NASA/TM—2013–217492

Generalized Fluid System Simulation Program, Version 6.0

A.K. Majumdar, A.C. LeClair, and R. Moore
Marshall Space Flight Center, Huntsville, Alabama

P.A. Schallhorn
Kennedy Space Center, Florida

October 2013

The NASA STI Program…in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and mission, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results…even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>

- E-mail your question via the Internet to <help@sti.nasa.gov>

- Fax your question to the NASA STI Help Desk at 443–757–5803

- Phone the NASA STI Help Desk at 443–757–5802

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

Generalized Fluid System Simulation Program, Version 6.0

A.K. Majumdar, A.C. LeClair, and R. Moore
Marshall Space Flight Center, Huntsville, Alabama

P.A. Schallhorn
Kennedy Space Center, Florida

Acknowledgments

PREFACE

The motivation to develop a general purpose computer program to compute pressure and flow distribution in a complex fluid network came from the need to calculate the axial load on the bearings in a turbopump. During the early years of space shuttle main engine development, several specific purpose codes were developed to model the turbopumps. However, it was difficult to use those codes for a new design without making extensive changes in the original code. Such efforts often turn out to be time consuming and inefficient. To satisfy the need to model these turbopumps in an efficient and timely manner, development of the Generalized Fluid System Simulation Program (GFSSP) was started at Marshall Space Flight Center (MSFC) in March 1994. The objective was to develop a general fluid flow system solver capable of handling phase change, compressibility, and mixture thermodynamics. Emphasis was given to construct a user-friendly program using a modular structured code. The intent of this effort was that an engineer with an undergraduate background in fluid mechanics and thermodynamics should be able to rapidly develop a reliable model. The interest in modular code development was intended to facilitate future modifications to the program.

The code development was carried out in several phases. At the end of each phase a workshop was held where the latest version of the code was released to MSFC engineers for testing, verification, and feedback. The steady state version of GFSSP (version 1.4) was first released in October 1996. This version is also commercially available through the Open Channel Foundation. The unsteady version was released in October 1997 (version 2.0). A graphical user interface (GUI) for GFSSP was developed and was part of Version 3.0 which was released in November 1999. GFSSP (version 3.0) won the NASA Software of the Year award in 2001. Fluid transient (water hammer) capability was added in version 4.0 that was released in March 2003. Conjugate heat transfer capability was added in version 5.0 which was released in February 2007. The main highlights of the present version (version 6.0) are: (1) extension of mixture option to handle phase change, (2) multiple pressure and flow regulators in a flow circuit, (3) fixed flow rate and the International System of Units option, and (4) two-dimensional flow modeling within a flow network system.

This Technical Memorandum provides a detailed discussion of the data structure, mathematical formulation, computer program, GUI and includes a number of example problems. Section 1 provides an introduction and overview of the code. Data structure of the code is described in section 2. The mathematical formulation that includes the description of governing equations and the solution procedure for solving the equations is described in section 3. The program structure is discussed in section 4. Section 5 describes GFSSP's GUI, which is called visual thermofluid dynamic analyzer for systems and components (VTASC). Several example problems are described in section 6. The new user may skip sections 2 to 4 initially, but will benefit from these sections after gaining some experience with the code.

EXECUTIVE SUMMARY

The Generalized Fluid System Simulation Program (GFSSP) is a general purpose computer program for analyzing steady state and time-dependent flow rates, pressures, temperatures, and concentrations in a complex flow network. The program is capable of modeling real fluids with phase changes, compressibility, mixture thermodynamics, conjugate heat transfer between solid and fluid, fluid transients, pumps, compressors, and external body forces such as gravity and centrifugal. The thermofluid system to be analyzed is discretized into nodes, branches, and conductors. The scalar properties such as pressure, temperature, and concentrations are calculated at nodes. Mass flow rates and heat transfer rates are computed in branches and conductors. The graphical user interface allows users to build their models using the 'point, drag, and click' method; the users can also run their models and post-process the results in the same environment.

Two thermodynamic property programs (GASP/WASP and GASPAK) provide required thermodynamic and thermophysical properties for 36 fluids: helium, methane, neon, nitrogen, carbon monoxide, oxygen, argon, carbon dioxide, fluorine, hydrogen, parahydrogen, water, kerosene (RP-1), isobutene, butane, deuterium, ethane, ethylene, hydrogen sulfide, krypton, propane, xenon, R-11, R-12, R-22, R-32, R-123, R-124, R-125, R-134A, R-152A, nitrogen trifluoride, ammonia, hydrogen peroxide, and air. The program also provides the options of using any incompressible fluid with constant density and viscosity or ideal gas. The users can also supply property tables for fluids that are not in the library.

Twenty-four different resistance/source options are provided for modeling momentum sources or sinks in the branches. These options include pipe flow, flow through a restriction, noncircular duct, pipe flow with entrance and/or exit losses, thin sharp orifice, thick orifice, square edge reduction, square edge expansion, rotating annular duct, rotating radial duct, labyrinth seal, parallel plates, common fittings and valves, pump characteristics, pump power, valve with a given loss coefficient, Joule-Thompson device, control valve, heat exchanger core, parallel tube, and compressible orifice. The program has the provision of including additional resistance options through User Subroutines.

GFSSP employs a finite volume formulation of mass, momentum, and energy conservation equations in conjunction with the thermodynamic equations of state for real fluids as well as energy conservation equations for the solid. The system of equations describing the fluid network is solved by a hybrid numerical method that is a combination of the Newton-Raphson and successive substitution methods. The application and verification of the code has been demonstrated through 25 example problems.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

APPENDICES G through HH (Supplemental Information on CD Inside Back Cover)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF ACRONYMS, SYMBOLS, AND ABBREVIATIONS

CFD	computational fluid dynamics
FVM	finite volume method
GFSSP	Generalized Fluid System Simulation Program
GOX	gaseous oxygen
GUI	graphical user interface
LOX	liquid oxygen
MOC	method of characteristics
MSFC	Marshall Space Flight Center (NASA)
RP-1	rocket propellent-1
SASS	simultaneous adjustment with successive substitution
SI	International System of Units
SIMPLE	semi-implicit pressure-linked equation
SP	solver and property
TM	Technical Memorandum
TPA	turbopump assembly
US	User Subroutine
VTASC	Visual Thermofluid dynamic Analyzer for Systems and Components

NOMENCLATURE

A	area (in^2)
A_0	pump characteristic curve coefficient
a	length (in)
a_{fluid}	speed of sound for LOX (ft/s)
B	passage width ($B = \pi D$)
B_0	pump characteristic curve coefficient
b	length (in)
C	heat capacity (Btu/s-°R)
C_c	convergence criterion
C_L	flow coefficient
C_p	specific heat (Btu/lb °F)
C_v	flow coefficient for a valve
c	concentration; clearance (in)
$c_{i,k}$	mass concentration of kth specie at ith node
D, d	diameter (in)
f	Darcy friction factor
G	bound mass (lb/ft^2-s)
g	gravitational acceleration (ft/s^2); specific heat ratio
g_c	conversion constant (= 32.174 lb-ft/lb$_f$-s^2)
h	enthalpy (Btu/lb)
h_{ij}	heat transfer coefficient (Btu/ft^2-s-°R)
J	mechanical equivalent of heat (778 ft-lb$_f$/Btu)
j	index of a neighboring node of the ith node
K, K_1	nondimensional head loss factor

K_e	exit loss coefficient
K_f	flow resistance coefficient $(lb_f\text{-}s^2/(lb\text{-}ft)^2$)
$K_{rotation}$	nondimensional rotating flow resistance coefficient
K_i	inlet loss coefficient
k	thermal conductivity (Btu/ft-s-°R)
k_v	empirical factor
L	length (in)
L^*	critical length of the pipe
L_Ω	resistance of the Joule-Thompson device
M	molecular weight
m	resident mass (lb); gap length (pitch)
$\dot{m}$	mass flow rate (lb/s)
m_p	pitch (in)
N	revolutions per minute (rpm); number of iterations
N_B	number of branches
N_E	number of equations
N_N	number of nodes
Nu	Nusselt number
n	number of teeth
P	pump power (hp)
Po	Poiseuille number
Pr	Prandtl number
p	pressure (lb_f /in^2)
$Q, \dot{q}$	heat source (Btu/s)
R	gas constant $(lb_f\text{-}ft/lb\text{-}°R)$
Re	Reynolds number $(Re = \rho uD/m)$
R_o	outer radius (in)
r	radius (in)
S	momentum source (lb_f)
$\dot{S}$	entropy generation (Btu/s-°R)

s	entropy (Btu/lb-°R)
T	fluid temperature (°F)
T_s	solid temperature (°F)
u_{ij}	velocity (ft/s)
$\dot{V}$	volumetic flow rate (ft^3/s)
V	volume (in^3)
V_f	viscosity correction factor
v	specific volume (ft^3/lb)
v_f	specific volume of a liquid (ft^3/lb)
v_g	specific volume of a gas (ft^3/lb)
w	Joule-Thomson device flow rate (lbm/hr); width
x	quality and mass fraction
x_k	molar concentrations
Y	two-phase factor in Miropolskii's equation
z	compressibility factor

Greek

α	multiplier for Labyrinth seal resistance; under-relaxation parameter
β	ratio of orifice diameter to inside pipe diameter
γ	specific heat
Δh	head loss (ft)
$\Delta \tau$	time step (s)
δ	distances between velocity locations (ft)
δ_{ij}	distance between two solid nodes (ft)
ε	absolute roughness (in), heat exchanger effectiveness, Labyrinth seal carryover factor; radius of orbit to the clearance
ε/D	relative roughness
ε_{ij}	emissivity
θ	angle between branch flow velocity vector and gravity vector (deg); angle between neighboring branches for computing shear (deg)
θ_r	time required to drain pressurized propellant tank (s)

η	efficiency
μ	viscosity (lb/ft-s)
ν	kinematic viscosity (ft^2/s)
ρ	density (lb/ft^3)
$\bar{\rho}$	molar density (lb-mol/ft^3)
σ	Stefan-Boltzman constant (= 4.7611×10^{-13} Btu/ft^2-R^4-s)
τ	time (s)
ω	angular velocity (rad/s); running speed (rpm)

Subscript

0	time equals zero
a	relating to inlet radius
amb	ambient
B	back
b	relating to outlet radius
c	cold
cr	critical
d	downstream
Dis	discharge
eff	effective
F	front
f	liquid
g	vapor
gen	generation
h	hot
Im	impeller
i	node; inner
ij	branch
k	index of the fluid component in a fluid mixture
l	liquid

NOMENCLATURE (Continued)

norm	normal
o	outer
or	orifice
p	parallel branch
prop	propellant
S	solid
s	entropy
trans	transverse
Turb	turbine
u	upstream branch
v	vapor

TECHNICAL MEMORANDUM

GENERALIZED FLUID SYSTEM SIMULATION PROGRAM, VERSION 6.0

1. INTRODUCTION

The need for a generalized computer program for thermofluid analysis in a flow network has been felt for a long time in aerospace industries. Designers of thermofluid systems often need to know pressures, temperatures, flow rates, concentrations, and heat transfer rates at different parts of a flow circuit for steady state or transient conditions. Such applications occur in propulsion systems for tank pressurization, internal flow analysis of rocket engine turbopumps, chilldown of cryogenic tanks and transfer lines, and many other applications of gas-liquid systems involving fluid transients and conjugate heat and mass transfer. Computer resource requirements to perform time-dependent, three-dimensional Navier-Stokes computational fluid dynamic (CFD) analysis of such systems are prohibitive and therefore are not practical. A possible recourse is to construct a fluid network consisting of a group of flow branches such as pipes and ducts that are joined together at a number of nodes. They can range from simple systems consisting of a few nodes and branches to very complex networks containing many flow branches simulating valves, orifices, bends, pumps, and turbines. In the analysis of existing or proposed networks, node pressures, temperatures, and concentrations at the system boundaries are usually known. The problem is to determine all internal nodal pressures, temperatures, concentrations, and branch flow rates. Such schemes are known as network flow analysis methods and they use largely empirical information to model fluid friction and heat transfer. For example, an accurate prediction of axial thrust in a liquid rocket engine turbopump requires the modeling of fluid flow in a very complex network. Such a network involves the flow of cryogenic fluid through extremely narrow passages, flow between rotating and stationary surfaces, phase changes, mixing of fluids, and heat transfer. Propellant feed system designers are often required to analyze pressurization or blowdown processes in flow circuits consisting of many series and parallel flow branches containing various pipe fittings and valves using cryogenic fluids. The designers of a fluid system are also required to know the maximum pressure in the pipeline after sudden valve closure or opening.

Available commercial codes are generally suitable for steady state, single-phase incompressible flow. Because of the proprietary nature of such codes, it is not possible to extend their capability to satisfy the above-mentioned needs. In the past, specific purpose codes were developed to model the space shuttle main engine turbopump. However, it was difficult to use those codes for a new design without making extensive changes in the original code. Such efforts often turn out to be time consuming and inefficient. Therefore, the Generalized Fluid System Simulation Program (GFSSP)[1] has been developed at NASA Marshall Space Flight Center (MSFC) as a general fluid flow system solver capable of handling phase changes, compressibility, mixture thermodynamics,

and transient operations. It also includes the capability to model external body forces such as gravity and centrifugal effects in a complex flow network. The objective of the present effort is to (a) develop a robust and efficient numerical algorithm to solve a system of equations describing a flow network containing phase changes, mixing, and rotation, and (b) to implement the algorithm in a structured, easy-to-use computer program.

