with no Issues.

Model Options

Name	Value	Refs
Title Cards	[below]	-
Version	V 5.0 Patch 5	-
Namelist Option	[1] INOPT Data After Title Cards	-
Fluids	Water	-
Transient Calculation	[0] Steady State	-
Steady State Mode	[1] GSS Calculation	-
Flow Parameter	[0] Liquid and Vapor Mass Flows	-
Pressure Input Option	[0] Specify Pressures	-
Water Packing	On	-
Pressure Convergence	1.0E-4	-
Steady-State Convergence	1.0E-4	-
Dispersed Fields	[0] Droplet Fields, [0] Bubble Fields	-
Pressure Iterations	10	-
Steady-State Iterations	100	-
Solute Tracking	[0] Off	-
Namelist Variables	3 Active Entries [below]	-
User Defined Units	None	-
Timestep Data	[1] Timesteps [below]	-
Trip Initiated Timestep Data	[0] Timesteps	-
Mixed Numerics	0 of 1 Enabled	-
System Gas/Liquids	Species/Mixed NC Disabled	-
Initial Condition Sets	No condition sets defined	-
Sensitivity Coefficients	[0] coefficients defined	-

Title Card

Timestep Data

Initial Timestep Size: -1.0(s)

End Time (s)	Minimum Size (s)	Maximum Size (s)	Heat vs Fluid Size Ratio	Max Conv. Power Diff (W)	Long Edit Interval (s)	Graphics Interval (s)	Restart Interval (s)	Short Edit Interval (s)
200.0	1.0E-6	0.01	10.0	1.0E20	100.0	1.0	100.0	1.0

Namelist Variables

Name	Value	Refs
Show Inactive Entries	true	-
DTSTRT - Initial timestep size	-1.0 s	-
IKFAC - Friction Option	[1] K-Factors	-
USESJC - Use Single Jun Components	[3] SJs with Angle Input	-

Model Options Code Input

```

free format
*m: SNAP:Symbolic Nuclear Analysis Package, Version 4.1.0, July 28, 2022
*m: PLUGIN:TRACE Version 4.5.2
*m: CODE:TRACE V 5.0 Patch 5
*m: DATE:5/2/23
*
*
*****
* main data *
*****
*
*      numtcr      ieos      inopt      nmat      id2o
*      0          0          1          0          0
*
*
*****
* namelist data *
*****
*
&inopts
dtstrt=-1.0,
ikfac=1,
usesjc=3,
fluids='H2O'
&end
*
*****
* Model Flags *
*****
*
*      dstep      timet
*      0          0.0
*
*      stdyst      transi      ncomp      njun      ipak
*      1          0          3          2          1
*
*      epso      epss
*      1.0E-4    1.0E-4
*
*      oitmax      sitmax      isolut      ncontr      nccfl
*      10         100         0          0          0
*
*      ntsv      ntcb      ntcf      ntrp      ntcp
*      1         0         0          0          0
*
*
*****
* component-number data *
*****
*
* Component input order (IORDER)
*-- type ---- num ----- name ----- + jun1 jun2 jun3
* PIPE * 10 s * + 20 10

```

```

* FILL * 20 s * inlet + 20
* BREAK * 30 e * outlet + 10
*
*****
* Timestep Data *
*****
* dtmin dtmax tend rtwfp
* 1.0E-6 0.01 200.0 10.0
* edint gfint dmpint sedint
* 100.0 1.0 100.0 1.0
*
* endflag
* -1.0

```

Hydraulic Components

Component Summary Table

Component	Volume (m ³)	Length (m)	Elevation Change (m)
Break 30 (outlet) [p. 155]	0.0	0.0	0.0
Fill 20 (inlet) [p. 155]	0.0	0.0	0.0
Pipe 10 [p. 156]	1.1290	23.0	-3.0330

Break 30 (outlet)

Name	Value	Refs
Break Type	[0] No Tables	-
Temperature Table Option	[0] Enter liquid/gas temp	-
Fluid State Option	[0] Last Interp State Held Const.	-
Length	1.0 m	-
Volume	1.0 m ³	-
Initial Gas Volume Fraction	0.997223	-
Initial Mixture Temperature	505.89 K	-
Initial Pressure	2.639E6 Pa	-
Initial Noncondensable PP	0.0 Pa	-
Adjacent Pressure Flag	false	-
Max Pressure Change Rate	1.0E20 Pa/s	-

Break 30 (outlet) Code Input

```

*
***** type      num      userid      component name
break      30      0      outlet
* jun1      ibty      isat      ioff      adjpress
  10      0      0      0      0
* dxin      volin      alpin      tin      pin
  1.0      1.0      0.997223      505.89      2.639E6
* pain      concin      rbmx      poff      belv
  0.0      0.0      1.0E20      0.0      0.0

```

Fill 20 (inlet)

Name	Value	Refs
Fill Type	[1] Constant Velocity	-
Length	1.0 m	-
Volume	1.0 m ³	-
Initial Gas Volume Fraction	0.997223	-
Initial Liquid Temperature	505.89 K	-
Initial Vapor Temperature	505.89 K	-
Initial Pressure	2.639E6 Pa	-
Initial Noncondensable PP	0.0 Pa	-
Liquid Vel	40.0 m/s	-
Vapor Vel	40.0 m/s	-
Max Acceleration	1.0E20 m/s ²	-
Rate Factor Table	None	-

Contan Coupling

Name	Value	Refs
Couple to CONTAN	false	-

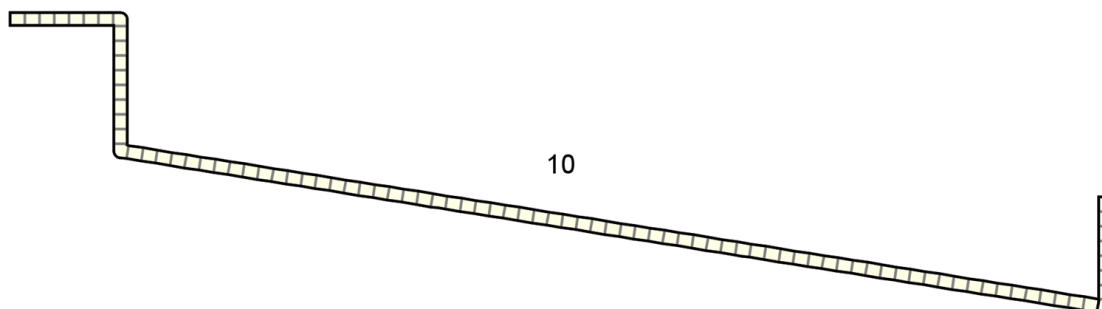
Fill 20 (inlet) Code Input

```

*
*****  type      num      userid      component name
fill      20        0          inlet
*
  jun1     ifty     ioff
  20       1        0
*
  twtold    rfmX     concin    felv
  0.0      1.0E20    0.0       0.0
*
  dxin     volin    alpin     vlin     tlin
  1.0      1.0     0.997223   40.0    505.89
*
  pin      pain     flowin    vvin     tvin
  2.639E6  0.0      0.0      40.0    505.89

```

Pipe 10



Name	Value	Refs
Component Geometry	Cells: 92, Junctions: 93 - Valid Values [below]	-
Initial Conditions	Valid Values [below]	-
Friction	Valid Values [below]	-
Fluid Power Options	Not Modeled	-
Wall Roughness	1.0E-4 m	-
Pipe Type	[0] No Accumulator	-
Number of Pipes	1	-
Enable Mixed Numerics	false	-

Hydraulic Connections

From Component	From Location	To Component	To Location
Pipe 10	Edge 1 inlet	Fill 20 (inlet) [p. 155]	Cell 1 inlet
Pipe 10	Edge 93 outlet	Break 30 (outlet) [p. 155]	Cell 1 inlet

Geometry

Pipe 10 Cell Geometry Data

Cell	Length (m)	Area (m ²)	Volume (m ³)	DZ (m)
1-7	0.25	0.049087385	0.012271846	0.0
8	0.25	0.049087385	0.012271846	-0.125
9-16	0.25	0.049087385	0.012271846	-0.24999999
17	0.25	0.049087385	0.012271846	-0.14454447
18-84	0.25	0.049087385	0.012271846	-0.039088953
85	0.25	0.049087385	0.012271846	0.10545552
86	0.25	0.049087385	0.012271846	0.25
87-92	0.25	0.049087385	0.012271846	0.24999999
Total	23.0	4.5160394	1.1290099	-3.0330488

Pipe 10 Edge Geometry Data

Edge	Flow Area (m ²)	Hydraulic Diameter (m)	CCFL Model
1-93	0.049087385	0.25	None

Pipe 10 Edge Orientation Data

Edge	Edge Angle (rad)	GRAV sin(Θ)
1-8	0.0	0.0
9-17	-1.571	-0.99999998
18-39	-0.157	-0.15635581
40-60	-0.157	-0.15635581

Edge	Edge Angle (rad)	GRAV sin(Θ)
61-85	-0.157	-0.15635581
86	1.571	1.0
87-93	1.571	0.99999998

Cell Initial Conditions

Cell	Pressure (Pa)	Liq. Temp (K)	Gas Temp (K)	Gas Frac. (-)	NC Pressure (Pa)
1-92	2.639E6	505.89	505.89	1.0	0.0

Edge Initial Conditions

Edge	Liquid Velocity (m/s)	Vapor Velocity (m/s)
1-93	0.0	0.0

Friction Data

Edge	Friction Correlation	Kfactors
1-8	0.0	Flow Factor + FRIC
9	0.2735	Flow Factor + FRIC
10-16	0.0	Flow Factor + FRIC
17	0.2594	Flow Factor + FRIC
18-40	0.0	Flow Factor + FRIC
41	0.2594	Flow Factor + FRIC
42-52	0.0	Flow Factor + FRIC
53	0.2594	Flow Factor + FRIC
54-84	0.0	Flow Factor + FRIC
85	0.2594	Flow Factor + FRIC
86-93	0.0	Flow Factor + FRIC

Pipe 10 Code Input

```

*
*****  type      num      userid      component name
pipe      10      0      unnamed
*  ncells      nodes      jun1      jun2      epsw
   92      0      20      10      1.0E-4
*  nsides
   0
*  ichf      iconc      pipetype      ipow      npipes
   1      0      0      0      1
*  radin      th      houtl      houtv      toutl
   0.0      0.0      0.0      0.0      0.0
*  toutv      pwin      pwoff      rpwmx      pwsc1
   0.0      0.0      0.0      0.0      0.0
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s

```

[illegible]

```

* fa * 0.049087385 0.049087385 0.049087385 0.049087385 s
* fa * 0.049087385 e
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.2735 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.2594 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.2594 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.2594 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.2594 0.0 0.0 0.0 s
* kfacs * 0.0 0.0 0.0 0.0 s
* kfacs * 0.0 e
* grav * 0.0 0.0 0.0 0.0 s
* grav * 0.0 0.0 0.0 0.0 s
* grav * -0.99999998 -0.99999998 -0.99999998 -0.99999998 s
* grav * -0.99999998 -0.99999998 -0.99999998 -0.99999998 s
* grav * -0.99999998 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 1.0 0.99999998 0.99999998 s
* grav * 0.99999998 0.99999998 0.99999998 0.99999998 s
* grav * 0.99999998 e
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s

```

* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25	0.25	0.25	0.25 s
* hd *	0.25 e			
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1 e			
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 e
* vl *	0.0	0.0	0.0	0.0 s
* vl *	0.0	0.0	0.0	0.0 s
* vl *	0.0	0.0	0.0	0.0 s

162

[illegible]

```

* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 e

```

Control Systems

Control Systems Summary Table

Component	Type
Problem Time 1 [p. 164]	Signal Variable

Problem Time 1

Name	Value	Refs
Parameter Type	[0] Problem Time	-
Time Option	[0] Current Timestep	-

Control Systems Code Input

```

*
*      idsv      isvn      ilcn      icn1      icn2
*      1         0         0         0         0

```

APPENDIX K. TRACE INPUTS FOR 98% INLET QUALITY PRESSURE TRANSIENT MODEL

TRACE Model Report

Title: TRACE Pressure Transient Model
Date: Tue May 02 22:53:21 PDT 2023
Version Info: TRACE V 5.0 Patch 5
SNAP Version 4.1.0
TRACE Plug-in Version 4.5.2
Java Version 11.0.12

Component		Count
Hydraulic Components	Breaks	1
	Fills	1
	Pipes	1
	Total:	3
Control Systems	Control Blocks	4
	Signal Variables	1
	Total:	5

Hydraulic Statistics

Total Fluid Volume 1.1290 m³
hydro. system 1 1.1290 m³

Total Fluid Length 23.0 m
hydro. system 1 23.0 m

Overall Elevation Change 4.9080 m

Hydraulic System Statistics

Hydraulic System	Minimum Location	Minimum Elevation (m)	Maximum Location	Maximum Elevation (m)	Elevation Change (m)
Hydro. System 1	Pipe 10 Cell 85	0.0	Pipe 10 Cell 1	4.9080	4.9080

Model Validation Tests

- 10 tests performed, 10 passes, 0 failures

Loop Check

- Test Description: Verifies that hydraulic components in a loop have a net elevation change of 0.0.
- Error Tolerance: 0.001m
- Test Result: Succeeded with no Issues.