This program requires that the flow network be resolved into nodes and branches. The program's preprocessor allows the user to interactively develop a fluid network simulation consisting of fluid nodes and branches, solid nodes, and conductors. In each branch, the momentum equation is solved to obtain the flow rate in that branch. At each fluid node, the conservation of mass, energy, and species equations are solved to obtain the pressures, temperatures, and species concentrations at that node. At each solid node, the energy conservation equation is solved to calculate temperature of the solid.

This Technical Memorandum (TM) documents the data structure, mathematical formulation, computer program, and graphical user interface (GUI). Use of the code is illustrated by 25 example problems. It also documents the verification and validation effort conducted by code developers and users. This section also presents an overview of the subsequent sections to provide users with a global perspective of the code.

1.1 Network Flow Analysis Methods

The oldest method for systematically solving a problem consisting of steady flow in a pipe network is the Hardy Cross method.[2] Not only is this method suited for hand calculations, but it has also been widely employed for use in computer-generated solutions. But as computers allowed much larger networks to be analyzed, it became apparent that the convergence of the Hardy Cross method might be very slow or even fail to provide a solution in some cases. The main reason for this numerical difficulty is that the Hardy Cross method does not solve the system of equations simultaneously. It considers a portion of the flow network to determine the continuity and momentum errors. The head loss and the flow rates are corrected and then it proceeds to an adjacent portion of the circuit. This process is continued until the whole circuit is completed. This sequence of operations is repeated until the continuity and momentum errors are minimized. It is evident that the Hardy Cross method belongs in the category of successive substitution methods and it is likely that it may encounter convergence difficulties for large circuits. In later years, the Newton-Raphson method has been utilized[3] to solve large networks. The Newton-Raphson method solves all the governing equations simultaneously and is numerically more stable and reliable than successive substitution methods.

The network analysis method[4] has been widely used in thermal analysis codes (SINDA/G[5] and SINDA/FLUINT[6]) using an electric analog. The partial differential equation of heat conduction is discretized into finite difference form expressing temperature of a node in terms of temperatures of neighboring nodes and ambient nodes. The set of finite difference equations is solved to calculate temperature of the solid nodes and heat fluxes between the nodes. There have been some limited applications of thermal network analysis methods to model fluid flows. Such attempts did not go far because of the inability of heat conduction equations to handle the nonlinear fluid

inertia term. There has been limited success in modeling compressible and two-phase flows by such methods.

At MSFC, another system analysis code, ROCETS[7] is routinely used for simulating flow in rocket engines. ROCETS has a very flexible architecture where users develop the system model by integrating component modules such as pumps, turbines, and valves. The user can also build any model of specific components to integrate into the system model. ROCETS solves the system of equations by a modified Newton-Raphson method.[8]

The finite volume method (FVM) has been widely used in solving Navier-Stokes equations in CFD.[9] The FVM divides the flow domain into a discrete number of control volumes and determines the conservation equations for mass, momentum, energy, and species for each control volume. Simultaneous solutions of these conservation equations provide the pressure, velocity components, temperature, and concentrations representative of the discrete control volumes. The numerical method is called 'pressure based' if the pressures are calculated from the mass conservation equation and density from the equation of state. On the other hand, a 'density based' numerical method uses the mass conservation equation to calculate density of the fluid and pressure from the equation of state. GFSSP uses a pressure-based FVM as the foundation of its numerical scheme.

1.1.1 Network Definitions

GFSSP constructs a fluid network using fluid and solid nodes. The fluid circuit is constructed with boundary nodes, internal nodes, and branches (fig. 1) while the solid circuit is constructed with solid nodes, ambient nodes, and conductors. The solid and fluid nodes are connected with solid-fluid conductors. Users must specify conditions such as pressure, temperature, and concentration of species at the boundary nodes. These variables are calculated at the internal nodes by solving conservation equations of mass, energy, and species in conjunction with the thermodynamic equation of state. Each internal node is a control volume where there is inflow and outflow of mass, energy, and species at the boundaries of the control volume. The internal node also has resident mass, energy, and concentration. The momentum conservation equation is expressed in flow rates and is solved in branches. At the solid node, the energy conservation equation for solid is solved to compute temperature of the solid node. Figure 1 shows a schematic and GFSSP flow circuit of a counter flow heat exchanger. Hot nitrogen gas is flowing through a pipe, colder nitrogen is flowing counter to the hot stream in the annulus pipe, and heat transfer occurs through metal tubes. The problem considered is to calculate flow rates and temperature distributions in both streams.

1.2 Units and Sign Conventions

GFSSP uses British gravitational units (commonly known as engineering units). Table 1 describes the units of variables used in the code. The units in the second column are the units that appear in the input and output data files. Users must specify the values in these units in their model. The units that are listed in the third column are internal to the code and used during the solution of the equations. These units must be used in user-provided subroutines. The user has
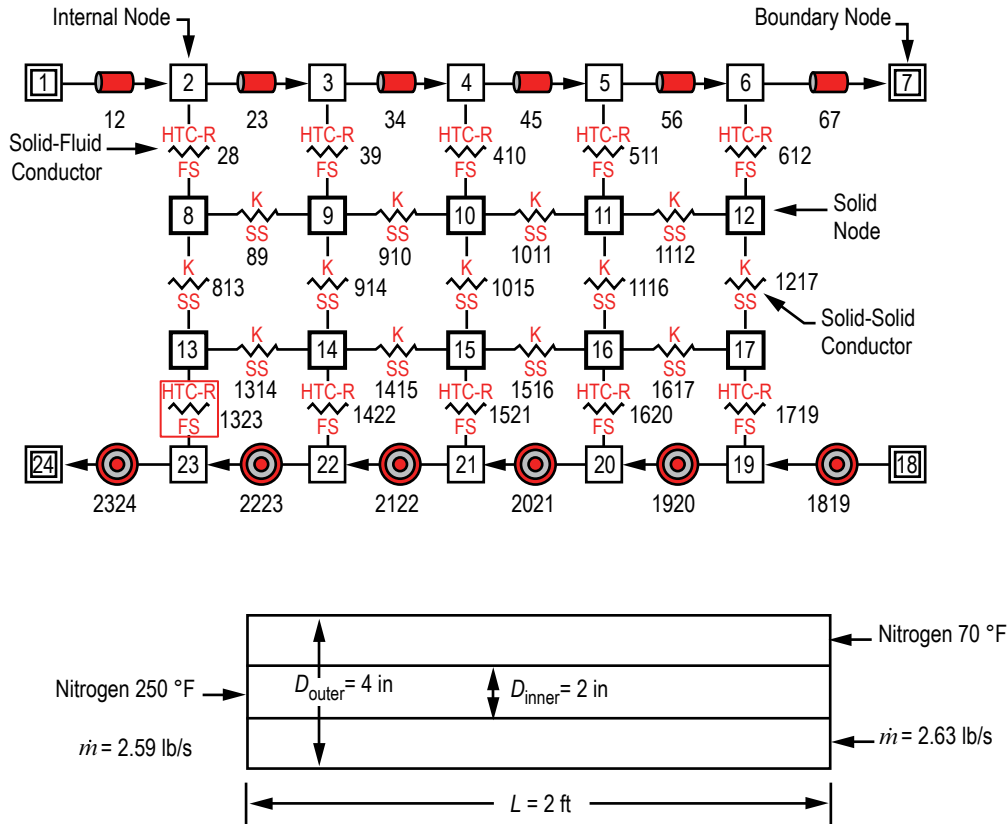
Figure 1. A typical flow network consists of fluid node, solid node, flow branches, and conductors.

Table 1. Units of variables in input/output and solver module.

Variables	Input/Output	Solver Module
Length	inches	feet
Area	inches^2	feet^2
Pressure	psia	psf
Temperature	°F	°R
Mass injection	lbm/s	lbm/s
Heat source	Btu/s or Btu/lbm	Btu/s or Btu/lbm

the option of entering the International System of Units (SI) into the preprocessor; these will be converted to engineering units when written to the input file (see sec. 5.1.3). GFSSP uses standard sign conventions for mass and heat transfer. Mass and heat input to a node is considered positive. Similarly mass and heat output from a node is considered negative.

1.3 Data Structure

GFSSP has a unique data structure (fig. 2) that allows constructing all possible arrangements of a flow network with no limit on the number of elements. The elements of a flow network are boundary nodes, internal nodes, and branches. For conjugate heat transfer problems, there are three additional elements: solid node, ambient node, and conductor. The relationship between a fluid node and a branch as well as a solid node and conductor is defined by a set of relational geometric properties. For example, the relational geometric properties of a node are number and name of branches connected to it. With the help of these properties, it is possible to define any structure of the network as it progresses through every junction of the network. The positive or negative flow direction is also defined locally. Unlike a structured coordinate system, there is no global definition of flow direction and origin. The development of a flow network can start from any point and can proceed in any direction.
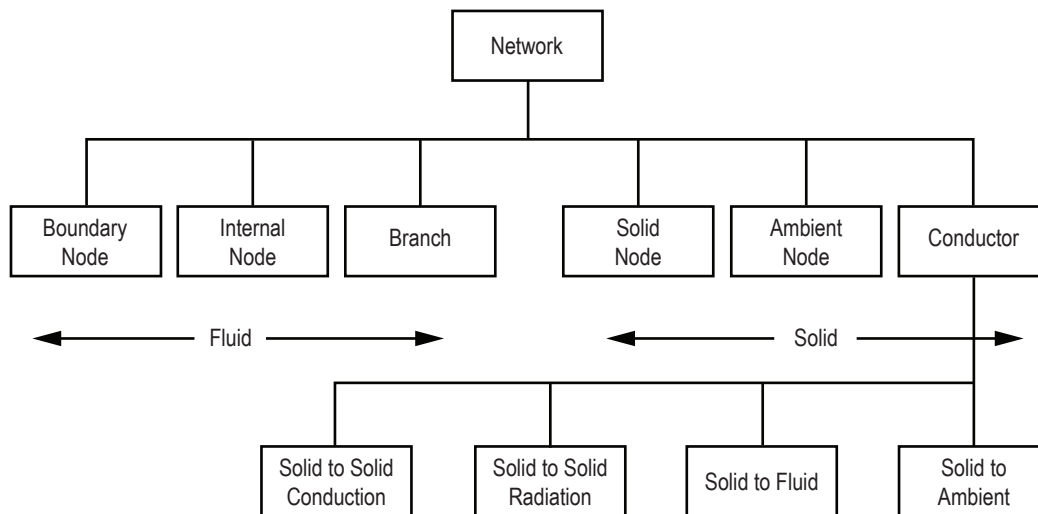
Figure 2. Data structure of the fluid-solid network has six major elements.

All elements of a network have properties. The properties can be classified into two categories: geometric and thermofluid. Geometric properties are again classified into two subcategories: relational and quantitative. Relational properties define the relationship of the element with the neighboring elements. Quantitative properties include geometric parameters such as area, length, and volume. GFSSP's data structure is discussed in detail in section 2.

1.4 Mathematical Formulation

GFSSP solves the conservation equations of mass and momentum in internal nodes and branches to calculate fluid properties. It also solves for energy conservation equations to calculate temperatures of solid nodes. Table 2 shows the mathematical closure that describes the unknown variables and the available equations to solve the variables. Pressure, temperature, species concentration, and resident mass in a control volume are calculated at the internal nodes, whereas the flow

Table 2. Mathematical closure.

Unknown Variables	Available Equations to Solve
Pressure	Mass conservation equation
Flow rate	Momentum conservation equation
Fluid temperature	Energy conservation equation of fluid
Solid temperature	Energy conservation equation of solid
Species concentrations	Conservation equations for species
Fluid mass (unsteady flow)	Thermodynamic equation of state

rate is calculated at the branch. The equations are coupled and nonlinear; therefore, they are solved by an iterative numerical scheme. GFSSP employs a unique numerical scheme known as simultaneous adjustment with successive substitution (SASS), which is a combination of Newton-Raphson and successive substitution methods. The coupling of equations is shown in figure 3. The mass and momentum conservation equations and the equation of state are solved by the Newton-Raphson method while the conservation of energy and species are solved by the successive substitution method.
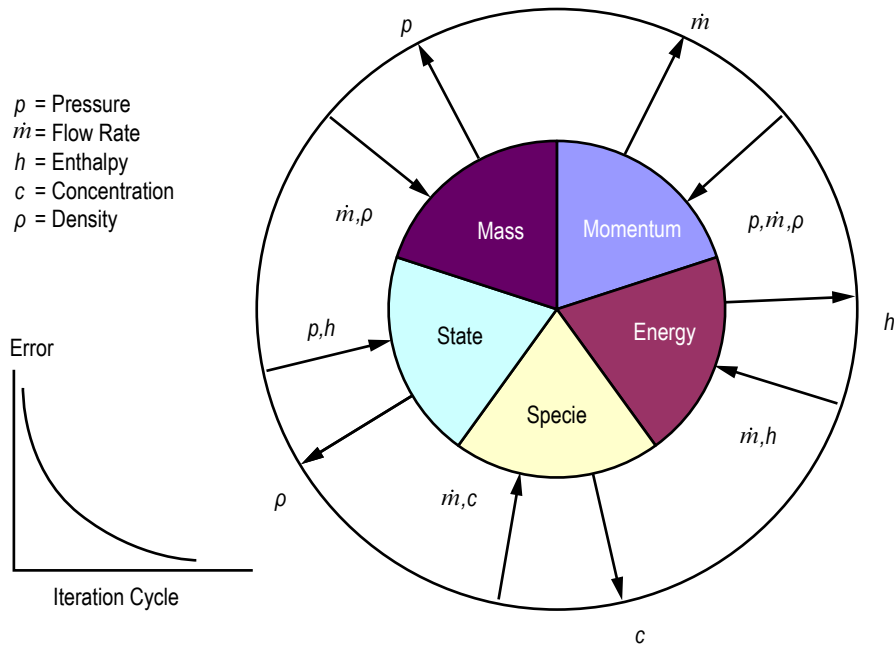
Figure 3. Schematic of mathematical closure of GFSSP—coupling of thermodynamics and fluid dynamics.

The total number of equations to be solved is determined from the number of internal nodes and branches. Figure 4 shows a typical interpropellant flow circuit in a rocket engine turbopump. In this circuit there are five boundary nodes and seven internal nodes. These nodes are connected by 12 branches. There are three inlet boundary nodes (48, 66, and 22) where oxygen, helium, and hydrogen enter into the fluid circuit. Mixtures of helium-hydrogen and helium-oxygen

exit the circuit through boundary nodes numbered 50 and 16, respectively. At each internal node four equations are solved to calculate pressure, temperature, and two concentrations. It should be noted that in a mixture of three components, concentrations of two components are solved. The concentration of the third component is determined from the fact that the sum of all concentrations must be unity. Flow rates are calculated in 12 branches. Therefore, GFSSP solves for $40\,(= 7 \times 4 + 12)$ equations to calculate all required variables in the circuit. Since the example problem is at steady state, resident mass in the control volume was not calculated. The transient model of the same fluid circuit would require solution of $47\,(= 7 \times 5 + 12)$ equations at each time step of the simulation. The mathematical formulation has been described in detail in section 3.
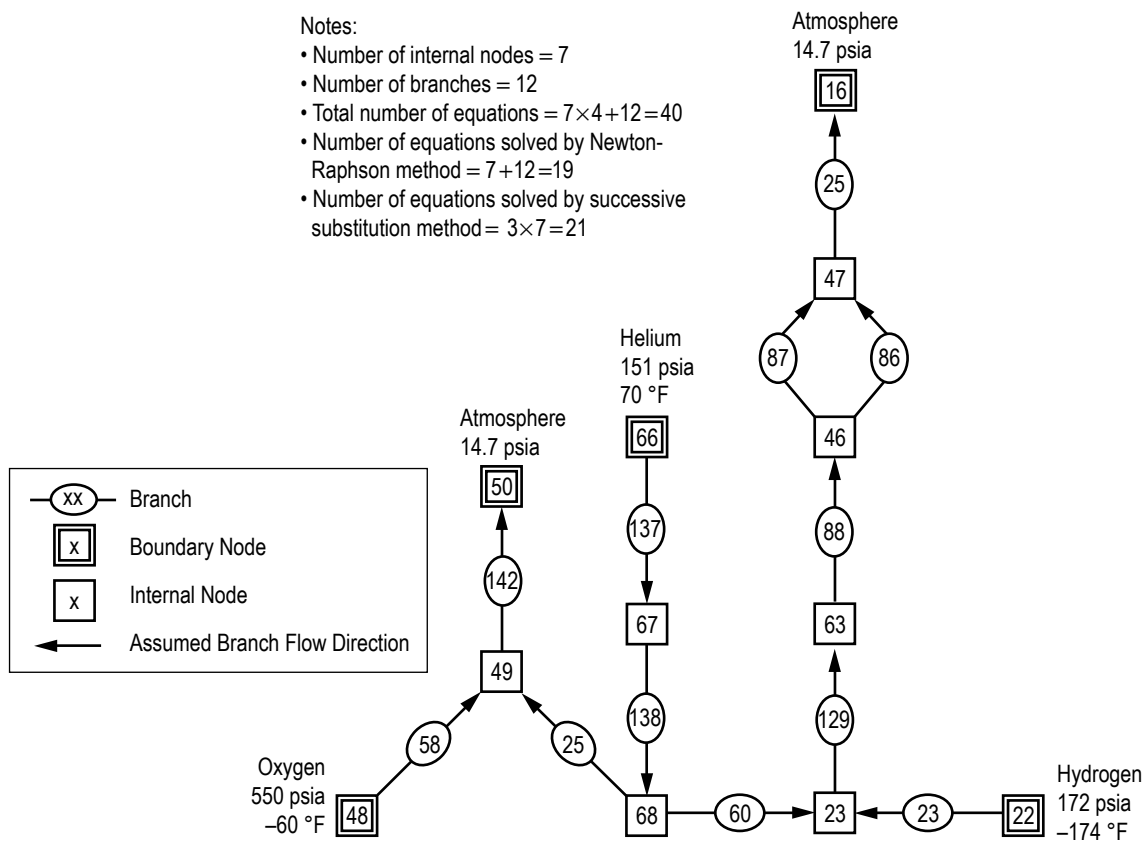
Figure 4. Interpropellant seal flow circuit in a rocket engine turbopump.

1.5 Fluid Properties

GFSSP is linked with two thermodynamic property programs, GASP[10] and WASP[11] and GASPAK,[12] that provide thermodynamic and thermophysical properties of selected fluids. Both programs cover a range of pressure and temperature that allows fluid properties to be evaluated for liquid, liquid-vapor (saturation), and vapor region. GASP and WASP provide properties of 12 fluids (table 3). GASPAK includes a library of 36 fluids (table 4).

Table 3. Fluids available in GASP and WASP.

Index	Fluid
1	Helium
2	Methane
3	Neon
4	Nitrogen
5	Carbon monoxide
6	Oxygen
7	Argon
8	Carbon dioxide
9	Fluorine
10	Hydrogen
11	Water
12	RP-1

Table 4. Fluids available in GASPAK.

Index	Fluid	Index	Fluid
1	Helium	19	Krypton
2	Methane	20	Propane
3	Neon	21	Xenon
4	Nitrogen	22	R-11
5	Carbon monoxide	23	R-12
6	Oxygen	24	R-22
7	Argon	25	R-32
8	Carbon dioxide	26	R-123
9	Parahydrogen	27	R-124
10	Hydrogen	28	R-125
11	Water	29	R-134A
12	RP-1	30	R-152A
13	Isobutane	31	Nitrogen Trifloride
14	Butane	32	Ammonia
15	Deuterium	33	Ideal gas
16	Ethane	34	Hydrogen peroxide
17	Ethylene	35	Air
18	Hydrogen sulfide		

1.6 Flow Resistances

In network flow analysis code, flow resistances are modeled by empirical laws. These empirical laws have been incorporated to model flow resistances for pipe flow, orifices, valves, and various pipe fittings. GFSSP models these flow resistances in the momentum conservation equation as a friction term. There are 24 different resistance options available to users to choose from. There is also a provision for introducing a new resistance option through User Subroutines. The available resistance options are shown in table 5.

Table 5. Resistance options in GFSSP.

Option	Type of Resistance	Input Parameters	Option	Type of Resistance	Input Parameters
1	Pipe flow	L (in), D (in), ε/D	13	Common fittings and valves (two K method)	D (in), K_1, K_2
2	Flow through restriction	C_L, A (in^2)	14	Pump characteristics*	A_0, B_0, C_0, A (in^2)
3	Noncircular duct	a (in), b (in)	15	Pump power	P (hp), η, A (in^2)
4	Pipe with entrance and exit loss	L (in), D (in), ε/D, K_i, K_e	16	Valve with given C_v	C_v, A
5	Thin, sharp orifice	D_1 (in), D_2 (in)	17	Joule-Thompson device	L_Ω, V_f, k_v, A
6	Thick orifice	L (in), D_1 (in), D_2 (in)	18	Control valve	See example 12 data file
7	Square reduction	D_1 (in), D_2 (in)	19	User defined	A (in^2)
8	Square expansion	D_1 (in), D_2 (in)	20	Heat exchanger core	A_f (in^2), A_s (in^2), A_c (in^2), L (in), K_c, K_e
9	Rotating annular duct	L (in), r_o (in), r_i (in), N (rpm)	21	Parallel tube	L (in), D (in), ε/D, n
10	Rotating radial duct	L (in), D (in), N (rpm)	22	Compressible orifice	C_L, A (in^2)
11	Labyrinth seal	r_i (in), c (in), m (in), n,	23	Labyrinth seal, Egli correlation	r_i (in), c (in), m (in), n, α
12	Flow between parallel plates	r_i (in), c (in), L (in)	24	Fixed flow	Flow (lb$_m$/s), A (in^2)

*Pump characteristics are expressed as $\Delta p = A_0 + B_0 \dot{m} + C_0 \dot{m}^2$, Δp – Pressure rise, lbf/ft^2, $\dot{m}$ – Flow rate, lbm/s.

1.7 Program Structure

GFSSP has three major parts (fig. 5). The first part is the GUI, visual thermofluid analyzer of systems and components (VTASC). VTASC allows users to create a flow circuit by a 'point and click' paradigm. It creates the GFSSP input file after the completion of the model building process.
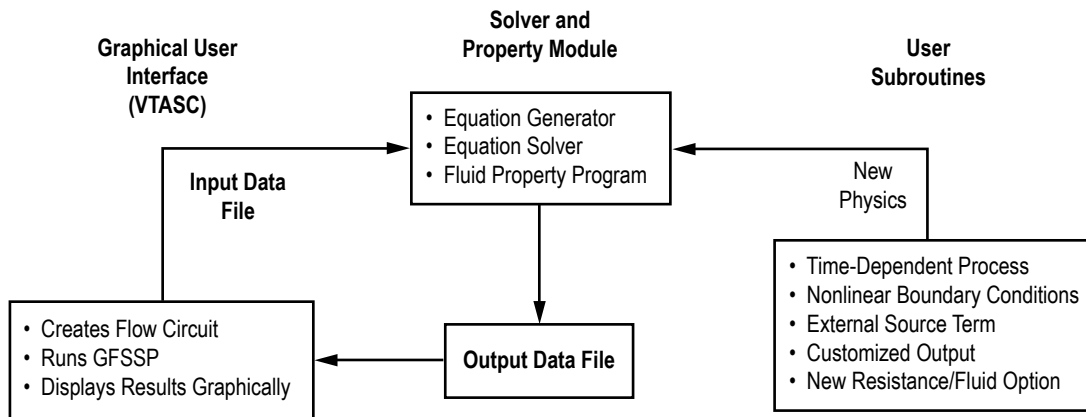
Figure 5. GFSSP's program structure showing the interaction of three major modules.

It can also create a customized GFSSP executable by compiling and linking User Subroutines with the solver module of the code. Users can run GFSSP from VTASC and post process the results in the same environment. The second major part of the program is the solver and property module. This is the heart of the program that reads the input data file, and generates the required conservation

equations for all internal nodes and branches with the help of thermodynamic property data. It also interfaces with User Subroutines to receive any specific inputs from users. Finally, output files are created for VTASC to read and display results. The User Subroutine is the third major part of the program. This consists of several blank subroutines that are called by the Solver Module. These subroutines allow the users to incorporate any new physical model, resistance option, fluid, etc. in the model. The computer program is discussed in detail in section 4.

1.8 Graphical User Interface

GFSSP's GUI (fig. 6) provides the users a platform to build and run their models. It also allows post-processing of results. The network flow circuit is first built using three basic elements: boundary node, internal node, and branch. Then the properties of the individual elements are assigned. Users are also required to define global options of the model that includes input/output files, fluid specification, and any special options such as rotation, heat exchanger, etc. During execution of the program, a run manager window opens up and users can monitor the progress of the numerical solution. On the completion of the run, it allows users to visualize the results in tabular form for steady state solutions and in graphical form for unsteady solutions. It also provides an interface to activate and import data to the plotting program, Winplot[13] for post-processing. The GUI is discussed in detail in section 5.
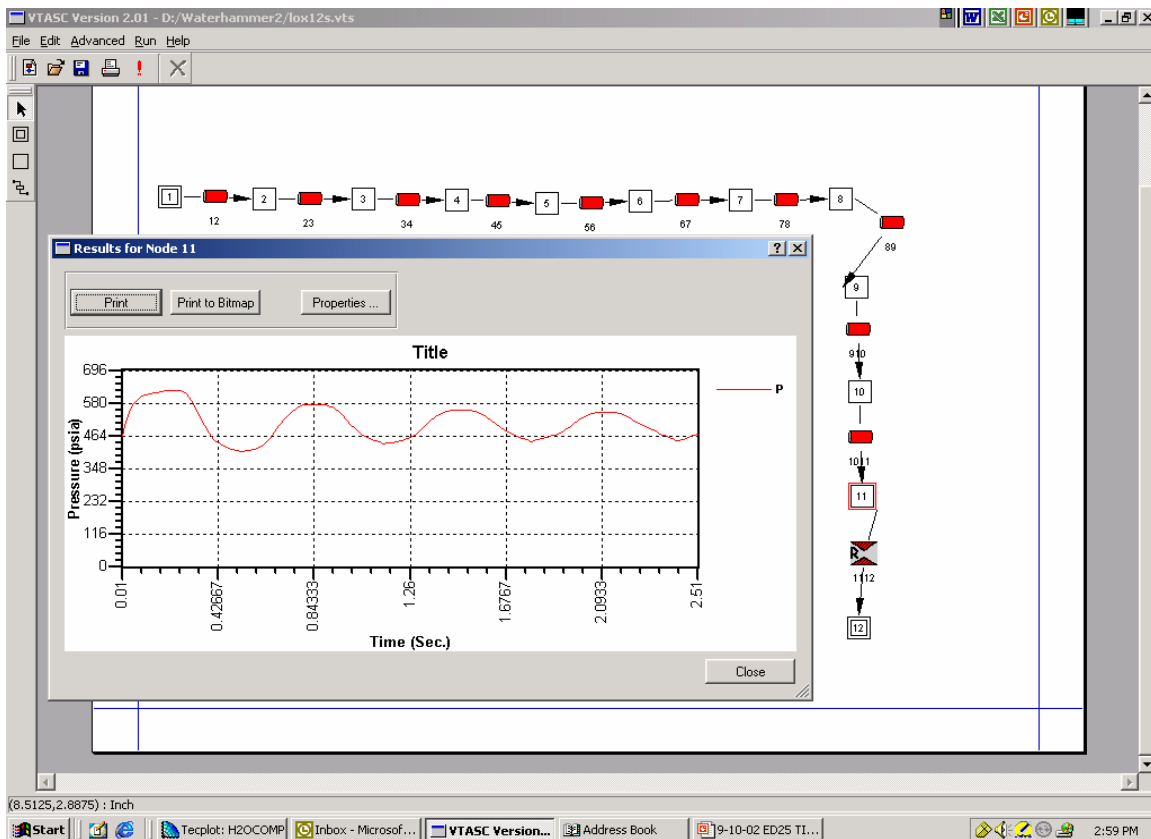
Figure 6. GFSSP's graphical user interface, VTASC allows creating, running, and viewing results in one environment.