Noncondensable Partial Pressure Test

- Test Description: When the NOAIR namelist variable is set to one, every PA value must be set to 0.0. This test examines every component in the model to determine if it has a PA value > 0.0.
- Test Result: Succeeded with no Issues.

Junction Flow Area Test

- Test Description: Verifies that every junction connected to two cells has a flow area that does not exceed 1.1 * the average flow area of the adjacent cells.
- Test Result: Succeeded with no Issues.

Abrupt Area Change Test

- Test Description: Verifies that adjacent cells flow areas are within a fraction of each other. The junction between them may also be included in this check.
- Include Edges between Adjacent Volumes: false
Flow Area Ratio: 2.0
- Test Result Succeeded with no Issues.

Edge Hydro-Diameter Test

- Test Description: Compares the hydraulic diameter specified for edges inside 1D hydraulic components with the calculated hydraulic diameter of the volumes on either side. An error is reported if the edge is not within the tolerance of one of the calculated diameters.
- Tolerance (Calculated Diameter vs Actual): 1.02
- Test Result: Succeeded with no Issues.

PUMPFRICTQ Consistency Test

- Test Description: Verifies that power provided by enabling the PUMPFRICTQ namelist option does not conflict with power added using a FLPOW component.
- Test Result: Succeeded with no Issues.

3D Hydraulic Connection Test

- Test Description: This test verifies the consistency of Hydraulic connections to the vessel. The component flow direction is compared target vessel cell face to ensure a consistent flow. This also verifies that the vessel cells have a non-zero volume fraction.
- Tolerance(Connection GRAV vs Expected GRAV): 1.0E-4
- Test Result: Succeeded with no Issues.

Adjacent Cell Volume Consistency Test

- Test Description: Verifies adjacent cell volumes do not exceed a user specified ratio.
- Include Crossflows: true
Include 3D Volumes: true
Volume Ratio: 10.0
- Test Result: Succeeded with no Issues.

Spawned Component Number Test

- Test Description: Spawned Component Number Test

Single Kinetics Model Test

- Test Description: Verifies that only a single power component will use the kinetics model.
- Test Result: Succeeded with no Issues.

Model Options

Name	Value	Refs
Title Cards	[above]	-
Version	V 5.0 Patch 5	-
Namelist Option	[1] INOPT Data After Title Cards	-
Fluids	Water	-
Transient Calculation	[1] Transient	-
Flow Parameter	[0] Liquid and Vapor Mass Flows	-
Pressure Input Option	[0] Specify Pressures	-
Water Packing	On	-
Pressure Convergence	1.0E-4	-
Steady-State Convergence	1.0E-4	-
Dispersed Fields	[0] Droplet Fields, [0] Bubble Fields	-
Pressure Iterations	10	-
Steady-State Iterations	10	-
Solute Tracking	[0] Off	-
Namelist Variables	3 Active Entries [above]	-
User Defined Units	None	-
Timestep Data	[1] Timesteps [above]	-
Trip Initiated Timestep Data	[0] Timesteps	-
Mixed Numerics	0 of 1 Enabled	-
System Gas/Liquids	Species/Mixed NC Disabled	-
Initial Condition Sets	No condition sets defined	-
Sensitivity Coefficients	[0] coefficients defined	-

Title Card

Timestep Data

- Initial Timestep Size: -1.0(s)

End Time (s)	Minimum Size (s)	Maximum Size (s)	Heat vs Fluid Size Ratio	Max Conv. Power Diff (W)	Long Edit Interval (s)	Graphics Interval (s)	Restart Interval (s)	Short Edit Interval (s)
666.0	1.0E-6	0.01	10.0	1.0E20	100.0	1.0	100.0	1.0

Namelist Variables

Name	Value	Refs
Show Inactive Entries	true	-
DTSTRT - Initial timestep size	-1.0 s	-
IKFAC - Friction Option	[1] K-Factors	-
USESJC - Use Single Jun Components	[3] SJs with Angle Input	-

Model Options Code Input

```

free format
*m: SNAP:Symbolic Nuclear Analysis Package, Version 4.1.0, July 28, 2022
*m: PLUGIN:TRACE Version 4.5.2
*m: CODE:TRACE V 5.0 Patch 5
*m: DATE:5/2/23
*
*
*****
* main data *
*****
*
*      numtcr      ieos      inopt      nmat      id2o
*      0          0          1          0          0
*
*
*****
* namelist data *
*****
*
&inopts
dtstrt=-1.0,
ikfac=1,
usesjc=3,
fluids='H2O'
&end
*
*****
* Model Flags *
*****
*
*      dstep      timet
*      0          0.0
*
*      stdyst      transi      ncomp      njun      ipak
*      0          1          4          2          1
*
*      epso      epss
*      1.0E-4    1.0E-4
*
*      oitmax      sitmax      isolut      ncontr      nccfl
*      10         10         0          0          0
*
*      ntsv      ntcb      ntcf      ntrp      ntcp
*      1         4         0          0          0
*
*
*****
* component-number data *
*****
*
* Component input order (IORDER)
*-- type ---- num ----- name ----- + jun1 jun2 jun3
* PIPE * 10 s * + 20 10

```

```

* HTSTR * 10001 s * Pipewall of Pipe 10 +
* FILL * 20 s * inlet + 20
* BREAK * 30 e * outlet + 10
*
*****
* Timestep Data *
*****
* dtmin dtmax tend rtwfp
* 1.0E-6 0.01 666.0 10.0
* edint gfind dmpint sedint
* 100.0 1.0 100.0 1.0
*
* endflag
* -1.0

```

Hydraulic Components

Component Summary Table

Component	Volume (m ³)	Length (m)	Elevation Change (m)
Break 30 (outlet) [p. 155]	0.0	0.0	0.0
Fill 20 (inlet) [p. 155]	0.0	0.0	0.0
Pipe 10 [p. 156]	1.1290	23.0	-3.0330

Break 30 (outlet)

Name	Value	Refs
Break Type	[6] Generalized Break	-
Temperature Table Option	[0] Enter liquid/gas temp	-
Fluid State Option	[0] Last Interp State Held Const.	-
Length	1.0 m	-
Volume	1.0 m ³	-
Initial Gas Volume Fraction	0.999855	-
Initial Mixture Temperature	505.89 K	-
Initial Pressure	2.639E6 Pa	-
Initial Noncondensable PP	0.0 Pa	-
Adjacent Pressure Flag	false	-

Scale Factors

Name	Value	Refs
Pressure Scale	1.0	-
Liquid Temperature Scale	1.0	-
Noncondensable Scale	1.0	-

State Controllers

Name	Value	Refs
Pressure Controller	Sum -3 [p. 187]	-

Name	Value	Refs
Liquid Temperature Controller	-Not Specified-	-
Vapor Temperature Controller	-Not Specified-	-
Vapor Volume Controller	-Not Specified-	-
Noncondensable Controller	-Not Specified-	-

Break 30 (outlet) Code Input

```

*
***** type      num      userid      component name
break      30      0      outlet
* jun1      ibty      isat      ioff      adjpress
  10      6      0      0      0
* ibtr      ibsv      nbtb      nbsv      nbrf
  0      0      0      0      0
* dxin      volin      alpin      tin      pin
  1.0      1.0      0.999855      505.89      2.639E6
* pain      concin      rbmx      poff      belv
  0.0      0.0      1.0E20      0.0      0.0
* ibpsv      ibtlsv      ibtvsv      ibasv      ibpasv
  -3      0      0      0      0
* ibcnsv
  0

```

Fill 20 (inlet)

Name	Value	Refs
Fill Type	[1] Constant Velocity	-
Length	1.0 m	-
Volume	1.0 m ³	-
Initial Gas Volume Fraction	0.999855	-
Initial Liquid Temperature	505.89 K	-
Initial Vapor Temperature	505.89 K	-
Initial Pressure	2.639E6 Pa	-
Initial Noncondensable PP	0.0 Pa	-
Liquid Vel	40.0 m/s	-
Vapor Vel	40.0 m/s	-
Max Acceleration	1.0E20 m/s ²	-
Rate Factor Table	None	-

Contan Coupling

Name	Value	Refs
Couple to CONTAN	false	-

Fill 20 (inlet) Code Input

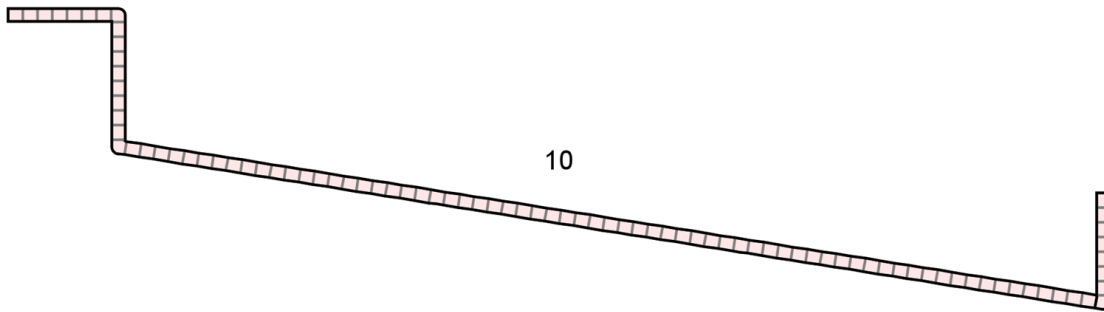
```

*
***** type      num      userid      component name
fill      20      0      inlet
* jun1      ifty      ioff

```

	20	1	0		
*	twtdold	rfmx	concin	felv	
	0.0	1.0E20	0.0	0.0	
*	dxin	volin	alpin	vlin	tlin
	1.0	1.0	0.999855	40.0	505.89
*	pin	pain	flowin	vvin	tvin
	2.639E6	0.0	0.0	40.0	505.89

Pipe 10



Name	Value	Refs
Component Geometry	Cells: 92, Junctions: 93 - Valid Values [above]	-
Initial Conditions	Valid Values [above]	-
Friction	Valid Values [above]	-
Fluid Power Options	Not Modeled	-
Wall Roughness	1.0E-4 m	-
Pipe Type	[0] No Accumulator	-
Number of Pipes	1	-
Enable Mixed Numerics	false	-

Hydraulic Connections

From Component	From Location	To Component	To Location
Pipe 10	Edge 1 inlet	Fill 20 (inlet) [p. 155]	Cell 1 inlet
Pipe 10	Edge 93 outlet	Break 30 (outlet) [p. 155]	Cell 1 inlet

Geometry

Pipe 10 Cell Geometry Data

Cell	Length (m)	Area (m ²)	Volume (m ³)	DZ (m)
1-7	0.25	0.049087385	0.012271846	0.0
8	0.25	0.049087385	0.012271846	-0.125
9-16	0.25	0.049087385	0.012271846	-0.24999999
17	0.25	0.049087385	0.012271846	-0.14454447
18-84	0.25	0.049087385	0.012271846	-0.039088953

Cell	Length (m)	Area (m ²)	Volume (m ³)	DZ (m)
85	0.25	0.049087385	0.012271846	0.10545552
86	0.25	0.049087385	0.012271846	0.25
87-92	0.25	0.049087385	0.012271846	0.24999999
Total	23.0	4.5160394	1.1290099	-3.0330488

Pipe 10 Edge Geometry Data

Edge	Flow Area (m ²)	Hydraulic Diameter (m)	CCFL Model
1-93	0.049087385	0.25	None

Pipe 10 Edge Orientation Data

Edge	Edge Angle (rad)	GRAV sin(Θ)
1-8	0.0	0.0
9-17	-1.571	-0.99999998
18-39	-0.157	-0.15635581
40-60	-0.157	-0.15635581
61-85	-0.157	-0.15635581
86	1.571	1.0
87-93	1.571	0.99999998

Cell Initial Conditions

Cell	Pressure (Pa)	Liq. Temp (K)	Gas Temp (K)	Gas Frac. (-)	NC Pressure (Pa)
1-92	2.639E6	505.89	505.89	1.0	0.0

Edge Initial Conditions

Edge	Liquid Velocity (m/s)	Vapor Velocity (m/s)
1-93	0.0	0.0

Friction Data

Edge	Friction Correlation	Kfactors
1-9	0.0	Flow Factor + FRIC
10	0.2552	Flow Factor + FRIC
11-17	0.0	Flow Factor + FRIC
18	0.24202	Flow Factor + FRIC
19-41	0.0	Flow Factor + FRIC
42	0.24202	Flow Factor + FRIC
43-53	0.0	Flow Factor + FRIC
54	0.24202	Flow Factor + FRIC
55-85	0.0	Flow Factor + FRIC
86	0.26608	Flow Factor + FRIC

Edge	Friction Correlation	Kfactors
87-93	0.0	Flow Factor + FRIC

Pipe Wall Temperatures

Axial Level	Radial 1 (K)	Radial 2 (K)	Radial 3 (K)	Radial 4 (K)	Radial 5 (K)	Radial 6 (K)	Radial 7 (K)
1-92	505.89	505.89	505.89	505.89	505.89	505.89	505.89

Outer Wall Component

Cell	Boundary Condition
1-92	Constant BCs

Pipe Wall Power Fractions

Axial Level	Radial 1	Radial 2	Radial 3	Radial 4	Radial 5	Radial 6	Radial 7
1-92	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Pipe 10 Code Input

```

*
*****  type      num      userid      component name
pipe      10      0      unnamed
*      ncells      nodes      jun1      jun2      epsw
*      92      0      20      10      1.0E-4
*      nsides
*      0
*      ichf      iconc      pipetype      ipow      npipes
*      2      0      0      0      1
*      radin      th      houtl      houtv      toutl
*      0.0      0.0      0.0      0.0      0.0
*      toutv      pwin      pwoff      rpwmx      pwscl
*      0.0      0.0      0.0      0.0      0.0
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 e
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s

```

[illegible]

[illegible]

176

177

```

* tv *      505.89    505.89    505.89    505.89 s
* tv *      505.89    505.89    505.89    505.89 s
* tv *      505.89    505.89    505.89    505.89 s
* tv *      505.89    505.89    505.89    505.89 s
* tv *      505.89    505.89    505.89    505.89 s
* tv *      505.89    505.89    505.89    505.89 s
* tv *      505.89    505.89    505.89    505.89 s
* tv *      505.89    505.89    505.89    505.89 s
* tv *      505.89    505.89    505.89    505.89 s
* tv *      505.89    505.89    505.89    505.89 s
* tv *      505.89    505.89    505.89    505.89 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* p  *      2.639E6    2.639E6    2.639E6    2.639E6 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
* pa *      0.0      0.0      0.0      0.0 s
*
*d: Spawned Pipewall from unnamed
*****  type      num      userid      component name
htstr      10001      0      Pipewall

```

```

*      nzhstr      grids      hscyl      ichf
      92          0          1          2
*      nofuelrod   plane      liqlev      iaxcnd
      1          3          0          0
*      nmwrx       nfcil      nfcil      hdri      hdro
      0          0          1          0.0      0.0
*      nhot        nodes      fmon       nzmax      reflood
      0          7          0          276      0
*      dtxht(1)    dtxht(2)    dznht      hgapo
      0.0        0.0        1.0E-3      6300.0
*
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 s
* idbcin *        2        2        2        2 e
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 s
* idbcon *        1        1        1        1 e
* hcomon1 *       10        1        0        0 e
* hcomon1 *       10        2        0        0 e
* hcomon1 *       10        3        0        0 e
* hcomon1 *       10        4        0        0 e
* hcomon1 *       10        5        0        0 e

```

* hcomon1 *	10	6	0	0 e
* hcomon1 *	10	7	0	0 e
* hcomon1 *	10	8	0	0 e
* hcomon1 *	10	9	0	0 e
* hcomon1 *	10	10	0	0 e
* hcomon1 *	10	11	0	0 e
* hcomon1 *	10	12	0	0 e
* hcomon1 *	10	13	0	0 e
* hcomon1 *	10	14	0	0 e
* hcomon1 *	10	15	0	0 e
* hcomon1 *	10	16	0	0 e
* hcomon1 *	10	17	0	0 e
* hcomon1 *	10	18	0	0 e
* hcomon1 *	10	19	0	0 e
* hcomon1 *	10	20	0	0 e
* hcomon1 *	10	21	0	0 e
* hcomon1 *	10	22	0	0 e
* hcomon1 *	10	23	0	0 e
* hcomon1 *	10	24	0	0 e
* hcomon1 *	10	25	0	0 e
* hcomon1 *	10	26	0	0 e
* hcomon1 *	10	27	0	0 e
* hcomon1 *	10	28	0	0 e
* hcomon1 *	10	29	0	0 e
* hcomon1 *	10	30	0	0 e
* hcomon1 *	10	31	0	0 e
* hcomon1 *	10	32	0	0 e
* hcomon1 *	10	33	0	0 e
* hcomon1 *	10	34	0	0 e
* hcomon1 *	10	35	0	0 e
* hcomon1 *	10	36	0	0 e
* hcomon1 *	10	37	0	0 e
* hcomon1 *	10	38	0	0 e
* hcomon1 *	10	39	0	0 e
* hcomon1 *	10	40	0	0 e
* hcomon1 *	10	41	0	0 e
* hcomon1 *	10	42	0	0 e
* hcomon1 *	10	43	0	0 e
* hcomon1 *	10	44	0	0 e
* hcomon1 *	10	45	0	0 e
* hcomon1 *	10	46	0	0 e
* hcomon1 *	10	47	0	0 e
* hcomon1 *	10	48	0	0 e
* hcomon1 *	10	49	0	0 e
* hcomon1 *	10	50	0	0 e
* hcomon1 *	10	51	0	0 e
* hcomon1 *	10	52	0	0 e
* hcomon1 *	10	53	0	0 e
* hcomon1 *	10	54	0	0 e
* hcomon1 *	10	55	0	0 e
* hcomon1 *	10	56	0	0 e
* hcomon1 *	10	57	0	0 e
* hcomon1 *	10	58	0	0 e
* hcomon1 *	10	59	0	0 e
* hcomon1 *	10	60	0	0 e
* hcomon1 *	10	61	0	0 e
* hcomon1 *	10	62	0	0 e
* hcomon1 *	10	63	0	0 e
* hcomon1 *	10	64	0	0 e
* hcomon1 *	10	65	0	0 e
* hcomon1 *	10	66	0	0 e
* hcomon1 *	10	67	0	0 e

[illegible]

182

```

* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 e
* rdx *          1.0 e
* radrd *      0.125 0.12666667 0.12833333      0.13 0.13166667 s
* radrd * 0.13333333      0.135 e
* matrdr *      6      6      6      6 s
* matrdr *      6      6 e
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 s
* nfax *      3      3      3      3 e
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s

```

184

185

*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	e

Control Systems

Control Systems Summary Table

Component	Type
Ramp -1 [p. 186]	Control Block
Ramp -2 [p. 186]	Control Block
Sum -3 [p. 187]	Control Block
Constant -5 [p. 187]	Control Block
Problem Time 1 [p. 164]	Signal Variable

Table 9. Ramp -1

Name	Value	Refs
Type	[46] Ramp	-
Gain	-1.3195E4	-
Minimum	-3.9585E5	-
Maximum	0.0	-
Constant 1	300.0	-
Constant 2	0.0	-

Table 10. Ramp -2

Name	Value	Refs
Type	[46] Ramp	-
Gain	1.3195E4	-

Name	Value	Refs
Minimum	0.0	-
Maximum	3.9585E5	-
Constant 1	330.0	-
Constant 2	0.0	-

Table 11. Sum -3

Name	Value	Refs
Type	[57] Sum	-
Gain	1.0	-
Minimum	-1.0E20	-
Maximum	1.0E20	-
Constant 1	0.0	-
Constant 2	0.0	-
Input Connections	[3] -1, -5, -2 [below]	-

Input Sources

Input	Source	Weight
Input 1	Ramp -1 [p. 186]	1.0
Input 2	Constant -5 [p. 187]	1.0
Input 3	Ramp -2 [p. 186]	1.0

Table 12. Constant -5

Name	Value	Refs
Type	[9] Constant	-
Gain	1.0	-
Minimum	-1.0E20	-
Maximum	1.0E20	-
Constant 1	2.639E6	-
Constant 2	0.0	-

Table 13. Problem Time 1

Name	Value	Refs
Parameter Type	[0] Problem Time	-
Time Option	[0] Current Timestep	-

Control Systems Code Input

*

```

*      idcb      icbn      icb1      icb2      icb3
*      -1       46        0         0         0
*      cbgain    cbxmin    cbmax    cbcon1    cbcon2
*      -1.3195E4 -3.9585E5      0.0      300.0      0.0
*
*      idcb      icbn      icb1      icb2      icb3
*      -2       46        0         0         0
*      cbgain    cbxmin    cbmax    cbcon1    cbcon2
*      1.3195E4      0.0    3.9585E5      330.0      0.0
*
*      idcb      icbn      icb1      icb2      icb3
*      -3       57       -1        -5        -2
*      cbgain    cbxmin    cbmax    cbcon1    cbcon2
*      1.0      -1.0E20    1.0E20      0.0      0.0
*
*      idcb      icbn      icb1      icb2      icb3
*      -5        9        0         0         0
*      cbgain    cbxmin    cbmax    cbcon1    cbcon2
*      1.0      -1.0E20    1.0E20    2.639E6      0.0
*
*      idsv      isvn      ilcn      icn1      icn2
*      1         0         0         0         0

```

APPENDIX L. TRACE INPUTS FOR VAPOR MASS FRACTION TRANSIENT MODEL

TRACE Model Report

Title: TRACE Mass Fraction Transient Model
Date: Tue May 02 22:56:31 PDT 2023
Version Info: TRACE V 5.0 Patch 5
SNAP Version 4.1.0
TRACE Plug-in Version 4.5.2
Java Version 11.0.12

Component		Count
Hydraulic Components	Breaks	1
	Fills	1
	Pipes	1
	Total:	3
Control Systems	Control Blocks	4
	Signal Variables	1
	Total:	5

Hydraulic Statistics

Total Fluid Volume 1.1290 m³
hydro. system 1 1.1290 m³

Total Fluid Length 23.0 m
hydro. system 1 23.0 m

Overall Elevation Change 4.9080 m

Hydraulic System Statistics

Hydraulic System	Minimum Location	Minimum Elevation (m)	Maximum Location	Maximum Elevation (m)	Elevation Change (m)
Hydro. System 1	Pipe 10 Cell 85	0.0	Pipe 10 Cell 1	4.9080	4.9080

Model Validation Tests

10 tests performed, 10 passes, 0 failures

Loop Check

Test Description: Verifies that hydraulic components in a loop have a net elevation change of 0.0.

Error Tolerance: 0.001m

Test Result: Succeeded with no Issues.

Noncondensable Partial Pressure Test

Test Description: When the NOAIR namelist variable is set to one, every PA value must be set to 0.0. This test examines every component in the model to determine if it has a PA value > 0.0 .

Test Result: Succeeded with no Issues.

Junction Flow Area Test

Test Description: Verifies that every junction connected to two cells has a flow area that does not exceed $1.1 \times$ the average flow area of the adjacent cells.

Test Result: Succeeded with no Issues.

Abrupt Area Change Test

Test Description: Verifies that adjacent cells flow areas are within a fraction of each other. The junction between them may also be included in this check.

Include Edges between Adjacent Volumes: false

Flow Area Ratio: 2.0

Test Result Succeeded with no Issues.

Edge Hydro-Diameter Test

Test Description: Compares the hydraulic diameter specified for edges inside 1D hydraulic components with the calculated hydraulic diameter of the volumes on either side. An error is reported if the edge is not within the tolerance of one of the calculated diameters.

Tolerance (Calculated Diameter vs Actual): 1.02

Test Result: Succeeded with no Issues.

PUMPFRICTQ Consistency Test

Test Description: Verifies that power provided by enabling the PUMPFRICTQ namelist option does not conflict with power added using a FLPOW component.

Test Result: Succeeded with no Issues.

3D Hydraulic Connection Test

Test Description: This test verifies the consistency of Hydraulic connections to the vessel. The component flow direction is compared target vessel cell face to ensure a consistent flow. This also verifies that the vessel cells have a non-zero volume fraction.

Tolerance(Connection GRAV vs Expected GRAV): $1.0E-4$

Test Result: Succeeded with no Issues.

Adjacent Cell Volume Consistency Test

Test Description: Verifies adjacent cell volumes do not exceed a user specified ratio.

Include Crossflows: true

Include 3D Volumes: true

Volume Ratio: 10.0

Test Result: Succeeded with no Issues.

Spawned Component Number Test

Test Description: Spawned Component Number Test

Single Kinetics Model Test

Test Description: Verifies that only a single power component will use the kinetics model.

Test Result: Succeeded with no Issues.