1.9 Example Problems

Several example problems have been included to aid users to become familiar with different options of the code. The example problems also provide the verification and validation of the code by comparing the code's predictions with analytical solution and experimental data. These examples include the following:

(1) Simulation of a flow system containing a pump, valve, and pipeline
(2) Flow network for a water distribution system
(3) Compressible flow in a converging-diverging nozzle
(4) Mixing of combustion gases and a cold gas stream
(5) Flow in a counter flow heat exchange
(6) Radial flow in a rotating radial disk
(7) Flow in a squeeze film damper
(8) Blowdown of a pressurized tank
(9) A reciprocating-piston cylinder
(10) Pressurization of a propellant tank
(11) Power balancing of a turbopump assembly
(12) Helium pressurization of liquid oxygen (LOX) and rocket propllent-1 (RP-1) propellent tanks
(13) Steady state conduction through a circular rod
(14) Chilldown of cryogenic transfer line
(15) Fluid transient (waterhammer) due to sudden valve closure
(16) Simulation of pressure regulator downstream of a pressurized tank
(17) Simulation of flow regulator downstream of a pressurized tank
(18) Subsonic Fanno flow
(19) Subsonic Rayleigh flow
(20) Modeling of closed-cycle liquid metal (lithium) loop with heat exchanger to heat helium gas
(21) Internal flow in a turbopump
(22) Simulation of a fluid network with fixed flow rate option
(23) Helium-assisted, buoyancy-driven flow in a vertical pipe carrying LOX with ambient heat leak
(24) Simulation of relief valve in a pressurized tank
(25) Two-dimensional recirculating flow in a driven cavity.

These example problems are discussed in detail in section 6.

2. DATA STRUCTURE

Conventional CFD codes generally use a structured coordinate system to express conservation equations for mass, momentum, and energy. The examples of structured coordinate system are rectangular cartesian, cylindrical polar, and spherical polar. In these coordinate systems, each control volume has a fixed number of neighboring control volumes with which it exchanges mass, momentum, and energy. In one dimension, each control volume has two neighbors; in two and three dimensions, it has four and six neighbors, respectively. However, in network flow analysis, a control volume can have an arbitrary number of neighbors as shown in figure 7. Therefore the network analysis code requires a unique data structure that allows each control volume to know its neighbors. This is achieved by introducing relational properties for each control volume.
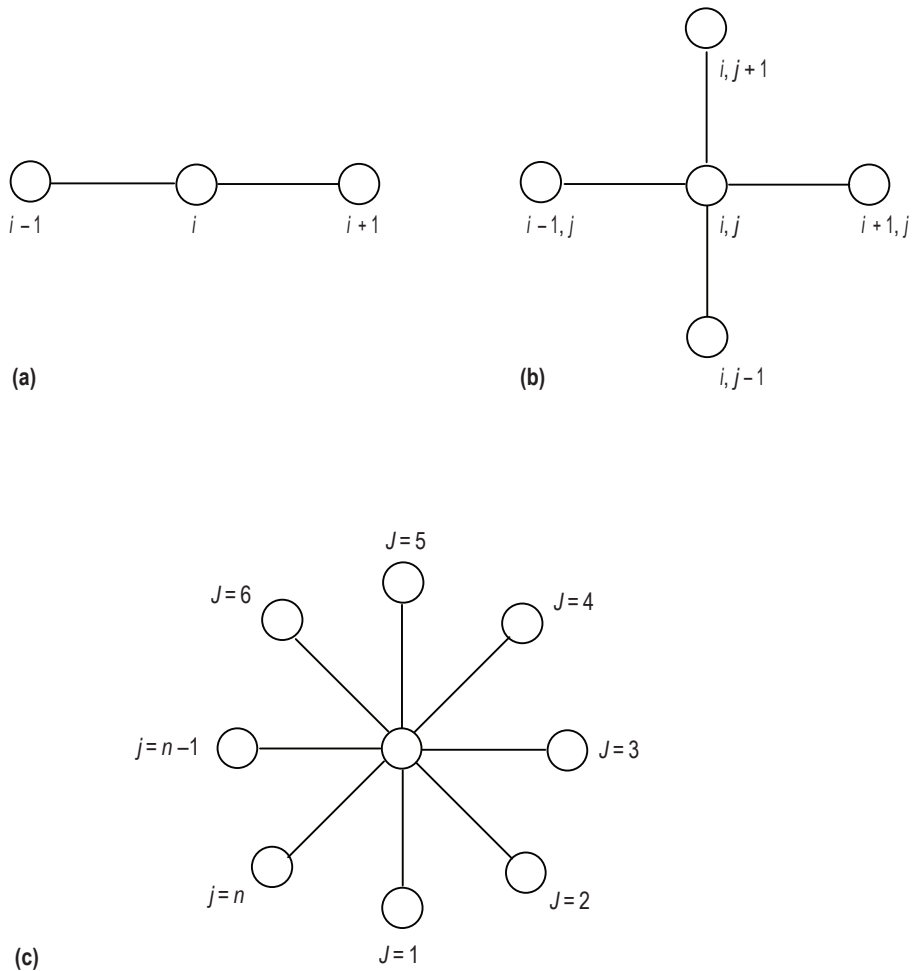
Figure 7. Examples of structured and unstructured coordinate systems: (a) One-dimensional structured coordinate, (b) two-dimensional structured coordinate, and (c) unstructured coordinate to represent flow network.

GFSSP has a unique data structure that allows constructing all possible arrangements of a flow network with no limit of number of elements. The elements of a flow network are fluid nodes and branches, solid nodes, and conductors. The relationship between a fluid node and a branch is defined by a set of relational geometric properties. Similarly, the relationship between a solid node and conductors is defined by a set of relational geometric properties. The connection between solid and fluid nodes for analyzing conjugate heat transfer is also defined by a set of relational properties. With the help of these properties, it is possible to define any structure of the network as it progresses through every junction of the network. The positive or negative flow direction is also defined locally. Unlike a structured coordinate system, there is no global definition of flow direction and origin. The development of a flow network can start from any point and can proceed in any direction. This section describes the data structure used to develop the governing equations to be described in section 3.

2.1 Network Elements and Properties

GFSSP constructs a flow network with three basic elements: (1) Boundary node, (2) internal node, and (3) branch. Thermodynamic states such as pressure, temperature, and species concentrations are assigned in boundary nodes. At internal nodes, GFSSP calculates all thermofluid dynamic variables such as pressure, temperature, enthalpy, entropy, species concentration, and thermophysical properties such as viscosity and conductivity. Flow rate and velocity are calculated in branches. A typical flow network consisting of a boundary node, internal node, and branch are shown in figure 1.

All nodes and branches are numbered arbitrarily by the user. GFSSP, however, assigns an index number to each node and branch as the user creates a new node or branch to construct a flow circuit. For example NODE(I) represents the node number where I is the pointer of the NODE array. As nodes are created, additional pointers are added to the array. Similarly, IBRANCH(I) represents the branch number where I is the pointer of the IBRANCH array. INDEX(I) defines the type of node. For an internal node, INDEX(I) = 1, whereas for a boundary node, INDEX(I) = 2. The internal node numbers are also designated as INODE(I), where index I ranges from 1 to the total number of internal nodes.

Conjugate heat transfer modeling requires extension of the fluid network to include a network of solid nodes with interfaces between solid and fluid nodes. With this interface, convective and radiation heat transfer between solid and fluid node is modeled. Three additional elements—solid nodes, ambient nodes, and conductors—become part of the integrated network.

All elements have properties. The properties can be classified into two categories: geometric and thermofluid (fig. 2). Geometric properties are again classified into two subcategories: relational and quantitative. Relational properties define the relationship of the element with the neighboring elements. Quantitative properties include geometric parameters such as area, length, and volume.

2.2 Internal and Boundary Node Thermofluid Properties

The thermofluid properties (fig. 8) of internal and boundary nodes are:

• Pressure
• Temperature
• Density
• Species concentration
• Enthalpy
• Entropy
• Gas constant
• Viscosity
• Conductivity
• Specific heat ratio.

For unsteady flow, each internal node also includes thermofluid properties at the previous time step.
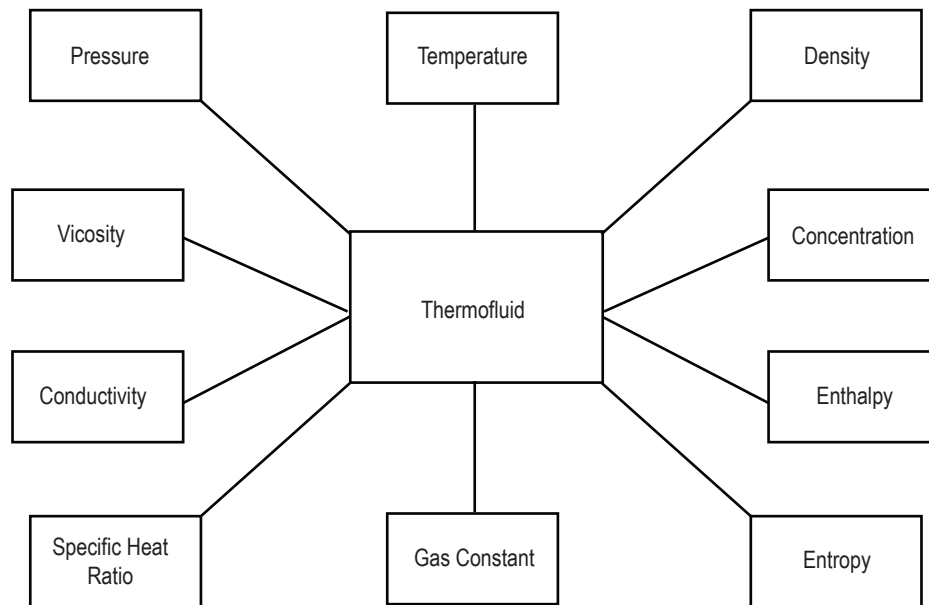
Figure 8. Thermofluid properties of internal and boundary nodes.

2.3 Internal Node Geometric Properties

The internal node has geometric properties of two kinds: relational and quantitative. The relational geometric properties of an internal node are:

• NUMBR(I), which defines the number of branches connected to the node of index I.

• NAMEBR(I,J), which defines the name of branch connected to node with index I; the index J extends from 1 to the number of branches connected to the node I, stored in NUMBR(I).