Model Options

Name	Value	Refs
Title Cards	[above]	-
Version	V 5.0 Patch 5	-
Namelist Option	[1] INOPT Data After Title Cards	-
Fluids	Water	-
Transient Calculation	[1] Transient	-
Flow Parameter	[0] Liquid and Vapor Mass Flows	-
Pressure Input Option	[0] Specify Pressures	-
Water Packing	On	-
Pressure Convergence	1.0E-4	-
Steady-State Convergence	1.0E-4	-
Dispersed Fields	[0] Droplet Fields, [0] Bubble Fields	-
Pressure Iterations	10	-
Steady-State Iterations	10	-
Solute Tracking	[0] Off	-
Namelist Variables	3 Active Entries [above]	-
User Defined Units	None	-
Timestep Data	[1] Timesteps [above]	-
Trip Initiated Timestep Data	[0] Timesteps	-
Mixed Numerics	0 of 1 Enabled	-
System Gas/Liquids	Species/Mixed NC Disabled	-
Initial Condition Sets	No condition sets defined	-
Sensitivity Coefficients	[0] coefficients defined	-

Title Card

Timestep Data

Initial Timestep Size: -1.0(s)

End Time (s)	Minimum Size (s)	Maximum Size (s)	Heat vs Fluid Size Ratio	Max Conv. Power Diff (W)	Long Edit Interval (s)	Graphics Interval (s)	Restart Interval (s)	Short Edit Interval (s)
666.0	1.0E-6	0.01	10.0	1.0E20	100.0	1.0	100.0	1.0

Namelist Variables

Name	Value	Refs
Show Inactive Entries	true	-
DTSTRT - Initial timestep size	-1.0 s	-
IKFAC - Friction Option	[1] K-Factors	-
USESJC - Use Single Jun Components	[3] SJs with Angle Input	-

Model Options Code Input

```

free format
*m: SNAP:Symbolic Nuclear Analysis Package, Version 4.1.0, July 28, 2022
*m: PLUGIN:TRACE Version 4.5.2
*m: CODE:TRACE V 5.0 Patch 5
*m: DATE:5/2/23
*
*
*****
* main data *
*****
*
*      numtcr      ieos      inopt      nmat      id2o
*      0          0          1          0          0
*
*
*****
* namelist data *
*****
*
&inopts
dtstrt=-1.0,
ikfac=1,
usesjc=3,
fluids='H2O'
&end
*
*****
* Model Flags *
*****
*
*      dstep      timet
*      0          0.0
*
*      stdyst      transi      ncomp      njun      ipak
*      0          1          4          2          1
*
*      epso      epss
*      1.0E-4    1.0E-4
*
*      oitmax      sitmax      isolut      ncontr      nccfl
*      10         10         0          0          0
*
*      ntsv      ntcb      ntcf      ntrp      ntcp
*      1         4         0          0          0
*
*
*****
* component-number data *
*****
*
* Component input order (IORDER)
*-- type ---- num ----- name ----- + jun1 jun2 jun3
* PIPE * 10 s * + 20 10

```

```

* HTSTR * 10001 s * Pipewall of Pipe 10 +
* FILL * 20 s * inlet + 20
* BREAK * 30 e * outlet + 10
*
*****
* Timestep Data *
*****
* dtmin dtmax tend rtwfp
  1.0E-6 0.01 666.0 10.0
* edint gfint dmpint sedint
  100.0 1.0 100.0 1.0
*
* endflag
  -1.0

```

Hydraulic Components

Component Summary Table

Component	Volume (m ³)	Length (m)	Elevation Change (m)
Break 30 (outlet) [p. 155]	0.0	0.0	0.0
Fill 20 (inlet) [p. 155]	0.0	0.0	0.0
Pipe 10 [p. 156]	1.1290	23.0	-3.0330

Break 30 (outlet)

Name	Value	Refs
Break Type	[0] No Tables	-
Temperature Table Option	[0] Enter liquid/gas temp	-
Fluid State Option	[0] Last Interp State Held Const.	-
Length	1.0 m	-
Volume	1.0 m ³	-
Initial Gas Volume Fraction	0.999181	-
Initial Mixture Temperature	505.89 K	-
Initial Pressure	2.639E6 Pa	-
Initial Noncondensable PP	0.0 Pa	-
Adjacent Pressure Flag	false	-
Max Pressure Change Rate	1.0E20 Pa/s	-

Break 30 (outlet) Code Input

```

*
***** type      num      userid      component name
break      30      0      outlet
*   jun1      ibty      isat      ioff      adjpress
      10      0      0      0      0
*   dxin      volin      alpin      tin      pin
      1.0      1.0      0.999181      505.89      2.639E6
*   pain      concin      rbmx      poff      belv
      0.0      0.0      1.0E20      0.0      0.0

```

Fill 20 (inlet)

Name	Value	Refs
Fill Type	[10] Gen. State CS	-
Flow Condition Input	< Uneditable >	-
Length	1.0 m	-
Volume	1.0 m ³	-
Initial Gas Volume Fraction	0.999855	-
Initial Liquid Temperature	505.89 K	-
Initial Vapor Temperature	505.89 K	-
Initial Pressure	2.639E6 Pa	-
Initial Noncondensable PP	0.0 Pa	-
Initial Coolant Mass Flow	0.0 kg/s	-
Liquid Vel	40.0 m/s	-
Vapor Vel	40.0 m/s	-
Timestep Holdover	0.0	-
Rate Factor Table	None	-

State Controllers

Name	Value	Refs
Vapor Volume Controller	Sum -3 [p. 187]	-
Liquid Flow Controller	-Not Specified-	-
Vapor Flow Controller	-Not Specified-	-
Noncondensable Controller	-Not Specified-	-
Pressure Controller	-Not Specified-	-
Liquid Temperature Controller	-Not Specified-	-
Vapor Temperature Controller	-Not Specified-	-

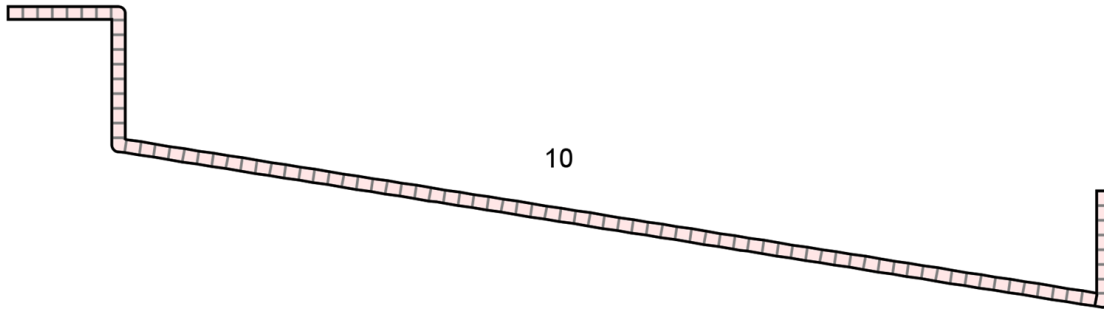
Fill 20 (inlet) Code Input

```

*
*****  type      num      userid      component name
fill      20        0          inlet
*   jun1      ifty      ioff      ifields      inpvel
      20        10        0          0          1
*   iftr      ifsv      nftb      nfsv      nfrf
      0          0          0          0          0
*   twtold      rfmX      concin      felv
      0.0        0.0        0.0        0.0
*   dxin      volin      alpin      vlin      tlin
      1.0        1.0        0.999855      40.0      505.89
*   pin      pain      flowin      vvin      tvin
      2.639E6      0.0        0.0        40.0      505.89
*   ifmlsv      ifmvsv      iftlsv      iftvsv      ifasv
      0          0          0          0          -3
*   ifpsv      ifpasv      ifcnsv
      0          0          0

```

Pipe 10



Name	Value	Refs
Component Geometry	Cells: 92, Junctions: 93 - Valid Values [above]	-
Initial Conditions	Valid Values [above]	-
Friction	Valid Values [above]	-
Fluid Power Options	Not Modeled	-
Wall Roughness	1.0E-4 m	-
Pipe Type	[0] No Accumulator	-
Number of Pipes	1	-
Enable Mixed Numerics	false	-

Hydraulic Connections

From Component	From Location	To Component	To Location
Pipe 10	Edge 1 inlet	Fill 20 (inlet) [p. 155]	Cell 1 inlet
Pipe 10	Edge 93 outlet	Break 30 (outlet) [p. 155]	Cell 1 inlet

Geometry

Pipe 10 Cell Geometry Data

Cell	Length (m)	Area (m ²)	Volume (m ³)	DZ (m)
1-7	0.25	0.049087385	0.012271846	0.0
8	0.25	0.049087385	0.012271846	-0.125
9-16	0.25	0.049087385	0.012271846	-0.24999999
17	0.25	0.049087385	0.012271846	-0.14454447
18-84	0.25	0.049087385	0.012271846	-0.039088953
85	0.25	0.049087385	0.012271846	0.10545552
86	0.25	0.049087385	0.012271846	0.25
87-92	0.25	0.049087385	0.012271846	0.24999999
Total	23.0	4.5160394	1.1290099	-3.0330488

Pipe 10 Edge Geometry Data

Edge	Flow Area (m ²)	Hydraulic Diameter (m)	CCFL Model
1-93	0.049087385	0.25	None

Pipe 10 Edge Orientation Data

Edge	Edge Angle (rad)	GRAV sin(Θ)
1-8	0.0	0.0
9-17	-1.571	-0.99999998
18-39	-0.157	-0.15635581
40-60	-0.157	-0.15635581
61-85	-0.157	-0.15635581
86	1.571	1.0
87-93	1.571	0.99999998

Cell Initial Conditions

Cell	Pressure (Pa)	Liq. Temp (K)	Gas Temp (K)	Gas Frac. (-)	NC Pressure (Pa)
1-92	2.639E6	505.89	505.89	1.0	0.0

Edge Initial Conditions

Edge	Liquid Velocity (m/s)	Vapor Velocity (m/s)
1-93	0.0	0.0

Friction Data

Edge	Friction Correlation	Kfactors
1-9	0.0	Flow Factor + FRIC
10	0.2552	Flow Factor + FRIC
11-17	0.0	Flow Factor + FRIC
18	0.24202	Flow Factor + FRIC
19-41	0.0	Flow Factor + FRIC
42	0.24202	Flow Factor + FRIC
43-53	0.0	Flow Factor + FRIC
54	0.24202	Flow Factor + FRIC
55-85	0.0	Flow Factor + FRIC
86	0.26608	Flow Factor + FRIC
87-93	0.0	Flow Factor + FRIC

Pipe Wall Temperatures

Axial Level	Radial 1 (K)	Radial 2 (K)	Radial 3 (K)	Radial 4 (K)	Radial 5 (K)	Radial 6 (K)	Radial 7 (K)
1-92	505.89	505.89	505.89	505.89	505.89	505.89	505.89

Outer Wall Component

Cell	Boundary Condition
1-92	Constant BCs

Pipe Wall Power Fractions

Axial Level	Radial 1	Radial 2	Radial 3	Radial 4	Radial 5	Radial 6	Radial 7
1-92	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Pipe 10 Code Input

```

*
*****  type      num      userid      component name
pipe      10      0      unnamed
*      ncells      nodes      jun1      jun2      epsw
      92      0      20      10      1.0E-4
*      nsides
      0
*      ichf      iconc      pipetype      ipow      npipes
      2      0      0      0      1
*      radin      th      houtl      houtv      toutl
      0.0      0.0      0.0      0.0      0.0
*      toutv      pwin      pwoff      rpwmx      pwsc1
      0.0      0.0      0.0      0.0      0.0
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 s
* dx *      0.25      0.25      0.25      0.25 e
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s
* vol * 0.012271846 0.012271846 0.012271846 0.012271846 s

```

198

```

* grav * -0.99999998 -0.99999998 -0.99999998 -0.99999998 s
* grav * -0.99999998 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 -0.15635581 -0.15635581 -0.15635581 s
* grav * -0.15635581 1.0 0.99999998 0.99999998 s
* grav * 0.99999998 0.99999998 0.99999998 0.99999998 s
* grav * 0.99999998 e
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 0.25 0.25 0.25 s
* hd * 0.25 e
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s
* nff * 1 1 1 1 s

```

200

[illegible]

*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 s
*	p	*	2.639E6	2.639E6	2.639E6	2.639E6 e
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 s
*	pa	*	0.0	0.0	0.0	0.0 e
* *						
*d: Spawned Pipewall from unnamed						
*****	type	num	userid	component name		
htstr		10001	0	Pipewall		
*	nzhstr	grids	hscyl	ichf		
	92	0	1	2		
*	nofuelrod	plane	liqlev	iaxcnd		
	1	3	0	0		
*	nmwrx	nfcil	nfcil	hdri	hdro	
	0	0	1	0.0	0.0	
*	nhot	nodes	fmon	nzmax	reflood	
	0	7	0	276	0	
*	dtxht(1)	dtxht(2)	dznht	hgapo		
	0.0	0.0	1.0E-3	6300.0		
*						

* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 e
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 e
* hcomon1 *	10	1	0	0 e
* hcomon1 *	10	2	0	0 e
* hcomon1 *	10	3	0	0 e
* hcomon1 *	10	4	0	0 e
* hcomon1 *	10	5	0	0 e
* hcomon1 *	10	6	0	0 e
* hcomon1 *	10	7	0	0 e
* hcomon1 *	10	8	0	0 e
* hcomon1 *	10	9	0	0 e
* hcomon1 *	10	10	0	0 e
* hcomon1 *	10	11	0	0 e
* hcomon1 *	10	12	0	0 e
* hcomon1 *	10	13	0	0 e
* hcomon1 *	10	14	0	0 e
* hcomon1 *	10	15	0	0 e
* hcomon1 *	10	16	0	0 e

* hcomon1 *	10	17	0	0 e
* hcomon1 *	10	18	0	0 e
* hcomon1 *	10	19	0	0 e
* hcomon1 *	10	20	0	0 e
* hcomon1 *	10	21	0	0 e
* hcomon1 *	10	22	0	0 e
* hcomon1 *	10	23	0	0 e
* hcomon1 *	10	24	0	0 e
* hcomon1 *	10	25	0	0 e
* hcomon1 *	10	26	0	0 e
* hcomon1 *	10	27	0	0 e
* hcomon1 *	10	28	0	0 e
* hcomon1 *	10	29	0	0 e
* hcomon1 *	10	30	0	0 e
* hcomon1 *	10	31	0	0 e
* hcomon1 *	10	32	0	0 e
* hcomon1 *	10	33	0	0 e
* hcomon1 *	10	34	0	0 e
* hcomon1 *	10	35	0	0 e
* hcomon1 *	10	36	0	0 e
* hcomon1 *	10	37	0	0 e
* hcomon1 *	10	38	0	0 e
* hcomon1 *	10	39	0	0 e
* hcomon1 *	10	40	0	0 e
* hcomon1 *	10	41	0	0 e
* hcomon1 *	10	42	0	0 e
* hcomon1 *	10	43	0	0 e
* hcomon1 *	10	44	0	0 e
* hcomon1 *	10	45	0	0 e
* hcomon1 *	10	46	0	0 e
* hcomon1 *	10	47	0	0 e
* hcomon1 *	10	48	0	0 e
* hcomon1 *	10	49	0	0 e
* hcomon1 *	10	50	0	0 e
* hcomon1 *	10	51	0	0 e
* hcomon1 *	10	52	0	0 e
* hcomon1 *	10	53	0	0 e
* hcomon1 *	10	54	0	0 e
* hcomon1 *	10	55	0	0 e
* hcomon1 *	10	56	0	0 e
* hcomon1 *	10	57	0	0 e
* hcomon1 *	10	58	0	0 e
* hcomon1 *	10	59	0	0 e
* hcomon1 *	10	60	0	0 e
* hcomon1 *	10	61	0	0 e
* hcomon1 *	10	62	0	0 e
* hcomon1 *	10	63	0	0 e
* hcomon1 *	10	64	0	0 e
* hcomon1 *	10	65	0	0 e
* hcomon1 *	10	66	0	0 e
* hcomon1 *	10	67	0	0 e
* hcomon1 *	10	68	0	0 e
* hcomon1 *	10	69	0	0 e
* hcomon1 *	10	70	0	0 e
* hcomon1 *	10	71	0	0 e
* hcomon1 *	10	72	0	0 e
* hcomon1 *	10	73	0	0 e
* hcomon1 *	10	74	0	0 e
* hcomon1 *	10	75	0	0 e
* hcomon1 *	10	76	0	0 e
* hcomon1 *	10	77	0	0 e
* hcomon1 *	10	78	0	0 e

[illegible]

206

```

* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 s
* dhtstrz *      0.25      0.25      0.25      0.25 e
* rdx *          1.0 e
* radrd *      0.125 0.12666667 0.12833333      0.13 0.13166667 s
* radrd * 0.13333333      0.135 e
* matrd *        6        6        6        6 s
* matrd *        6        6 e
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 s
* nfax *         3         3         3         3 e
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s
* rftn *      505.89      505.89      505.89      505.89 s

```

208

209

*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	s
*	rftn	*	505.89	505.89	505.89	505.89	e

Control Systems

Control Systems Summary Table

Component	Type
Ramp -1 [p. 186]	Control Block
Ramp -2 [p. 186]	Control Block
Sum -3 [p. 187]	Control Block
Constant -5 [p. 187]	Control Block
Problem Time 1 [p. 164]	Signal Variable

Ramp -1

Name	Value	Refs
Type	[46] Ramp	-
Gain	-9.4933E-5	-
Minimum	-2.848E-3	-
Maximum	0.0	-
Constant 1	300.0	-
Constant 2	0.0	-

Ramp -2

Name	Value	Refs
Type	[46] Ramp	-
Gain	9.4933E-5	-
Minimum	0.0	-
Maximum	2.848E-3	-
Constant 1	330.0	-
Constant 2	0.0	-

Sum -3

Name	Value	Refs
Type	[57] Sum	-
Gain	1.0	-

Name	Value	Refs
Minimum	-1.0E20	-
Maximum	1.0E20	-
Constant 1	0.0	-
Constant 2	0.0	-
Input Connections	[3] -1, -5, -2 [above]	-

Input Sources

Input	Source	Weight
Input 1	Ramp -1 [p. 186]	1.0
Input 2	Constant -5 [p. 187]	1.0
Input 3	Ramp -2 [p. 186]	1.0

Constant -5

Name	Value	Refs
Type	[9] Constant	-
Gain	1.0	-
Minimum	-1.0E20	-
Maximum	1.0E20	-
Constant 1	0.999855	-
Constant 2	0.0	-

Problem Time 1

Name	Value	Refs
Parameter Type	[0] Problem Time	-
Time Option	[0] Current Timestep	-

Control Systems Code Input

```

*
*      idcb      icbn      icb1      icb2      icb3
*      -1        46        0         0         0
*      cbgain    cbxmin    cbmax    cbcon1    cbcon2
*      -9.4933E-5 -2.848E-3  0.0      300.0     0.0
*
*      idcb      icbn      icb1      icb2      icb3
*      -2        46        0         0         0
*      cbgain    cbxmin    cbmax    cbcon1    cbcon2
*      9.4933E-5  0.0      2.848E-3  330.0     0.0
*
*      idcb      icbn      icb1      icb2      icb3
*      -3        57       -1        -5        -2
*      cbgain    cbxmin    cbmax    cbcon1    cbcon2
*      1.0      -1.0E20    1.0E20    0.0      0.0
*
*      idcb      icbn      icb1      icb2      icb3

```

	-5	9	0	0	0	
*	cbgain	cbxmin	cbmax	cbcon1	cbcon2	
	1.0	-1.0E20	1.0E20	0.999855	0.0	
*						
*	idsv	isvn	ilcn	icn1	icn2	
	1	0	0	0	0	

APPENDIX M. TRACE INPUTS FOR GUO ET AL. PRESSURE OSCILLATION COMPARISON MODEL

TRACE Model Report

Title: Guo TRACE Comparison Model
Date: Tue May 02 23:14:34 PDT 2023
Version Info: TRACE V 5.0 Patch 5
SNAP Version 4.1.0
TRACE Plug-in Version 4.5.2
Java Version 11.0.12

Component		Count
Hydraulic Components	Breaks	1
	Fills	1
	Pipes	1
	Total:	3
Control Systems	Control Blocks	4
	Signal Variables	1
	Total:	5

Hydraulic Statistics

Total Fluid Volume 0.0011395 m³
hydro. system 1 0.0011395 m³

Total Fluid Length 6.4480 m
hydro. system 1 6.4480 m

Overall Elevation Change 0.0 m

Hydraulic System Statistics

Hydraulic System	Minimum Location	Minimum Elevation (m)	Maximum Location	Maximum Elevation (m)	Elevation Change (m)
Hydro. System 1	Fill 30	0.0	Fill 30	0.0	0.0

Model Validation Tests

10 tests performed, 10 passes, 0 failures

Loop Check

Test Description: Verifies that hydraulic components in a loop have a net elevation change of 0.0.

Error Tolerance: 0.001m

Test Result: Succeeded with no Issues.

Noncondensable Partial Pressure Test

Test Description: When the NOAIR namelist variable is set to one, every PA value must be set to 0.0. This

test examines every component in the model to determine if it has a PA value > 0.0 .

Test Result: Succeeded with no Issues.

Junction Flow Area Test

Test Description: Verifies that every junction connected to two cells has a flow area that does not exceed $1.1 \times$ the average flow area of the adjacent cells.

Test Result: Succeeded with no Issues.

Abrupt Area Change Test

Test Description: Verifies that adjacent cells flow areas are within a fraction of each other. The junction between them may also be included in this check.

Include Edges between Adjacent Volumes: false

Flow Area Ratio: 2.0

Test Result Succeeded with no Issues.

Edge Hydro-Diameter Test

Test Description: Compares the hydraulic diameter specified for edges inside 1D hydraulic components with the calculated hydraulic diameter of the volumes on either side. An error is reported if the edge is not within the tolerance of one of the calculated diameters.

Tolerance (Calculated Diameter vs Actual): 1.02

Test Result: Succeeded with no Issues.

PUMPFRIQC Consistency Test

Test Description: Verifies that power provided by enabling the PUMPFRIQC namelist option does not conflict with power added using a FLPOW component.

Test Result: Succeeded with no Issues.

3D Hydraulic Connection Test

Test Description: This test verifies the consistency of Hydraulic connections to the vessel. The component flow direction is compared target vessel cell face to ensure a consistent flow. This also verifies that the vessel cells have a non-zero volume fraction.

Tolerance(Connection GRAV vs Expected GRAV): $1.0E-4$

Test Result: Succeeded with no Issues.

Adjacent Cell Volume Consistency Test

Test Description: Verifies adjacent cell volumes do not exceed a user specified ratio.

Include Crossflows: true

Include 3D Volumes: true

Volume Ratio: 10.0

Test Result: Succeeded with no Issues.

Spawned Component Number Test

Test Description: Spawned Component Number Test

Single Kinetics Model Test

Test Description: Verifies that only a single power component will use the kinetics model.

Test Result: Succeeded with no Issues.

Model Options

Name	Value	Refs
Title Cards	[above]	-
Version	V 5.0 Patch 5	-
Namelist Option	[1] INOPT Data After Title Cards	-
Fluids	Water	-
Transient Calculation	[1] Transient	-
Flow Parameter	[0] Liquid and Vapor Mass Flows	-
Pressure Input Option	[0] Specify Pressures	-
Water Packing	On	-
Pressure Convergence	1.0E-4	-
Steady-State Convergence	1.0E-4	-
Dispersed Fields	[0] Droplet Fields, [0] Bubble Fields	-
Pressure Iterations	10	-
Steady-State Iterations	10	-
Solute Tracking	[0] Off	-
Namelist Variables	3 Active Entries [above]	-
User Defined Units	None	-
Timestep Data	[1] Timesteps [above]	-
Trip Initiated Timestep Data	[0] Timesteps	-
Mixed Numerics	0 of 1 Enabled	-
System Gas/Liquids	Species/Mixed NC Disabled	-
Initial Condition Sets	No condition sets defined	-
Sensitivity Coefficients	[0] coefficients defined	-

Title Card

Timestep Data

Initial Timestep Size: -1.0(s)

End Time (s)	Minimum Size (s)	Maximum Size (s)	Heat vs Fluid Size Ratio	Max Conv. Power Diff (W)	Long Edit Interval (s)	Graphics Interval (s)	Restart Interval (s)	Short Edit Interval (s)
704.0	1.0E-6	1.0	10.0	1.0E20	100.0	1.0	100.0	1.0

Namelist Variables

Name	Value	Refs
Show Inactive Entries	true	-
DTSTRT - Initial timestep size	-1.0 s	-
IKFAC - Friction Option	[1] K-Factors	-
USESJC - Use Single Jun Components	[3] SJs with Angle Input	-

Model Options Code Input

```

free format
*m: SNAP:Symbolic Nuclear Analysis Package, Version 4.1.0, July 28, 2022
*m: PLUGIN:TRACE Version 4.5.2
*m: CODE:TRACE V 5.0 Patch 5
*m: DATE:5/2/23
*
*
*****
* main data *
*****
*
*      numtcr      ieos      inopt      nmat      id2o
*      0          0          1          0          0
*
*
*****
* namelist data *
*****
*
&inopts
dtstrt=-1.0,
ikfac=1,
usesjc=3,
fluids='H2O'
&end
*
*****
* Model Flags *
*****
*
*      dstep      timet
*      0          0.0
*
*      stdyst      transi      ncomp      njun      ipak
*      0          1          4          2          1
*
*      epso      epss
*      1.0E-4    1.0E-4
*
*      oitmax      sitmax      isolut      ncontr      nccfl
*      10         10         0          0          0
*
*      ntsv      ntcb      ntcf      ntrp      ntcp
*      1         4         0          0          0
*
*
*****
* component-number data *
*****
*
* Component input order (IORDER)
*-- type ---- num ----- name ----- + jun1 jun2 jun3
* PIPE * 10 s * Helix + 10 20

```

```

* HTSTR * 10001 s * Pipewall of Pipe 10 +
* BREAK * 20 s * + 20
* FILL * 30 e * + 10
*
*****
* Timestep Data *
*****
* dtmin dtmax tend rtwfp
  1.0E-6 1.0 704.0 10.0
* edint gfind dmpint sedint
  100.0 1.0 100.0 1.0
*
* endflag
  -1.0