The quantitative geometric property of an internal node is node volume, which is necessary to calculate resident mass for unsteady calculation. The resident mass that determines the capacitance of the node is not required for steady state calculations. The data structure of geometric properties of an internal node is shown in figure 9. Figure 10 shows an example of relational geometric property of a node. Following are the relational geometric properties of node 1:

Number of branches connected to node I, NUMBR(I) = 4
Name of the branches connected to node I,
NAMEBR(I, 1) = 31
NAMEBR(I, 2) = 41
NAMEBR(I, 3) = 51
NAMEBR(I, 4) = 12.
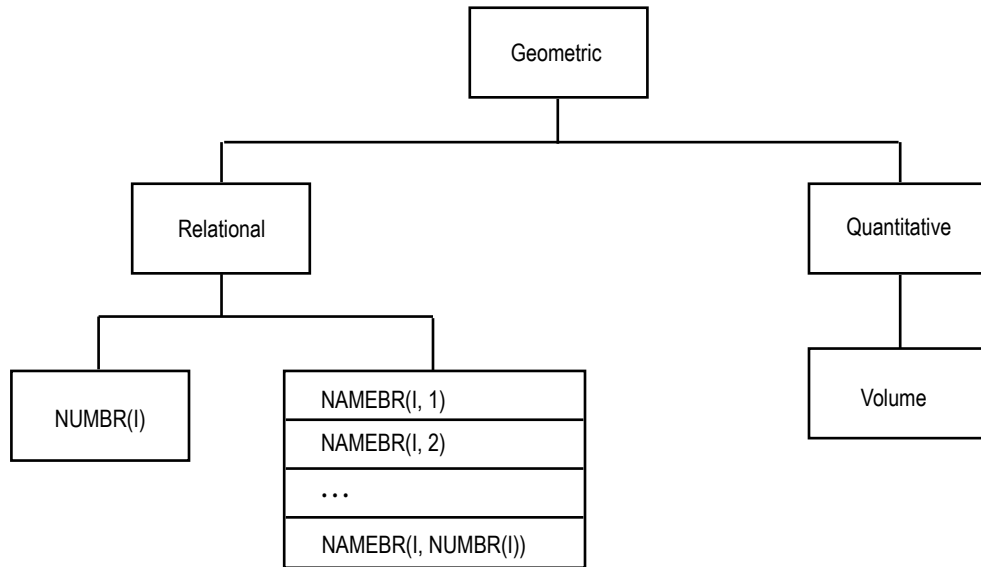
Figure 9. Data structure of geometric property of an internal node.
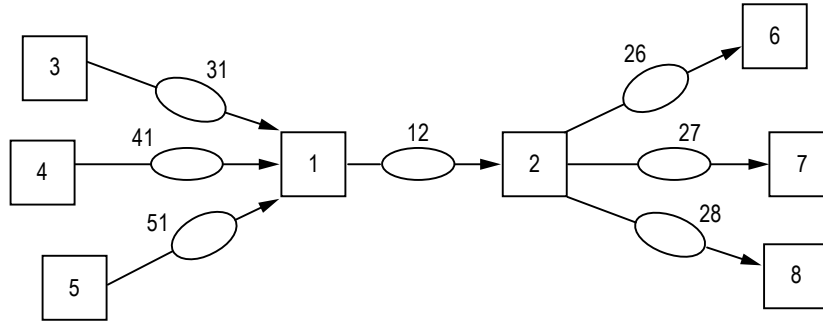
Figure 10. **E**xample of node and branch relational properties.

2.4 Branch Properties

The branch has geometric properties of both kinds: relational and quantitative. The relational geometric properties of a branch are as follows:

• Name of upstream node
• Name of downstream node
• Number of upstream branches
• Name of upstream branches
• Number of downstream branches
• Name of downstream branches
• Index number of resistance option.

Figure 11 shows the geometric relational property of a branch. An example of those properties in a typical flow network is shown in figure 10. Each relational property of branch number 12 (IBRANCH(I) = 12) in figure 10 is now defined:

Name of upstream node, IBRUN (I) = 1
Number of upstream branches, NOUBR (I) = 3
Name of upstream branches,
NMUBR(I, 1) = 31
NMUBR(I, 2) = 41
NMUBR(I, 3) = 51

Name of downstream node, IBRDN(I) = 2
Number of downstream branches, NODBR(I) = 3
Name of downstream branches,
NMDBR(I, 1) = 26
NMDBR(I, 2) = 27
NMDBR(I, 3) = 28.

16
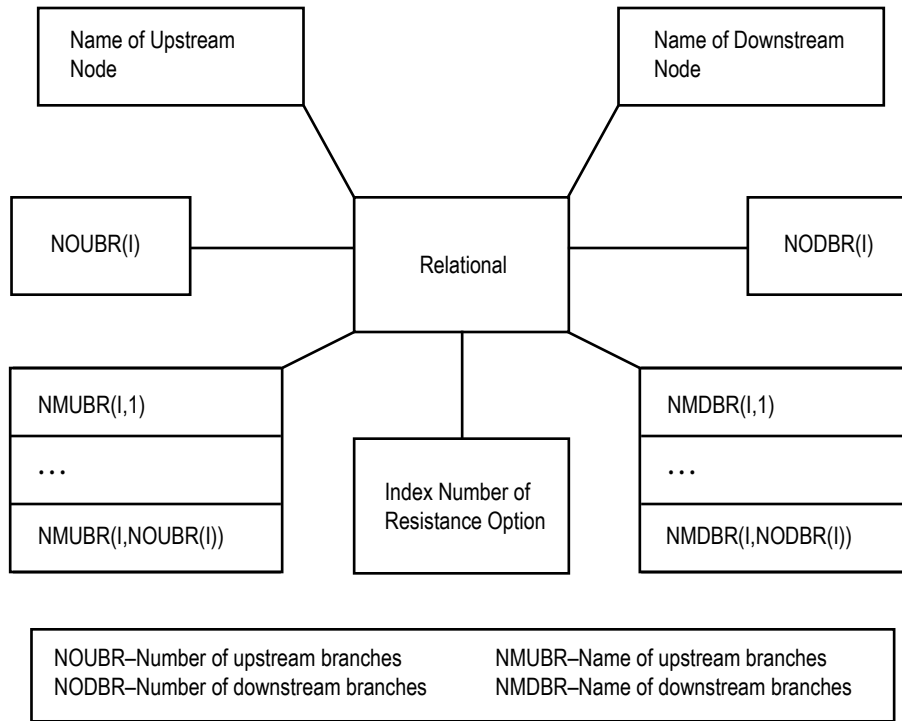
Figure 11. Relational geometric properties of a branch.

The quantitative geometric properties of a branch are:

• Area
• Volume
• Radial distance of upstream node from the axis of rotation
• Radial distance of downstream node from the axis of rotation
• Rotational speed of the branch
• Six additional generic geometric parameters to characterize a given resistance option.

The thermofluid properties of a branch (fig. 12) are:

• Flow rate
• Velocity
• Resistance coefficient.

For unsteady flow each branch also, includes the quantitative geometric and thermofluid dynamic properties at the previous time step.
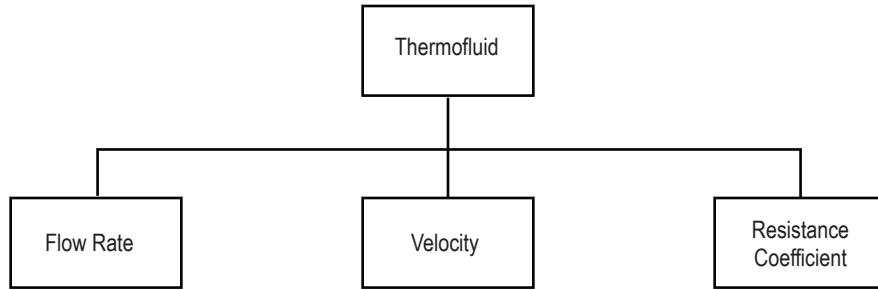
Figure 12. Thermofluid properties of a branch.

2.5 Fluid-Solid Network for Conjugate Heat Transfer

In fluid-solid network for conjugate heat transfer, solid nodes, ambient nodes, and conductors for heat transfer become part of the GFSSP network. Network elements for conjugate heat transfer are shown in figure 13. There are four types of conductors: solid to solid conduction, solid to solid radiation, solid to fluid, and solid to ambient. A typical GFSSP network for conjugate heat transfer is shown in figure 14. A solid node can be connected to a fluid node and ambient node. To determine solid temperature, conduction, convection, and radiation heat transfer between solid-solid, solid-fluid, and solid-ambient are computed.
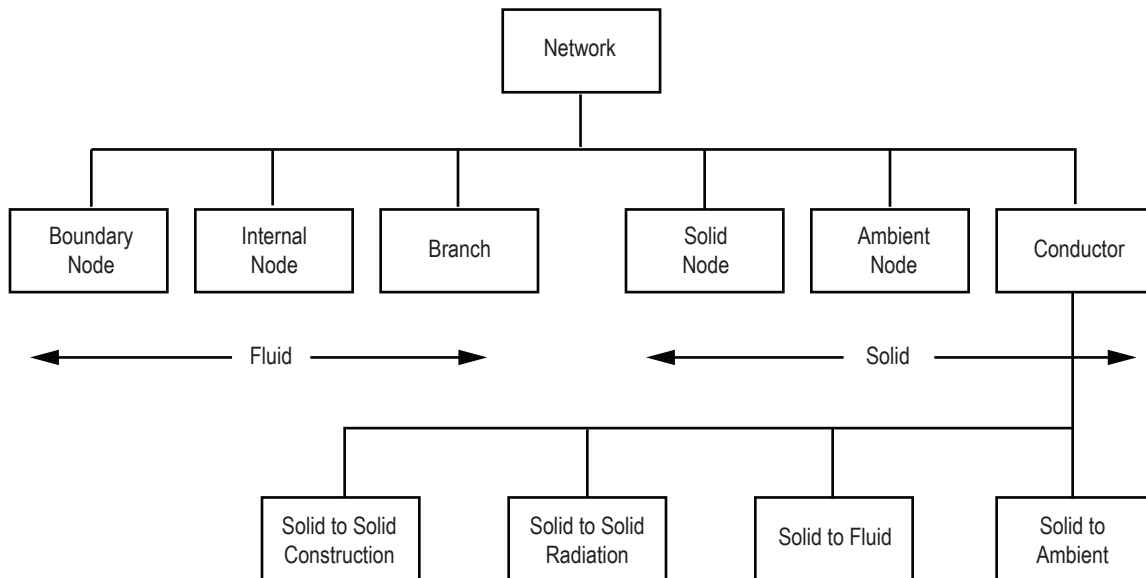
Figure 13. Network elements for conjugate heat transfer.

18
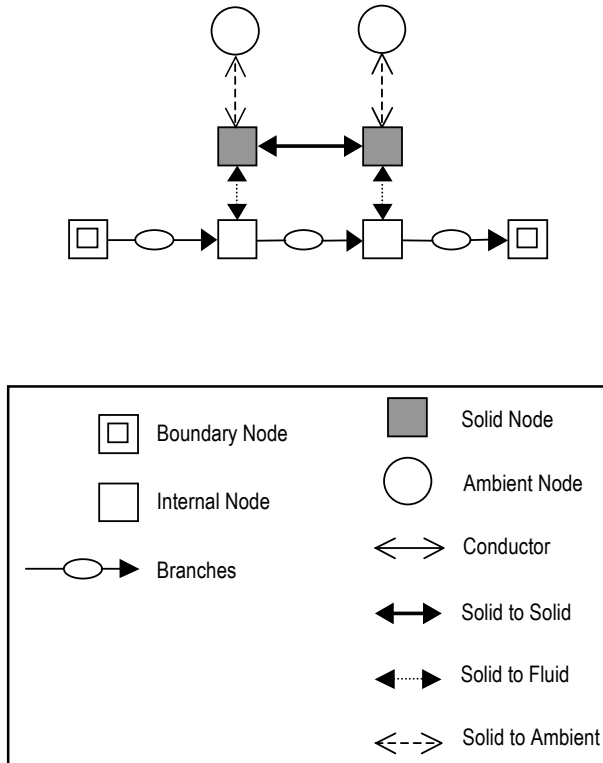
Figure 14. GFSSP network for conjugate heat transfer.

2.6 Solid Node Properties

The properties of a solid node are shown in figure 15. In addition to name, material, mass, and specific heat, there are six more relational properties that identify the number and names of solid to solid, solid to fluid, and solid to ambient conductors.
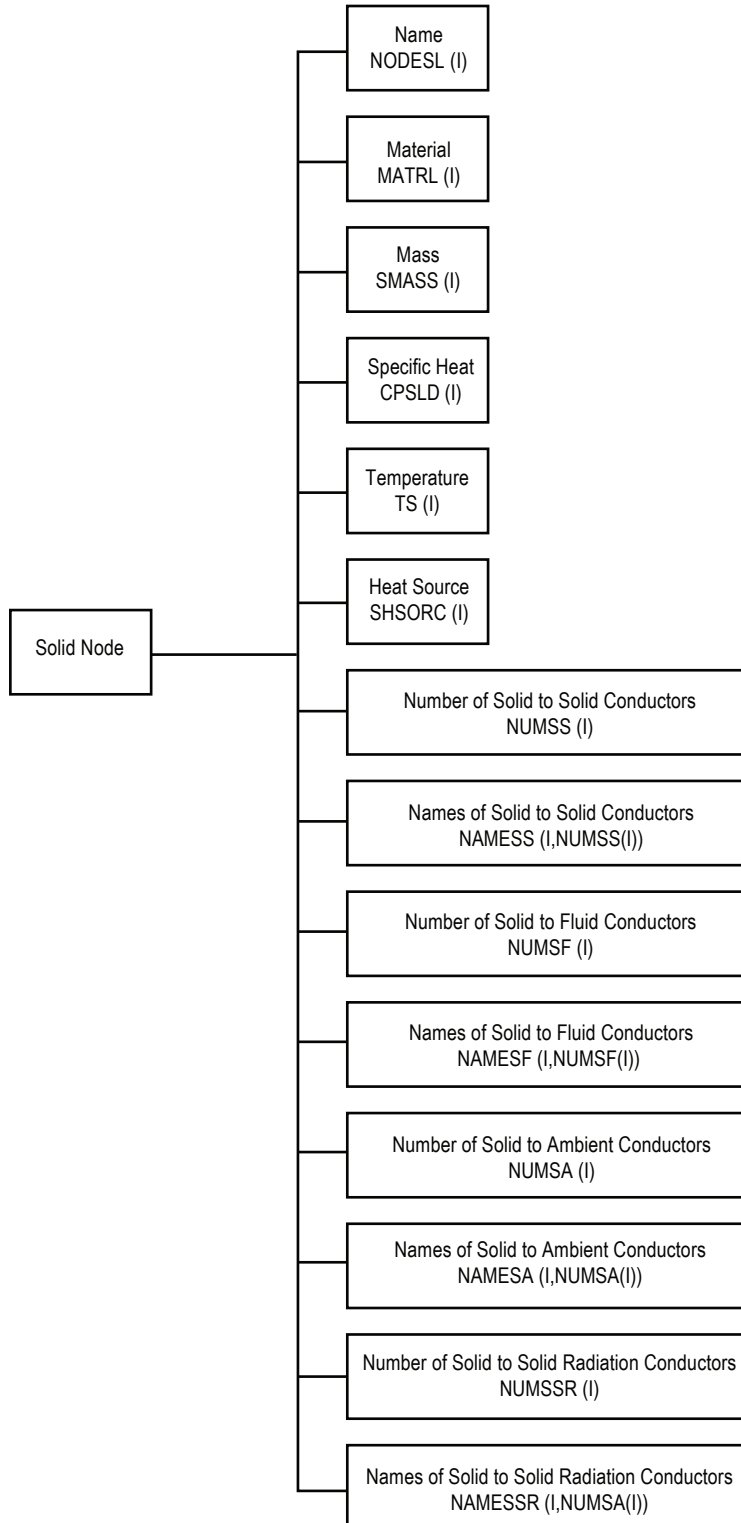
Figure 15. Properties of solid node.