```

Hydraulic Components

Component Summary Table

Component	Volume (m ³)	Length (m)	Elevation Change (m)
Break 20 [p. 217]	0.0	0.0	0.0
Fill 30 [p. 218]	0.0	0.0	0.0
Pipe 10 (Helix) [p. 156]	0.0011395	6.4480	0.0

Break 20

Name	Value	Refs
Break Type	[6] Generalized Break	-
Temperature Table Option	[0] Enter liquid/gas temp	-
Fluid State Option	[0] Last Interp State Held Const.	-
Length	1.0 m	-
Volume	1.0 m ³	-
Initial Gas Volume Fraction	1.0	-
Initial Mixture Temperature	477.22 K	-
Initial Pressure	1.6915E6 Pa	-
Initial Noncondensable PP	0.0 Pa	-
Adjacent Pressure Flag	false	-

Scale Factors

Name	Value	Refs
Pressure Scale	1.0	-
Liquid Temperature Scale	1.0	-
Noncondensable Scale	1.0	-

State Controllers

Name	Value	Refs
Pressure Controller	Sum -4 [p. 235]	-

Name	Value	Refs
Liquid Temperature Controller	-Not Specified-	-
Vapor Temperature Controller	-Not Specified-	-
Vapor Volume Controller	-Not Specified-	-
Noncondensable Controller	-Not Specified-	-

Break 20 Code Input

```

*
***** type      num      userid      component name
break      20      0      unnamed
*   jun1      ibty      isat      ioff      adjpress
      20      6      0      0      0
*   ibtr      ibsv      nbtr      nbsv      nbrf
      0      0      0      0      0
*   dxin      volin      alpin      tin      pin
      1.0      1.0      1.0      477.22      1.6915E6
*   pain      concin      rbmx      poff      belv
      0.0      0.0      1.0E20      0.0      0.0
*   ibpsv      ibtlsv      ibtvsv      ibasv      ibpasv
      -4      0      0      0      0
*   ibcnsv
      0

```

Fill 30

Name	Value	Refs
Fill Type	[1] Constant Velocity	-
Length	1.0 m	-
Volume	1.0 m ³	-
Initial Gas Volume Fraction	1.0	-
Initial Liquid Temperature	477.22 K	-
Initial Vapor Temperature	477.22 K	-
Initial Pressure	1.6915E6 Pa	-
Initial Noncondensable PP	0.0 Pa	-
Liquid Vel	110.0 m/s	-
Vapor Vel	110.0 m/s	-
Max Acceleration	1.0E20 m/s ²	-
Rate Factor Table	None	-

Contan Coupling

Name	Value	Refs
Couple to CONTAN	false	-

Fill 30 Code Input

```

*
***** type      num      userid      component name
fill      30      0      unnamed
*   jun1      ifty      ioff

```

```

10      1      0
*      twtold      rfm      concin      felv
0.0      1.0E20      0.0      0.0
*      dxin      volin      alpin      vlin      tlin
1.0      1.0      1.0      110.0      477.22
*      pin      pain      flowin      vvin      tvin
1.6915E6      0.0      0.0      110.0      477.22

```

Pipe 10 (Helix)

10

Name	Value	Refs
Component Geometry	Cells: 92, Junctions: 93 - Valid Values [above]	-
Initial Conditions	Valid Values [above]	-
Friction	Valid Values [above]	-
Fluid Power Options	Enabled [below]	-
Wall Roughness	1.0E-4 m	-
Pipe Type	[0] No Accumulator	-
Number of Pipes	1	-
Enable Mixed Numerics	false	-

Hydraulic Connections

From Component	From Location	To Component	To Location
Pipe 10	Edge 1 inlet	Fill 30 [p. 218]	Cell 1 inlet
Pipe 10	Edge 93 outlet	Break 20 [p. 217]	Cell 1 inlet

Geometry

Pipe 10 Cell Geometry Data

Cell	Length (m)	Area (m ²)	Volume (m ³)	DZ (m)
1-92	0.070086957	1.767146E-4	1.238539E-5	0.0
Total	6.448	0.016257742	1.139456E-3	0.0

Pipe 10 Edge Geometry Data

Edge	Flow Area (m ²)	Hydraulic Diameter (m)	CCFL Model
1-93	1.767146E-4	0.015	None

Pipe 10 Edge Orientation Data

Edge	Edge Angle (rad)	GRAV sin(Θ)
1-93	0.0	0.0

Cell Initial Conditions

Cell	Pressure (Pa)	Liq. Temp (K)	Gas Temp (K)	Gas Frac. (-)	NC Pressure (Pa)
1-92	1.6915E6	477.22	477.22	1.0	0.0

Edge Initial Conditions

Edge	Liquid Velocity (m/s)	Vapor Velocity (m/s)
1-93	0.0	0.0

Friction Data

Edge	Friction Correlation	Kfactors
1-93	0.0	Flow Factor + FRIC

Pipe Wall Temperatures

Axial Level	Radial 1 (K)	Radial 2 (K)	Radial 3 (K)	Radial 4 (K)	Radial 5 (K)	Radial 6 (K)	Radial 7 (K)	Radial 8 (K)
1-92	477.22	477.22	477.22	477.22	477.22	477.22	477.22	477.22

Outer Wall Component

Cell	Boundary Condition
1-92	Constant BCs

Pipe Wall Power Fractions

Axial Level	Radial 1	Radial 2	Radial 3	Radial 4	Radial 5	Radial 6	Radial 7	Radial 8
1-92	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Fluid Options

Main Tube

Property	Value
Initial Fluid Power	1.0E5 W
Fluid Power Off	0.0 W
Maximum Power Rate	1.0E20 W/s
Power Scale Factor	1.0
Fluid Power Table	0 Rows
Fluid Power Rate Factor Table	0 Rows

Pipe 10 (Helix) Code Input

```

*
*****  type      num      userid      component name
pipe      10        0          Helix
*   ncells      nodes      jun1      jun2      epsw
    92          0        10        20      1.0E-4
*   nsides
    0

```

221

222

* grav *	0.0	0.0	0.0	0.0 s
* grav *	0.0 e			
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015	0.015	0.015	0.015 s
* hd *	0.015 e			
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1	1	1	1 s
* nff *	1 e			
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s
* alp *	1.0	1.0	1.0	1.0 s

224

[illegible]

```

* p * 1.6915E6 1.6915E6 1.6915E6 1.6915E6 s
* p * 1.6915E6 1.6915E6 1.6915E6 1.6915E6 s
* p * 1.6915E6 1.6915E6 1.6915E6 1.6915E6 s
* p * 1.6915E6 1.6915E6 1.6915E6 1.6915E6 e
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
* pa * 0.0 0.0 0.0 0.0 s
*
*d: Spawned Pipewall from Helix
***** type num userid component name
htstr 10001 0 Pipewall
* nzhstr grids hscyl ichf
92 0 1 2
* nofuelrod plane liqlev iaxcnd
1 3 0 0
* nmwrx nfci nfcil hdri hdro
0 0 1 0.0 0.0
* nhot nodes fmon nzmax reflood
0 8 0 276 0
* dtxht(1) dtxht(2) dznht hgapo
0.0 0.0 1.0E-3 6300.0
*
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s
* idbcin * 2 2 2 2 s

```

* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 s
* idbcin *	2	2	2	2 e
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 s
* idbcon *	1	1	1	1 e
* hcomon1 *	10	1	0	0 e
* hcomon1 *	10	2	0	0 e
* hcomon1 *	10	3	0	0 e
* hcomon1 *	10	4	0	0 e
* hcomon1 *	10	5	0	0 e
* hcomon1 *	10	6	0	0 e
* hcomon1 *	10	7	0	0 e
* hcomon1 *	10	8	0	0 e
* hcomon1 *	10	9	0	0 e
* hcomon1 *	10	10	0	0 e
* hcomon1 *	10	11	0	0 e
* hcomon1 *	10	12	0	0 e
* hcomon1 *	10	13	0	0 e
* hcomon1 *	10	14	0	0 e
* hcomon1 *	10	15	0	0 e
* hcomon1 *	10	16	0	0 e
* hcomon1 *	10	17	0	0 e
* hcomon1 *	10	18	0	0 e
* hcomon1 *	10	19	0	0 e
* hcomon1 *	10	20	0	0 e
* hcomon1 *	10	21	0	0 e
* hcomon1 *	10	22	0	0 e
* hcomon1 *	10	23	0	0 e
* hcomon1 *	10	24	0	0 e
* hcomon1 *	10	25	0	0 e
* hcomon1 *	10	26	0	0 e
* hcomon1 *	10	27	0	0 e
* hcomon1 *	10	28	0	0 e
* hcomon1 *	10	29	0	0 e
* hcomon1 *	10	30	0	0 e
* hcomon1 *	10	31	0	0 e
* hcomon1 *	10	32	0	0 e
* hcomon1 *	10	33	0	0 e
* hcomon1 *	10	34	0	0 e
* hcomon1 *	10	35	0	0 e

* hcomon1 *	10	36	0	0 e
* hcomon1 *	10	37	0	0 e
* hcomon1 *	10	38	0	0 e
* hcomon1 *	10	39	0	0 e
* hcomon1 *	10	40	0	0 e
* hcomon1 *	10	41	0	0 e
* hcomon1 *	10	42	0	0 e
* hcomon1 *	10	43	0	0 e
* hcomon1 *	10	44	0	0 e
* hcomon1 *	10	45	0	0 e
* hcomon1 *	10	46	0	0 e
* hcomon1 *	10	47	0	0 e
* hcomon1 *	10	48	0	0 e
* hcomon1 *	10	49	0	0 e
* hcomon1 *	10	50	0	0 e
* hcomon1 *	10	51	0	0 e
* hcomon1 *	10	52	0	0 e
* hcomon1 *	10	53	0	0 e
* hcomon1 *	10	54	0	0 e
* hcomon1 *	10	55	0	0 e
* hcomon1 *	10	56	0	0 e
* hcomon1 *	10	57	0	0 e
* hcomon1 *	10	58	0	0 e
* hcomon1 *	10	59	0	0 e
* hcomon1 *	10	60	0	0 e
* hcomon1 *	10	61	0	0 e
* hcomon1 *	10	62	0	0 e
* hcomon1 *	10	63	0	0 e
* hcomon1 *	10	64	0	0 e
* hcomon1 *	10	65	0	0 e
* hcomon1 *	10	66	0	0 e
* hcomon1 *	10	67	0	0 e
* hcomon1 *	10	68	0	0 e
* hcomon1 *	10	69	0	0 e
* hcomon1 *	10	70	0	0 e
* hcomon1 *	10	71	0	0 e
* hcomon1 *	10	72	0	0 e
* hcomon1 *	10	73	0	0 e
* hcomon1 *	10	74	0	0 e
* hcomon1 *	10	75	0	0 e
* hcomon1 *	10	76	0	0 e
* hcomon1 *	10	77	0	0 e
* hcomon1 *	10	78	0	0 e
* hcomon1 *	10	79	0	0 e
* hcomon1 *	10	80	0	0 e
* hcomon1 *	10	81	0	0 e
* hcomon1 *	10	82	0	0 e
* hcomon1 *	10	83	0	0 e
* hcomon1 *	10	84	0	0 e
* hcomon1 *	10	85	0	0 e
* hcomon1 *	10	86	0	0 e
* hcomon1 *	10	87	0	0 e
* hcomon1 *	10	88	0	0 e
* hcomon1 *	10	89	0	0 e
* hcomon1 *	10	90	0	0 e
* hcomon1 *	10	91	0	0 e
* hcomon1 *	10	92	0	0 e
* htc2 *	0.0	300.0	e	
* htc2 *	0.0	300.0	e	
* htc2 *	0.0	300.0	e	
* htc2 *	0.0	300.0	e	
* htc2 *	0.0	300.0	e	

[illegible]

[illegible]

231

232

233

```

* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 s
* rftn * 477.22 477.22 477.22 477.22 e

```

Control Systems

Control Systems Summary Table

Component	Type
Constant -1 [p. 234]	Control Block
Ramp -2 [p. 186]	Control Block
Ramp -3 [p. 235]	Control Block
Sum -4 [p. 235]	Control Block
Problem Time 1 [p. 164]	Signal Variable

Constant -1

Name	Value	Refs
Type	[9] Constant	-
Gain	1.0	-
Minimum	-1.0E20	-
Maximum	1.0E20	-
Constant 1	1.6915E6	-
Constant 2	0.0	-

Ramp -2

Name	Value	Refs
Type	[46] Ramp	-
Gain	1.29E4	-
Minimum	0.0	-
Maximum	6.675E5	-
Constant 1	300.0	-
Constant 2	0.0	-

Ramp -3

Name	Value	Refs
Type	[46] Ramp	-
Gain	-1.29E4	-
Minimum	-6.675E5	-
Maximum	0.0	-
Constant 1	352.0	-
Constant 2	0.0	-

Sum -4

Name	Value	Refs
Type	[57] Sum	-
Gain	1.0	-
Minimum	-1.0E20	-
Maximum	1.0E20	-
Constant 1	0.0	-
Constant 2	0.0	-
Input Connections	[3] -2, -1, -3 [below]	-

Input Sources

Input	Source	Weight
Input 1	Ramp -2 [p. 186]	1.0
Input 2	Constant -1 [p. 234]	1.0
Input 3	Ramp -3 [p. 235]	1.0

Problem Time 1

Name	Value	Refs
Parameter Type	[0] Problem Time	-
Time Option	[0] Current Timestep	-

Control Systems Code Input

```

*
*   idcb   icbn   icb1   icb2   icb3
*   -1      9      0      0      0
*   cbgain  cbxmin  cbmax  cbcon1  cbcon2
*   1.0    -1.0E20  1.0E20  1.6915E6  0.0
*
*   idcb   icbn   icb1   icb2   icb3
*   -2     46      0      0      0
*   cbgain  cbxmin  cbmax  cbcon1  cbcon2
*   1.29E4   0.0    6.675E5  300.0  0.0
*

```

```

*      idcb      icbn      icb1      icb2      icb3
*      -3       46        0         0         0
*      cbgain    cbxmin    cbmax    cbcon1    cbcon2
*      -1.29E4   -6.675E5    0.0     352.0     0.0
*
*      idcb      icbn      icb1      icb2      icb3
*      -4       57       -2       -1       -3
*      cbgain    cbxmin    cbmax    cbcon1    cbcon2
*      1.0      -1.0E20    1.0E20    0.0     0.0
*
*      idsv      isvn      ilcn      icn1      icn2
*      1         0         0         0         0

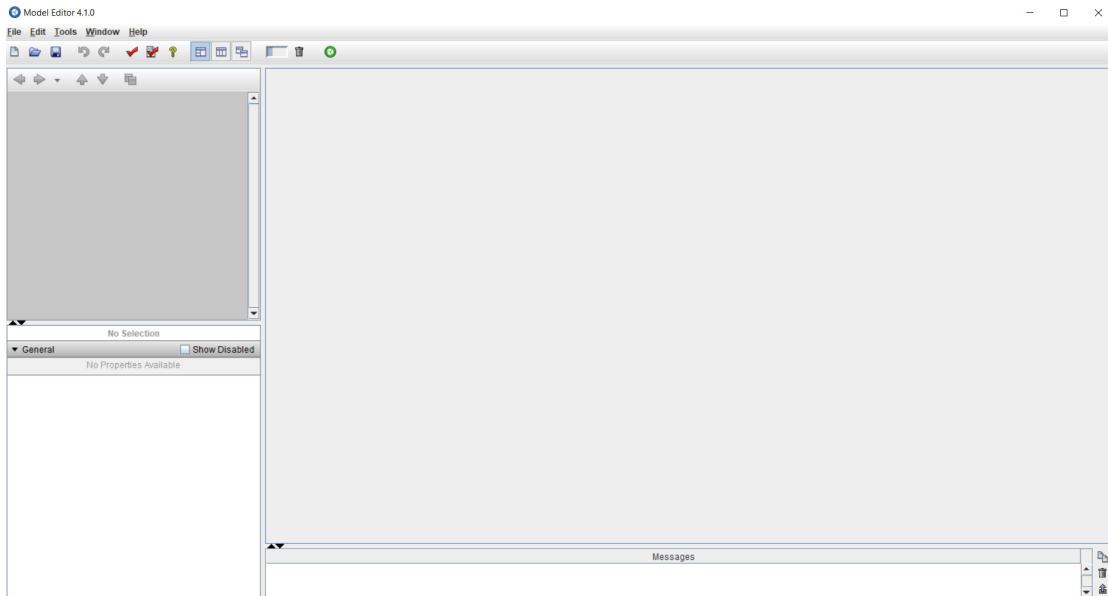
```

APPENDIX N. TRACE USER INTERFACE MODEL DEVELOPMENT EXAMPLE

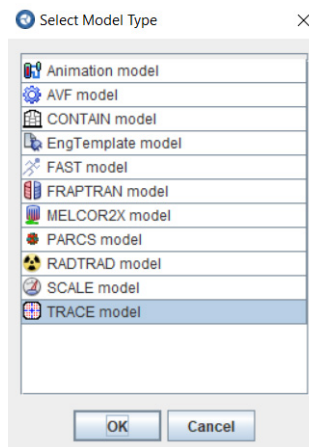
1. **Create a new TRACE model.** Open the SNAP model editor. It is the blue circular icon shown below.



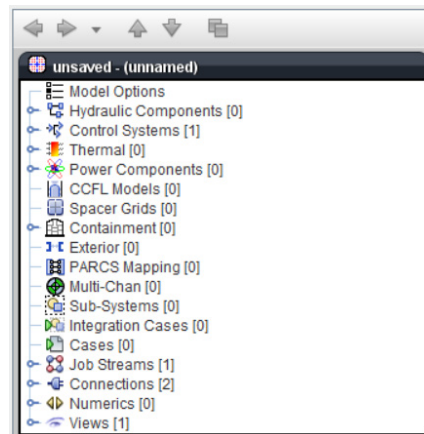
Once open, the model editor screen should look like this:



The top left section is where model options will appear. The lower left section is where the properties of selected components will appear. The large section on the right is the default view where model components will be displayed. In the top right task bar, choose file > new > TRACE model.

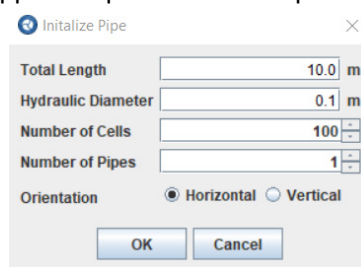


You should now see a new tree on the right with TRACE model options:

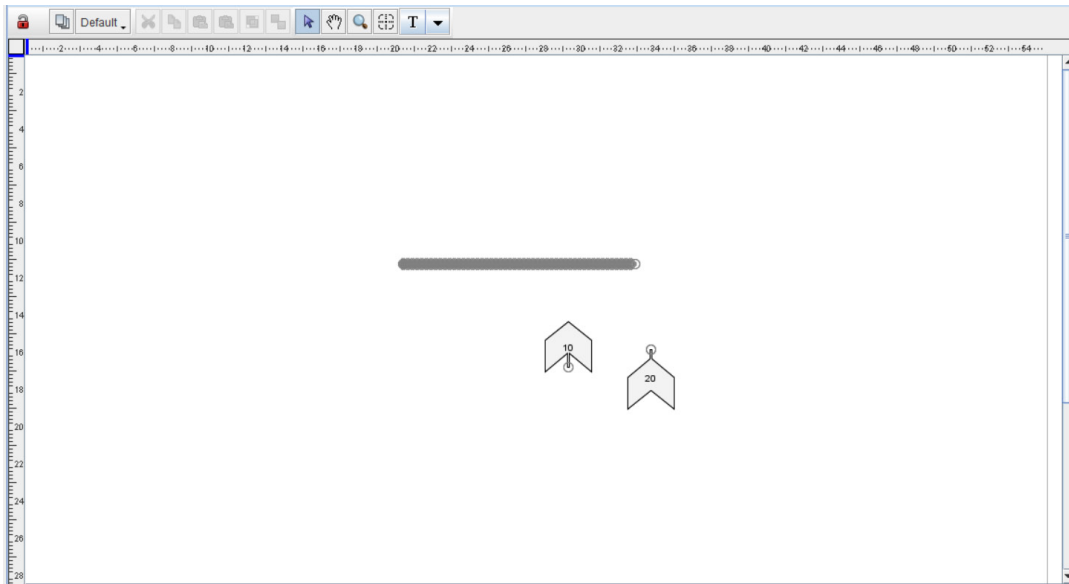


Select file > save as, and save your model under a desired file name. In this case, “Stonhill_TRACE_FQ” was chosen.

2. **Create fill and break components.** In the TRACE model option tree, expand the section called “Hydraulic Components” and right click “Breaks” and select “New”. This should create a break under hydraulic components > breaks in the model option tree. Right click on break you created in the tree and select add to view > default view. Repeat these actions for “Fills”. Your default view should now include your fill and break components. These are analogous to the inlet and outlet of the pipe.
3. **Create the pipe component.** right click “Pipes” and select “New”. A window with pipe initialization options will appear. Input the desired parameters:



Right click on break you created in the tree and select add to view > default view. Your default view should now include all three components:



4. **Change Pipe inputs.** select Views > Default View, and change the pixels per meter to 100 for a better view of the pipe cells. Double click on the pipe component in the default view and add the following characteristics:
 - a. **Initial conditions:** Input the first row of initial conditions, copy that row, select the other 99 rows, and paste the same conditions in.

Initial Conditions - Pipe 30 (Pipe) ×

Row Order: Automatic

Cell Number	Pressure (Pa)	Liquid Temp (K)	Vapor Temp (K)	Gas Volume Fraction	NC Partial Pressure (Pa)
1	2.639E6	500.002	500.002	1.0	0.0
2	2.639E6	500.002	500.002	1.0	0.0
3	2.639E6	500.002	500.002	1.0	0.0
4	2.639E6	500.002	500.002	1.0	0.0
5	2.639E6	500.002	500.002	1.0	0.0
6	2.639E6	500.002	500.002	1.0	0.0
7	2.639E6	500.002	500.002	1.0	0.0
8	2.639E6	500.002	500.002	1.0	0.0
9	2.639E6	500.002	500.002	1.0	0.0
10	2.639E6	500.002	500.002	1.0	0.0

Steam Tables

Cells Edges

Close

b. Inlet:

Define Connection

Target Component: Fill 20

Target Node: 1

Target Face: [JUN1] Inlet

OK Cancel

c. Outlet:

Define Connection

Target Component Break 10

Target Node 1

Target Face [JUN1] Inlet

OK Cancel

Ensure that the desired pipe setting match those shown in the component tree:

Pipe 30 (Pipe) - Properties View

🔍 **Pipe 30 (Pipe)**

▼ **General** ☐ Show Disabled

Component Name	Pipe	⏪ ⏩ 📄 ?
Component Number	30	📄 ?
Description	<none>	⚠️ 📄 ?
Component Geometry	Cells: 100	⚠️ 📄 ?
Initial Conditions	[Valid Conditions]	⚠️ 📄 ?
Friction	Kfac (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	⚠️ 📄 ?
Fluid Power Options	Not Modeled	⚠️ 📄 ?
Wall Roughness	0.0 (m) ⏪ ⏩	📄 ?
Inlet	Fill 20 Cell 1 inlet	⚠️ 📄 ?
Outlet	Break 10 Cell 1 inlet	⚠️ 📄 ?
Cross Flow Connections	[0] Connections	⚠️ 📄 ?
Pipe Type	[0] No Accumulator ▼	📄 ?
Number of Pipes	1 ⏪ ⏩	📄 ?
Leak Paths	[0] Leak Paths	⚠️ 📄 ?
Enable Mixed Numerics	☐ True ☒ False	📄 ?

▼ **Pipe Wall**

Use Pipewall	☐ True ☒ False	📄 ?
--------------	---	-----

▶ **Wall Power**

▶ **Trace Species**

Close

If so, close this window and proceed.

5. **Change fill inputs.** Double click on the fill component in the default view section. Change the fill settings as needed and close the window.

Fill 20 (Fill) - Properties View

Fill 20 (Fill)

☐ Show Disabled

Component Name	Fill	
Component Number	20	
Description	<none>	
Inlet	Pipe 30 Cell 1 inlet	
Fill Type	[1] Constant Velocity	
Length	1.0 (m)	
Volume	1.0 (m ³)	
Initial Gas Volume Fraction	0.0 (-)	
Initial Liquid Temperature	500.002 (K)	
Initial Vapor Temperature	500.002 (K)	
Initial Pressure	2.639E6 (Pa)	
Initial Noncondensable PP	0.0 (Pa)	
Liquid Vel	40.0 (m/s)	
Vapor Vel	40.0 (m/s)	
Max Acceleration	1.0E20 (m/s ²)	
Rate Factor Table	Rows: 0 []	

▼ Contan Coupling

Couple to CONTAN ☐ True ☒ False

► Scale Factors

► State Controllers

Close

6. **Change break inputs.** Double click on the break component in the default view section. Change the break settings close the window.