2.7 Solid to Solid Conductor

The properties of a solid to solid conductor are shown in figure 16. The relational properties are names of connecting solid and fluid nodes. The geometric properties are area and distance between adjacent solid nodes. The thermophysical property includes conductivity and effective conductance.
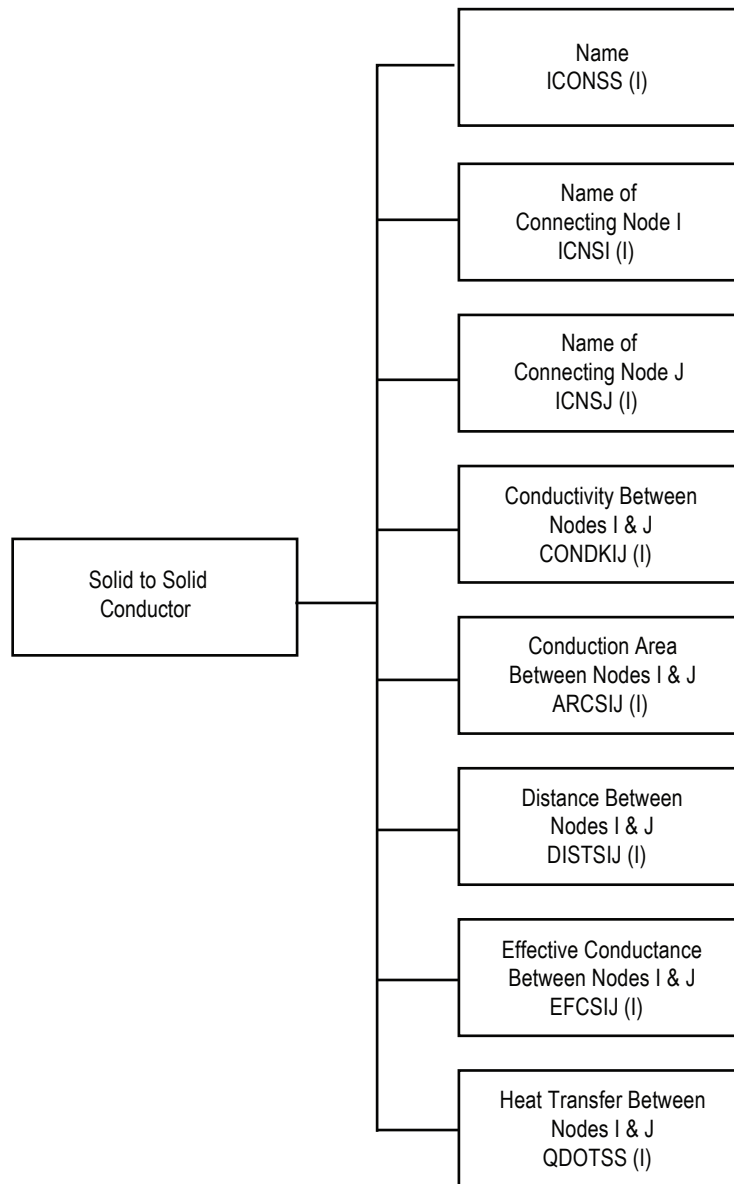
Figure 16. Properties of solid to solid conductor.

2.8 Solid to Fluid Conductor

The properties of solid to fluid conductors are shown in figure 17. The relational properties are names of connecting solid and fluid nodes. The geometric and thermofluid properties are heat transfer area, heat transfer coefficient, effective conductance, and emissivity of solid and fluid nodes.
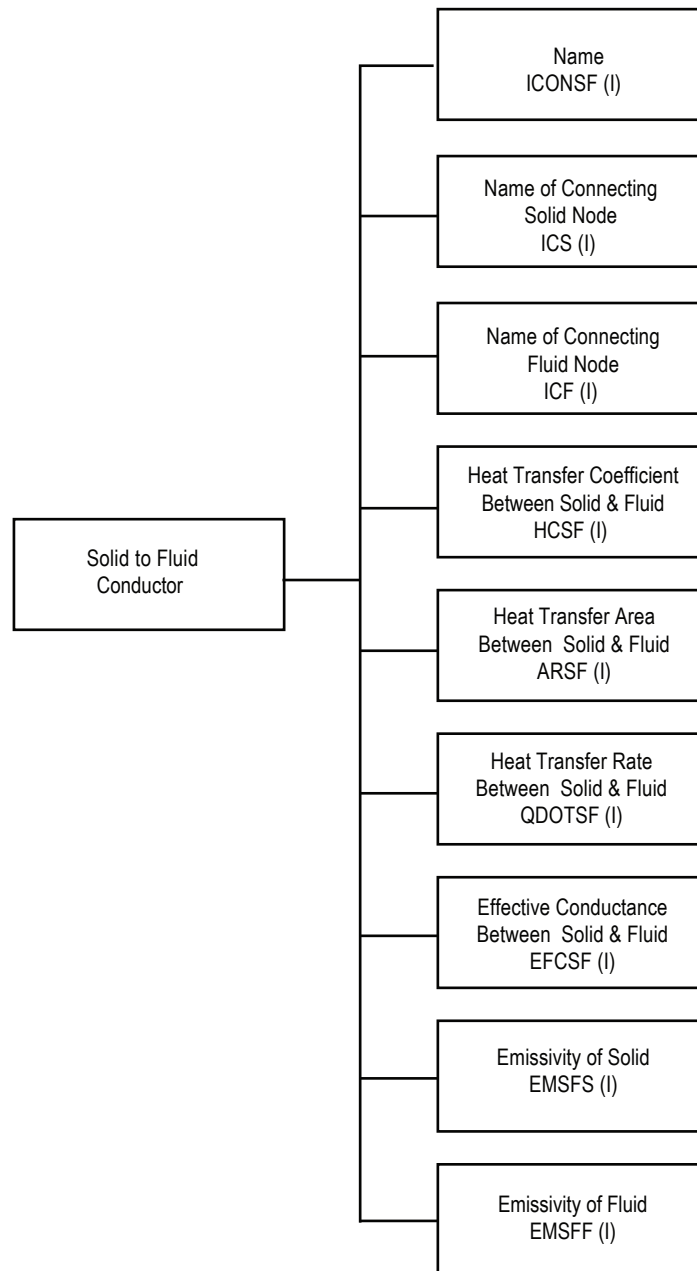
Figure 17. Properties of solid to fluid conductor.

2.9 Ambient Node Properties

Ambient node has only two properties: name and temperature (fig. 18).

```
              ┌──────────────┐
              │   Ambient    │
              │    Node      │
              └──────────────┘
                     │
          ┌──────────┴──────────┐
  ┌───────────────┐      ┌───────────────┐
  │     Name      │      │  Temperature  │
  │    NODEAM     │      │  RE TAMB (I)  │
  └───────────────┘      └───────────────┘
```
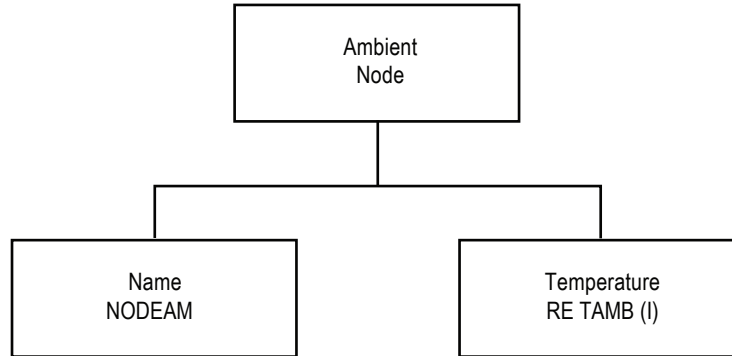
Figure 18. Properties of ambient node.

2.10 Solid to Ambient Conductor

The properties of a solid to ambient conductor are shown in figure 19. The relational properties of a solid to ambient conductor are names of connecting solid and ambient nodes. The geometric and thermofluid properties include heat transfer area, heat transfer coefficient, effective conductance, and emissivity of solid and ambient.
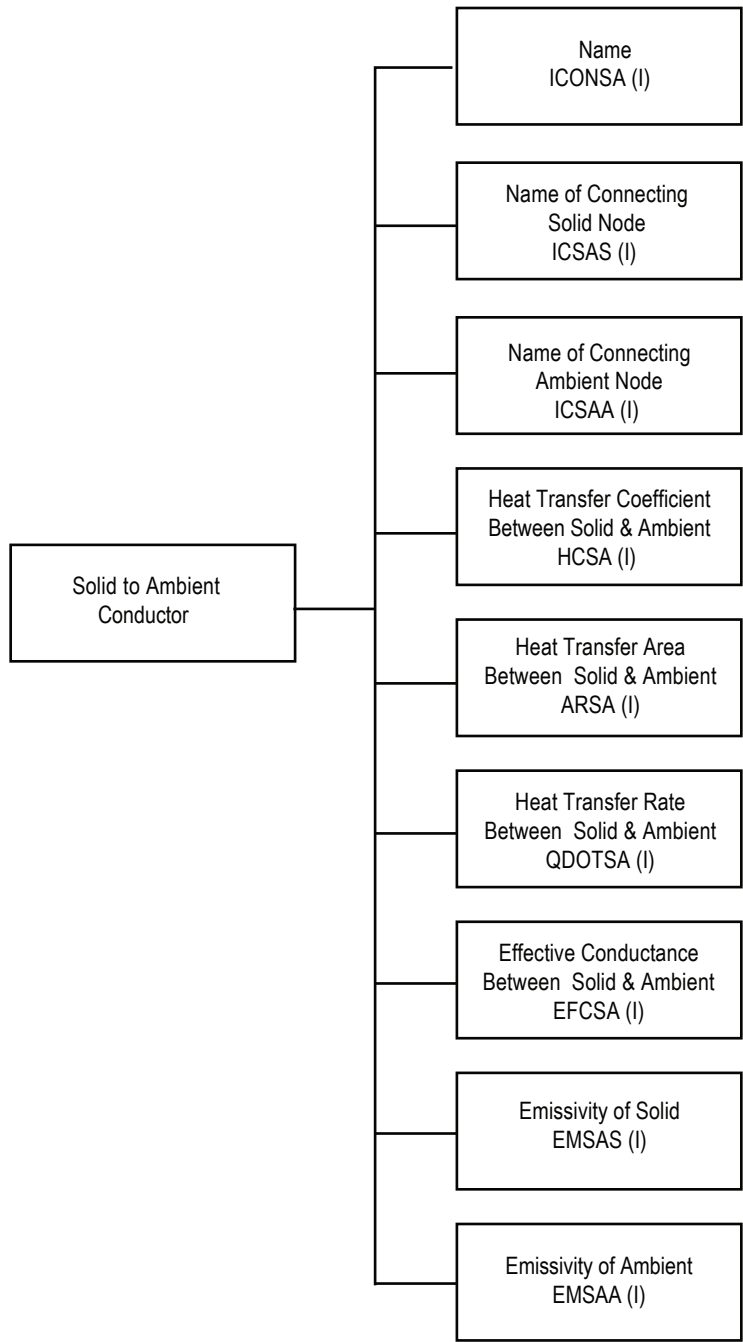
Figure 19. Properties of solid to ambient conductor.

3. MATHEMATICAL FORMULATION

GFSSP assumes a Newtonian, nonreacting and one-dimensional flow in the flow circuit. The flow can be steady or unsteady, laminar or turbulent, incompressible or compressible, with or without heat transfer, phase change, mixing, and rotation. The analysis of thermofluid dynamics in a complex network requires resolution of the system into fluid nodes and branches, and solid nodes and conductors. GFSSP calculates scalar properties such as pressure, temperature, and density at the nodes, and vector properties such as flow rates, heat fluxes at fluid branches, and conductors, respectively. Fluid nodes can be either internal nodes where properties are calculated or boundary nodes where properties are specified. Temperatures are calculated at the solid nodes and specified at the ambient nodes. This section describes all governing equations and solution procedure.

3.1 Governing Equations

Figure 20 displays a schematic showing adjacent nodes, their connecting branches, and the indexing system. In order to solve for the unknown variables, mass, energy, and fluid species, conservation equations are written for each internal node and flow rate equations are written for each branch.
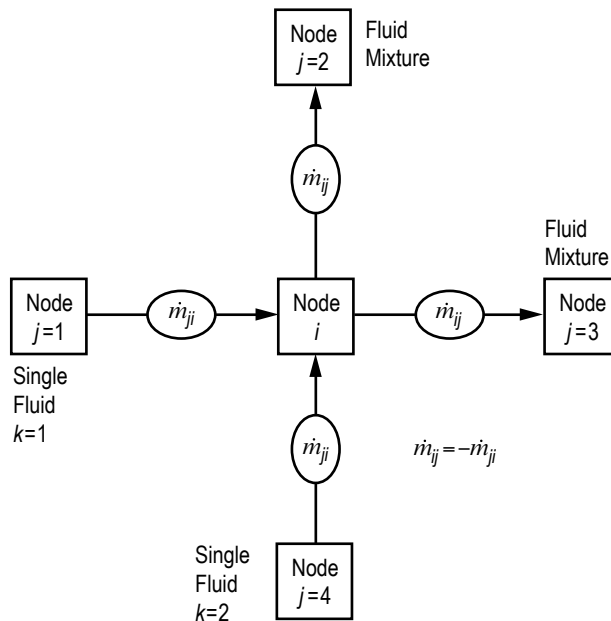
Figure 20. Schematic of GFSSP nodes, branches, and indexing practice.

3.1.1 Mass Conservation Equation

Following is the mass conservation equation:

$$\frac{m_{\tau+\Delta\tau} - m_\tau}{\Delta\tau} = -\sum_{j=1}^{j=n} \dot{m}_{ij} \ . \tag{1}$$

Equation (1) requires that for the unsteady formulation, the net mass flow from a given node must equate to the rate of change of mass in the control volume. In the steady state formulation, the left side of the equation is zero. This implies that the total mass flow rate into a node is equal to the total mass flow rate out of the node. Each term in equation (1) has the unit of lb/s.

3.1.2 Momentum Conservation Equation

The flow rate in a branch is calculated from the momentum conservation equation (eq. (2)), which represents the balance of fluid forces acting on a given branch. A typical branch configuration is shown in figure 21. Inertia, pressure, gravity, friction, and centrifugal forces are considered in the conservation equation. In addition to these five forces, a source term, S, has been provided in the equation to input pump characteristics or to input power to a pump in a given branch. If a pump is located in a given branch, all other forces except pressure are set to zero. The source term, S, is set to zero in all branches without a pump or other external momentum source:

$$\frac{(mu)_{\tau+\Delta\tau} - (mu)_\tau}{g_c \Delta\tau} + \text{MAX}\left|\dot{m}_{ij}, 0\right|\left(u_{ij} - u_u\right) - \text{MAX}\left|-\dot{m}_{ij}, 0\right|\left(u_{ij} - u_u\right)$$

-----Unsteady----- ------------------Longitudinal Inertia--------------------

$$+ \text{MAX}\left|\dot{m}_{\text{trans}}, 0\right|\left(u_{ij} - u_p\right) - \text{MAX}\left|-\dot{m}_{\text{trans}}, 0\right|\left(u_{ij} - u_p\right)$$

------------------------Transverse Inertia------------------------

$$= \left(p_i - p_j\right)A_{ij} + \frac{\rho g V \cos\theta}{g_c} - K_f \dot{m}_{ij}\left|\dot{m}_{ij}\right|A_{ij} + \frac{\rho K_{\text{rot}}^2 \omega^2 A}{g_c} + \mu \frac{u_p - u_{ij}}{g_c \delta_{ij,p}} A_s$$

--Pressure-- --Gravity-- ----Friction---- -Centrifugal- ----Shear----

$$- \rho A_{\text{norm}} u_{\text{norm}} u_{ij} / g_c + \left(\mu_d \frac{u_d - u_{ij}}{\delta_{ij,d}} - \mu_u \frac{u_{ij} - u_u}{\delta_{ij,u}}\right)\frac{A_{ij}}{g_c} + S \ .$$

-Moving Boundary- ----------Normal Stress--------- -Source-

$$\tag{2}$$
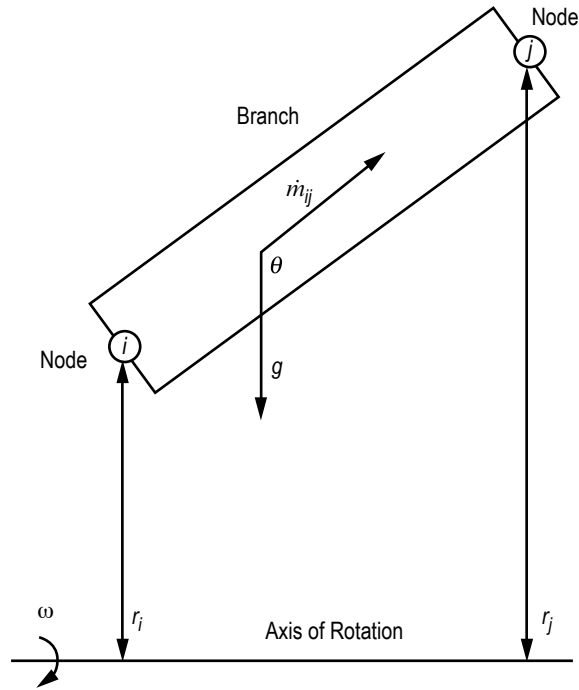
Figure 21. Schematic of a branch showing gravity and rotation.

The momentum equation consists of 11 terms. There will be no occasion when all 11 terms will be present in a control volume. Users have the ability to include or exclude all terms except the pressure term. For example, the friction and shear term will never be active at the same branch. The shear term will be activated for multidimensional flow modeling when the friction term must be set to zero. The pressure term will be active under all circumstances. The left-hand side of the momentum equation represents the inertia of the fluid. The surface and body forces applied in the control volume are assembled in the right-hand side of the equation. Each term of equation (2) has the unit of lb_f. Following are descriptions of the 11 terms:

(1) Unsteady—This term represents rate of change of momentum with time. For steady state flow, time step is set to an arbitrary large value and this term is reduced to zero.

(2) Longitudinal inertia—This term is important when there is a significant change in velocity in the longitudinal direction due to change in area and density. An upwind differencing scheme is used to compute the velocity differential. Flow in a converging-diverging nozzle is an example where this term must be active.

(3) Transverse inertia—This term is important for multidimensional flow. It accounts for any longitudinal momentum being transported by a transverse velocity component. Once again, an upwind differencing scheme is used to compute the velocity differential.

(4) Pressure—This term represents the pressure gradient in the branch. The pressures are located at the upstream and downstream face of a branch.

27

(5) Gravity—This term represents the effect of gravity. The gravity vector makes an angle (θ) with the assumed flow direction vector. At $\theta = 180°$, the fluid is flowing against gravity; at $\theta = 90°$, the fluid is flowing horizontally, and gravity has no effect on the flow.

(6) Friction—This term represents the frictional effect. Friction was modeled as a product of K_f and the square of the flow rate and area. K_f is a function of the fluid density in the branch and the nature of the flow passage being modeled by the branch. The calculation of K_f for different types of flow passages is described in section 3.1.7.

(7) Centrifugal—This term in the momentum equation represents the effect of the centrifugal force. This term will be present only when the branch is rotating as shown in figure 21. $K_{rotation}$ is the factor representing the fluid rotation. $K_{rotation}$ is unity when the fluid and the surrounding solid surface rotate with the same speed. This term also requires knowledge of the distances from the axis of rotation between the upstream and downstream faces of the branch.

(8) Shear—This term represents shear force exerted on the control volume by a neighboring branch. This term is active only for multidimensional flow. The friction term is deactivated when this term is present. This term requires knowledge of distances between branches to compute the shear stress.

(9) Moving boundary—This term represents force exerted on the control volume by a moving boundary. This term is not active for multidimensional calculations.

(10) Normal stress—This term represents normal viscous force. This term is important for highly viscous flows.

(11) Source—This term represents a generic source term. Any additional force acting on the control volume can be modeled through the source term. In a system level model, a pump can be modeled by this term. A detailed description of modeling a pump by this source term (S) appears in sections 3.1.7.14 and 3.1.7.15.

A simplified form of the momentum equation has also been provided to compute flow rate for compressible flow in an orifice. For the ratio of downstream to upstream pressure,

$$\frac{p_j}{p_i} < p_{cr} \ , \tag{3a}$$

where

$$p_{cr} = \left(\frac{2}{\gamma + 1} \right)^{\frac{\gamma}{\gamma - 1}} . \tag{3b}$$

The flow rate in a branch is calculated from

$$\dot{m}_{ij} = C_{Lij} A \sqrt{p_i \rho_i g_c \frac{2\gamma}{\gamma-1} (p_{cr})^{2/\gamma} \left[1 - (p_{cr})^{(\gamma-1)/\gamma} \right]} \ . \tag{3c}$$

If $\dfrac{p_j}{p_i} > p_{cr}$, the flow rate in a branch is calculated from:

$$\dot{m}_{ij} = C_{Lij} A \sqrt{p_i \rho_i g_c \frac{2\gamma}{\gamma-1} \left(\frac{p_j}{p_i} \right)^{2/\gamma} \left[1 - \left(\frac{p_j}{p_i} \right)^{(\gamma-1)/\gamma} \right]} \ . \tag{3d}$$

It may be noted that this is a special form of the momentum equation and no other terms of the momentum equation can be activated when the compressible orifice equation (eqs. (3c) or (3d)) is in use.

3.1.3 Energy Conservation Equations

GFSSP solves for the energy conservation equations for both fluid and solid at internal fluid nodes and solid nodes. The energy conservation equation for fluid is solved for all real fluids with or without heat transfer. For conjugate heat transfer, the energy conservation equation for solid nodes is solved in conjunction with the energy equation of fluid nodes. The heat transfer between solid and fluid nodes is calculated at the interface and used in both equations as source and sink terms.

3.1.3.1 Energy Conservation Equation of Single Fluid. The energy conservation equation for node i, shown in figure 20, can be expressed following the first or second law of thermodynamics. The first law formulation uses enthalpy as the dependent variable, while the second law formulation uses entropy. The energy conservation equation based on enthalpy is shown in equation (4a):

$$\frac{m \left(h - \dfrac{p}{\rho J} \right)_{\tau+\Delta\tau} - m \left(h - \dfrac{p}{\rho J} \right)_{\tau}}{\Delta\tau} = \sum_{j=1}^{j=n} \left\{ \mathrm{MAX} \left[-\dot{m}_{ij}, 0 \right] h_j - \mathrm{MAX} \left[\dot{m}_{ij}, 0 \right] h_i \right\}$$

$$+ \frac{\mathrm{MAX} \left[-\dot{m}_{ij}, 0 \right]}{\left| \dot{m}_{ij} \right|} \left[\left(p_i - p_j \right) + K_{ij} \dot{m}_{ij}^2 \right] \left(v_{ij} A \right) + Q_i \ . \tag{4a}$$

Equation (4a) shows that for transient flow, the rate of increase of internal energy in the control volume is equal to the rate of energy transport into the control volume minus the rate of energy transport from the control volume plus the rate of work done on the fluid by the pressure force plus the rate of work done on the fluid by the viscous force plus the rate of heat transfer into the control volume. The term $(p_i - p_j) v_{ij} A_{ij}$ represents work input to the fluid due to rotation or

29

having a pump in the upstream branch of node i. The term $K_{ij} \dot{m}_{ij}^2 v_{ij} A_{ij}$ represents viscous work in the upstream branch of the node i where v_{ij} and A_{ij} are velocity and area of the upstream branch.

For a steady state situation, the energy conservation equation, equation (4a), states that the net energy flow from a given node must equate to zero. In other words, the total energy leaving a node is equal to the total energy coming into the node from neighboring nodes and from any external heat sources (Q_i) coming into the node and work done on the fluid by pressure and viscous forces. The MAX operator used in equation (4a) is known as an upwind differencing scheme and has been extensively employed in the numerical solution of Navier-Stokes equations in convective heat transfer and fluid flow applications.[9] When the flow direction is not known beforehand, this operator allows the transport of energy only from its upstream neighbor. In other words, the upstream neighbor influences its downstream neighbor but not vice versa. The second term on the right-hand side represents the work done on the fluid by the pressure and viscous force. The difference between the steady and unsteady formulation lies in the left side of the equation. For a steady state situation, the left side of equation (4a) is zero, whereas in unsteady cases, the left-hand side of the equation must be evaluated.

The energy conservation equation based on entropy is shown in equation (4b):

$$\frac{(ms)_{\tau+\Delta\tau} - (ms)_{\tau}}{\Delta\tau} = \sum_{j=1}^{j=n} \left\{ \text{MAX}\left[-\dot{m}_{ij},0\right] s_j - \text{MAX}\left[\dot{m}_{ij},0\right] s_i \right\}$$

$$+ \sum_{j=1}^{j=n} \left\{ \frac{\text{MAX}\left[-\dot{m}_{ij},0\right]}{\left|\dot{m}_{ij}\right|} \right\} \dot{S}_{ij,gen} + \frac{Q_i}{T_i} \quad . \tag{4b}$$

The entropy generation rate due to fluid friction in a branch is expressed as

$$\dot{S}_{ij,gen} = \frac{\dot{m}_{ij} \Delta p_{ij,viscous}}{\rho_u T_u J} = \frac{K_f \left(\left|\dot{m}_{ij}\right|\right)^3}{\rho_u T_u J} \quad . \tag{4c}$$

Equation (4b) shows that for unsteady flow, the rate of increase of entropy in the control volume is equal to the rate of entropy transport into the control volume plus the rate of entropy generation in all upstream branches due to fluid friction plus the rate of entropy added to the control volume due to heat transfer. The first term on the right-hand side of the equation represents the convective transport of entropy from neighboring nodes. The second term represents the rate of entropy generation in branches connected to the ith node. The third term represents entropy change due to heat transfer. Each term in equation (4b) has the unit of Btu/R-s.