Break 10

▼ General Show Disabled

Component Name	unnamed		
Component Number	10		
Description	<none>	E	
Inlet	Pipe 30 Cell 100 out...	E	
Break Type	[0] No Tables		
Temperature Table Option	[0] Enter liquid/gas t...		
Fluid State Option	[0] Last Interp State ...		
Length	1.0 (m)		
Volume	1.0 (m³)		
Initial Gas Volume Fraction	1.0 (-)		
Initial Mixture Temperature	500.002 (K)		
Initial Pressure	2.639E6 (Pa)		
Initial Noncondensable PP	0.0 (Pa)		
Adjacent Pressure Flag	☐ True ☒ False		
Max Pressure Change Rate	1.0E20 (Pa/s)		

7. **Create job stream.** In the model options tree, right click on Job Streams > Single_Step and select add to view > default view. Click on Job Streams > Single_Step and change the name as desired. In this case, "Stonhill_TRACE_FQ_100" The root folder is the folder that the data will be written to. Ensure that the following job stream settings are chosen:

Stonhill_TRACE_FQ_100

▼ General Show Disabled

Name	Stonhill_TRACE_FQ_100		
Description	<none>	E	
Stream Type	Basic Job Stream	E	
Platform	Local		
Root Folder	TRACE_Runs	E	
Relative Location			
Log Level	Information		
View in Job Status	☒ Yes ☐ No		
Linear Execution	☐ Yes ☒ No		
ECI Support	☐ Enabled ☒ Disabled		

8. **Set run options.** Click on Model Options at the top of the model options tree. Select Timestep Data from this tree and ensure that the desired timestep settings are entered:

Edit Timestep Data

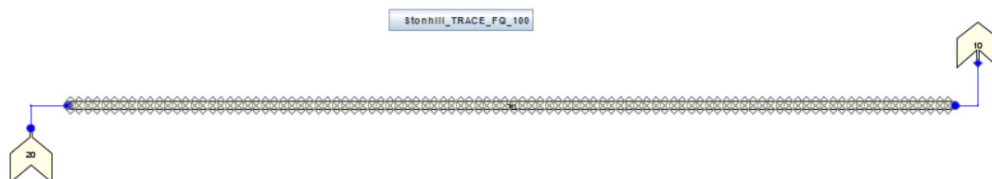
Initial Timestep Size: -1.0 s

End Time	Minimum Size	Maximum Size	Heat vs Fluid Size	Max Conv. Power Diff	Long Edit Interval	Graphics Interval	Restart Interval	Short Edit Interval
100.0	1.0E-6	1.0	10.0	1.0E20	100.0	1.0	100.0	1.0

Click okay. Ensure that these settings are reflected in the model options tree:

Model Options	
▼ General ☐ Show Disabled	
Model Name	unnamed
Title Cards	<none>
Description	<none>
Version	V 5.0 Patch 5
Namelist Option	[1] INOPT Data ...
Fluids	H2O
Restart Number	☐ Auto
Start Time	☐ Auto (s)
Transient Calculation	[0] Steady State
Steady State Mode	[1] GSS Calcula...
Flow Parameter	[0] Liquid and V...
Pressure Input Option	[0] Specify Pres...
Water Packing	On
Pressure Convergence	1.0E-4 (-)
Steady-State Convergence	1.0E-4 (-)
Dispersed Fields	[0] Droplet Field...
Pressure Iterations	10
Steady-State Iterations	10
Solute Tracking	[0] Off
Namelist Variables	Valid values
User Defined Units	< none >
Timestep Data	[1] Timesteps
Trip Initiated Timestep Data	[0] Timesteps
Mixed Numerics	0 of 1 Enabled
Adjusted Additive Flow Loss	[0] Loss Control...
System Gas/Liquids	None
Noncond. Gas Option	☐ [1] Air
Model Validation	[10] Active Tests...
Initial Condition Sets	[0] Initial Condi...
Sensitivity Coefficients	[0] Coefficients

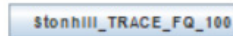
- Run the model.** Your default view should look include a box labeled with your job stream name and blue connections between your pipe and the fill and break:



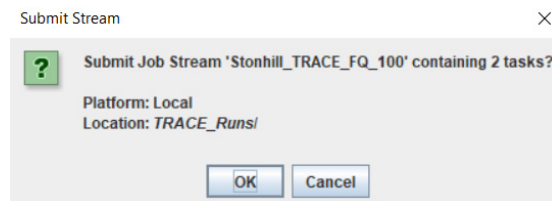
If the padlock icon at the top left of the default view is red and unlocked, click it so that it changes to green and locked. This will allow you to lock the default view and run the simulation. Now, open the SNAP job editor. The icon should look like the model editor icon but in green.



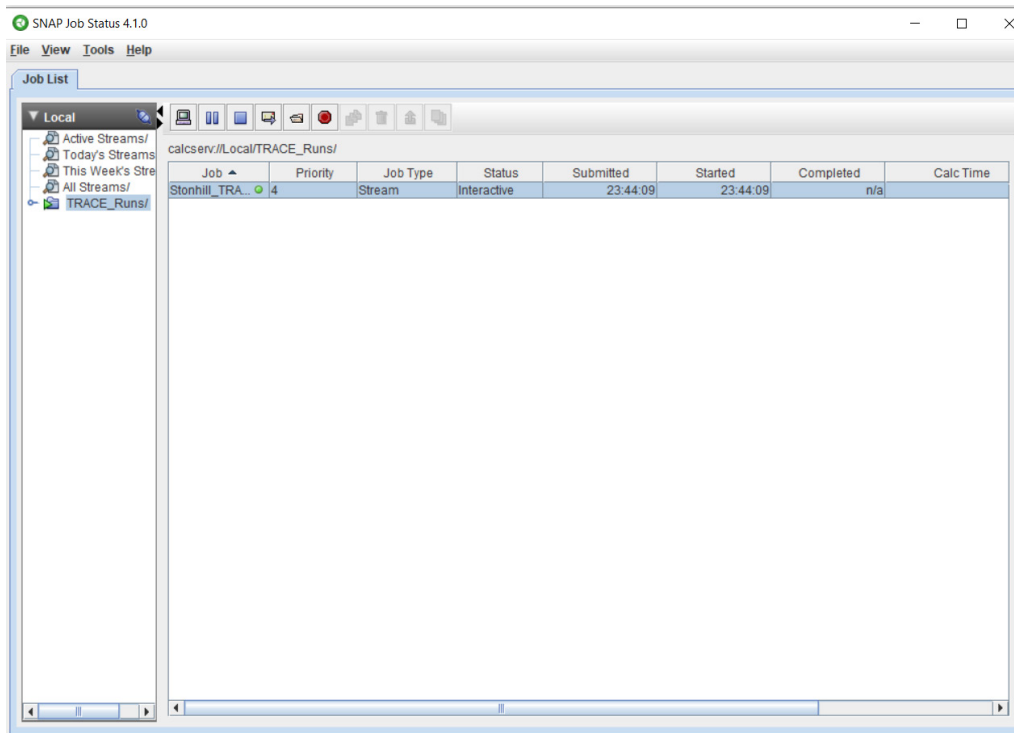
Click on the job stream box in your model editor default view.



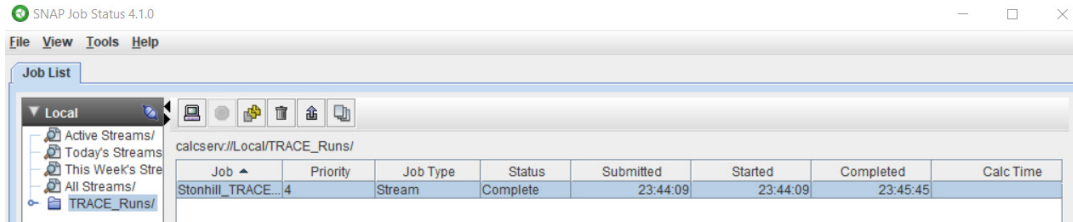
Click yes to the dialog box that pops up.



Now, go to the SNAP job editor window. Click on all streams in the tree on the left side. Your job submission should then populate in the main window:



When a time populates in the “Completed” column, the simulation has converged successfully. This should take no more than 30 seconds for steady-state simulations and two minutes for transient cases.



The data from the simulation is now written to a folder within the one you specified in the job stream section of the model editor. The results can be viewed through the Aptplot application which is part of the TRACE suite. The data can also be imported into MATLAB through the procedure used in Appendix F through Appendix I. For transient model development, refer to the TRACE User Input Manual for additional guidance.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- [1] M. Holmgren. *X Steam, Thermodynamic Properties of Water and Steam*. 2006 [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/9817-x-steam-thermodynamic-properties-of-water-and-steam.gov/chemistry/fluid/>
- [2] National Institute of Standards and Technology, “Thermophysical properties of fluid systems,”. 2023 [Online]. Available: <https://webbook.nist>
- [3] J. M. Quibén, “Experimental and analytical study of two-phase pressure drops during evaporation in horizontal tubes,” Ph.D. dissertation, Dept. of Mech. Eng., Pierre and Marie Curie Univ., Paris, FR, 2005.
- [4] J. G. Collier and J. R. Thome, *Convective Boiling and Condensation*, 3rd ed. New York, NY, USA: Oxford University Press, 1996.
- [5] B. J. Azzopardi and P. B. Whalley, “The effect of flow patterns on two-phase flow in a T junction,” *Int. J. Multiph. Flow*, vol. 8, no. 5, pp. 491–507, Oct. 1982 [Online]. Available: [https://doi.org/10.1016/0301-9322\(82\)90020-9](https://doi.org/10.1016/0301-9322(82)90020-9)
- [6] Crane LTD. Technical Publications Unit, “Crane flow of fluids through valves, fittings and pipe,” Ongar, Essex, UK, Rep. 410, 1980 [Online]. Available: <https://www.cranesbu.com/en/news/view/new-flow-of-fluids-tp-410-2018-edition-now-available>
- [7] B. J. Azzopardi, “The effect of the side arm diameter on the two-phase flow split at a ‘T’ junction,” *Int. J. Multiph. Flow*, vol. 10, no. 4, pp. 509–512, Aug. 1984 [Online]. Available: [https://doi.org/10.1016/0301-9322\(84\)90059-4](https://doi.org/10.1016/0301-9322(84)90059-4)
- [8] W. T. M. Al-Tameemi, “Two-phase flow in straight pipes and across 90 degrees sharp-angled mitre elbows,” Ph.D dissertation, Dept. of Mech. Eng., Univ. of Sheffield, Sheffield, SYK, UK, 2018.
- [9] M. J. Moran, H. N. Shapiro, D. D. Boettner, and M. B. Bailey, *Fundamentals of Engineering Thermodynamics*, 8th ed. New York, NY, USA: John Wiley and Sons, 2014.
- [10] F. P. Incropera, *Fundamentals of Heat and Mass Transfer*, 6th ed. Hoboken, NJ, USA: John Wiley and Sons, 2007.
- [11] M. Kuramae and M. Ito, “Some considerations on the physical models of a heat pipe for analyzing dynamic heat transfer characteristics,” *Society of Chemical Engineers Japan*, vol. 15, pp. 336–344, Jan. 1989 [Online]. Available: <https://doi.org/10.1252/kakoronbunshu.15.1064>

- [12] V. P. Carey, *Liquid-Vapor Phase-Change Phenomena*, 2nd ed. Bristol, PA, USA: Taylor & Francis Group LLC., 2008.
- [13] B. E. Axelsson, “Steady state modeling of wet steam in a piping system,” M.S. thesis, Dept. of Mech. Eng., NPS, Monterey, CA, USA, 2022 [Online]. Available: <https://calhoun.nps.edu/handle/10945/71429>
- [14] L. J. Guo, Z. P. Feng, and X. J. Chen, “Pressure drop oscillation of steam–water two-phase flow in a helically coiled tube,” *Int. J. Heat Mass Transf.*, vol. 44, no. 8, pp. 1555–1564, Apr. 2001 [Online]. Available: [https://doi.org/10.1016/S0017-9310\(00\)00211-8](https://doi.org/10.1016/S0017-9310(00)00211-8)
- [15] Division of Assessment and Special Projects, “TRACE V5.0 theory manual: Field equations, solution methods, and physical models,” United States Nuclear Regulatory Commission, Washington, DC, USA, 2020 [Online]. Available: <https://www.nrc.gov/docs/ML1200/ML120060218.pdf>
- [16] B. R. Munson, A. P. Rothmayer, and T. H. Okiishi, *Fundamentals of Fluid Mechanics*, 7th ed. Hoboken, NJ, USA: John Wiley and Sons, 2012.
- [17] L. F. Moody, “Friction factors for pipe flow,” *American Society of Mechanical Engineers*, vol. 66, pp. 671678, 1944.
- [18] I. E. Idel’chik, *Handbook of Hydraulic Resistance*. Moscow, USSR: State Energy Publishing House, 1960.
- [19] J. Nikuradze, “Laws of flow in rough pipes,” *Open Journal of Fluid Dynamics*, vol. 4, Aug. 1933 [Online]. Available: <https://doi.org/10.1080/14685240902806491>
- [20] E. N. Sieder and G. E. Tate, “Heat transfer and pressure drop of liquids in tubes,” *Ind. Eng. Chem.*, vol. 28, no. 12, pp. 1429–1435, Dec. 1936 [Online]. Available: <https://doi.org/10.1021/ie50324a027>

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California



DUDLEY KNOX LIBRARY

NAVAL POSTGRADUATE SCHOOL

WWW.NPS.EDU

WHERE SCIENCE MEETS THE ART OF WARFARE