3.1.3.2 Energy Conservation Equation of Fluid Species. Energy conservation equations of fluid species are necessary for modeling fluid mixtures. GFSSP asumes a fluid mixture to be homogeneous, and therefore the mass and momentum equations are identical to those of a single fluid.

GFSSP has three options to model a mixture to calculate the temperature and thermophysical properties of the mixture. The first two options can be used for a mixture of gas and/or liquid as long as there is no change of phase in any mixture component. The third option handles a mixture of liquid and gas where the liquid or gas may go through a phase change. In all three options the conservation equations of fluid species are solved as described in section 3.1.4. However, the three options differ in the way energy equations are handled. In the first option (referred to as the temperature option), the energy equation is solved in terms of temperature as described in section 3.1.6. In the second option (referred to as the enthalpy-1 option), a mixture enthalpy was calculated for the energy conservation equation from enthalpies of fluid species, and temperature was calculated by an iterative method from a mixture enthalpy equation. In the third option (referred to as the enthalpy-2 option), separate energy equations for each species are solved and the temperature of the mixture is calculated by averaging the thermal mass (product of mass and specific heat) of all components. In this section, the energy conservation equations for both enthalpy options are described:

(1) Enthalpy-1 option—The enthalpy at node i in figure 20 is calculated from the following equation which is derived from the energy conservation equation expressed in terms of individual species concentration and enthalpy:

$$h_{i,\tau+\Delta\tau} = \frac{\displaystyle\sum_{j=1}^{j=n}\sum_{k=1}^{k=nf} x_{j,k}h_{j,k}\mathrm{MAX}\left[-\dot{m}_{ij},0\right]+\dfrac{\left(mh_i\right)_\tau}{\Delta\tau}+\dot{Q}_i}{\displaystyle\sum_{j=1}^{j=n}\sum_{k=1}^{k=nf} x_{j,k}\mathrm{MAX}\left[\dot{m}_{ij},0\right]+\dfrac{m_{\tau+\Delta\tau}}{\Delta\tau}} \,.$$

(5)

The method of calculating temperature in the enthalpy-1 option is described in section 3.1.6.

(2) Enthalpy-2 option—In this option, a separate energy equation for individual species is solved. The energy equation for individual species (k) can be expressed as:

$$\frac{\left(m_i h_{ik} - \dfrac{p}{\rho_k J}\right)_{\tau+\Delta\tau} - \left(m_i h_{ik} - \dfrac{p}{\rho_k J}\right)_\tau}{\Delta\tau}$$

Transient term

$$= \sum_{j=1}^{j=n}\left\{\mathrm{MAX}\left[-\dot{m}_{ij},0\right]h_{ji} - \mathrm{MAX}\left[\dot{m}_{ij},0\right]h_{ik}\right\}+\dot{Q}_{ik} \,.$$

Advection term   + Source term   (6)

The external heat source is expressed as:

$$\dot{Q}_{ik} = \bar{c}_{ik}\dot{Q}_i \,,$$

where

$\dot{Q}_i$ = external heat source, i.e., heat from solid node, etc.

$\overline{c}_{i_{k'}}$ = molar concentration of kth species in the ith node.

It may be noted that work input and viscous work were neglected in the species energy equation. The method of calculating temperature and mixture properties is also described in section 3.1.6.

3.1.3.3 Energy Conservation Equation of Solid. Typically, a solid node can be connected with other solid nodes, fluid nodes, and ambient nodes. Figure 22 shows a typical arrangement where a solid node is connected with other solid nodes, fluid nodes, and ambient nodes. The energy conservation equation for solid node i can be expressed as:

$$\frac{\partial}{\partial \tau}\left(mC_pT_s^i\right) = \sum_{j_s=1}^{n_{ss}} \dot{q}_{ss} + \sum_{j_f=1}^{n_{sf}} \dot{q}_{sf} + \sum_{j_a=1}^{n_{sa}} \dot{q}_{sa} + \dot{S}_i \; . \tag{7}$$
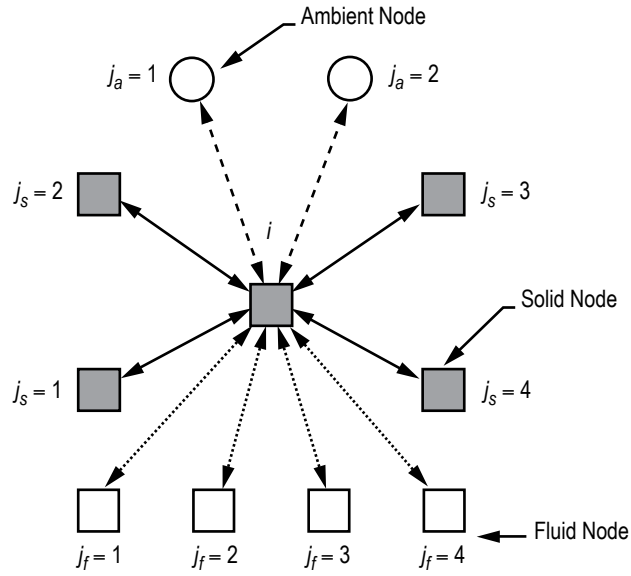
Figure 22. Schematic showing the connection of a solid node with neighboring solid, fluid, and ambient nodes.

The left-hand side of the equation represents rate of change of temperature of the solid node, i. The right-hand side of the equation represents the heat transfer from the neighboring node and heat source or sink. The heat transfer from neighboring solid, fluid, and ambient nodes, respectively, can be expressed as

$$\dot{q}_{ss} = k_{ij_s} A_{ij_s} / \delta_{ij_s} \left(T_s^{j_s} - T_s^i\right) , \tag{7a}$$

$$\dot{q}_{sf} = h_{ij_f} A_{ij_f} \left(T_f^{j_f} - T_s^i \right) , \tag{7b}$$

and

$$\dot{q}_{sa} = h_{ij_a} A_{ij_a} \left(T_a^{j_a} - T_s^i \right) . \tag{7c}$$

The heat transfer rate can be expressed as a product of conductance and temperature differential. The conductance for equations (7a)–(7c) are as follows:

$$C_{ij_s} = \frac{k_{ij_s} A_{ij_s}}{\delta_{ij_s}}, \quad C_{ij_f} = h_{ij_f} A_{ij_f}, \text{ and } C_{ij_f} = h_{ij_a} A_{ij_a} , \tag{7d}$$

respectively, where effective heat transfer coefficients for solid to fluid and solid to ambient nodes are expressed as the sum of the convection and radiation:

$$h_{ij_f} = h_{c,ij_f} + h_{r,ij_f}$$

$$h_{ij_a} = h_{c,ij_a} + h_{r,ij_a} \tag{7e}$$

and

$$h_{r,ij_f} = \frac{\sigma \left[\left(T_f^{j_f} \right)^2 + \left(T_s^i \right)^2 \right] \left[T_f^{j_f} + T_s^i \right]}{1 / \varepsilon_{ij,f} + 1 / \varepsilon_{ij,s} - 1}$$

$$h_{r,ij_a} = \frac{\sigma \left[\left(T_a^{j_a} \right)^2 + \left(T_s^i \right)^2 \right] \left[T_a^{j_a} + T_s^i \right]}{1 / \varepsilon_{ij,a} + 1 / \varepsilon_{ij,s} - 1} . \tag{7f}$$

GFSSP provides users with four different options for specifying heat transfer coefficient:

(1) User can provide a constant heat transfer coefficient.
(2) User can specify the Dittus-Boelter equation[14] for single-phase flow where the Nusselt number is expressed as:

$$\frac{h_c D}{k_f} = 0.023 (\text{Re})^{0.8} (\text{Pr})^{0.33} , \tag{7g}$$

where $\text{Re} = \rho u D / \mu_f$ and $\text{Pr} = C_p \mu_f / k_f$.

(3) User can specify modified Miropolskii's correlation[15] for two-phase flow:

$$\mathrm{Nu} = 0.023 \left(\mathrm{Re}_{mix} \right)^{0.8} \left(\mathrm{Pr}_v \right)^{0.4} \left(Y \right)$$

$$\mathrm{Re}_{mix} = \left(\frac{\rho u D}{\mu_v} \right) \left[x + \left(\frac{\rho_v}{\rho_l} \right) (1-x) \right]$$

$$\mathrm{Pr}_v = \left(\frac{C_p \mu_v}{k_v} \right)$$

$$Y = 1 - 0.1 \left(\frac{\rho_l}{\rho_v} - 1 \right)^{0.4} (1-x)^{0.4} \; . \tag{7h}$$

(4) User can provide a new correlation in the User Subroutine to be described in section 4. Equation (7) can be rearranged to determine T_s^i:

$$T_s^i = \frac{\displaystyle\sum_{j_s=1}^{n_{ss}} C_{ij_s} T_s^{j_s} + \sum_{j_f=1}^{n_{sf}} C_{ij_f} T_f^{j_f} + \sum_{j_a=1}^{n_{sa}} C_{ij_a} T_a^{j_a} + \frac{\left(m C_p \right)_m}{\Delta \tau} T_{s,m}^i + \dot{S}}{\dfrac{m C_p}{\Delta \tau} + \displaystyle\sum_{j_s=1}^{n_{ss}} C_{ij_s} + \sum_{j_f=1}^{n_{sf}} C_{ij_f} + \sum_{j_a=1}^{n_{sa}} C_{ij_a}} \; . \tag{8}$$

3.1.4 Fluid Species Conservation Equation

For a fluid mixture, density is a function of mass fraction of fluid species. In order to calculate the density of the mixture, the concentration of the individual fluid species within the branch must be determined. The concentration for the kth species can be written as

$$\frac{\left(m_i c_{i,k} \right)_{\tau + \Delta \tau} - \left(m_i c_{i,k} \right)_\tau}{\Delta \tau} = \sum_{j=1}^{j=n} \left\{ \mathrm{MAX} \left[-\dot{m}_{ij}, 0 \right] c_{j,k} - \mathrm{MAX} \left[\dot{m}_{ij}, 0 \right] c_{i,k} \right\} + \dot{S}_{i,k} \; . \tag{9}$$

For a transient flow, equation (7) states that the rate of increase of the concentration of the kth species in the control volume equals the rate of transport of the kth species into the control volume minus the rate of transport of the kth species out of the control volume plus the generation rate of the kth species in the control volume.

Like equation (4), for steady state conditions, equation (7) requires that the net mass flow of the kth species from a given node must equate to zero. In other words, the total mass flow rate of the given species into a node is equal to the total mass flow rate of the same species out of that node. For steady state, the left side of equation (7) is zero. For the unsteady formulation, the resident mass in the control volume is changing and therefore the left side must be computed. Each term in equation (5) has the unit of lb/s.

3.1.5 Thermodynamic and Thermophysical Properties

The momentum conservation equation, equation (2), requires knowledge of the density and the viscosity of the fluid within the branch. These properties are functions of the temperatures, pressures, and concentrations of fluid species for a mixture. Three thermodynamic property routines have been integrated into the program to provide the required fluid property data. GASP[10] provides the thermodynamic and transport properties for 10 fluids. These fluids include hydrogen, oxygen, helium, nitrogen, methane, carbon dioxide, carbon monoxide, argon, neon, and fluorine. WASP[11] provides the thermodynamic and transport properties for water and steam. For RP-1 fuel, a lookup table of properties has been generated by a modified version of GASP. An interpolation routine has been developed to extract the required properties from the tabulated data. GASPAK[12] provides thermodynamic properties for helium, methane, neon, nitrogen, carbon monoxide, oxygen, argon, carbon dioxide, hydrogen, parahydrogen, water, RP-1, isobutane, butane, deuterium, ethane, ethylene, hydrogen sulfide, krypton, propane, xenon, R-11, R-12, R-22, R-32, R-123, R-124, R-125, R-134A, R-152A, nitrogen trifluoride, ammonia, hydrogen peroxide, and air.

3.1.5.1 Equation of State for a Real Fluid. Transient flow calculations require the knowledge of resident mass in a control volume. The resident mass is calculated from the equation of state for real fluid that can be expressed as

$$m = \frac{pV}{zRT} \ . \tag{10}$$

It may be noted that equation (10) is valid for liquid, gas, and gas-liquid mixture. For an ideal gas compressibility factor, z is unity. The compressibility factor for real gas is computed from the equation of state of real fluids using the above-mentioned thermodynamic property programs. For a two-phase mixture, z is computed from the following relation:

$$z = \frac{p}{\rho_{mix}RT} \ , \tag{10a}$$

where

$$\rho_{mix} = \frac{\rho_l \rho_g}{x\rho_l - (1-x)\rho_g} \tag{10b}$$

and

$$x = \frac{s - s_l}{s_g - s_l} \ . \tag{10c}$$

3.1.6 Mixture Property Calculations

This section describes the procedures developed for GFSSP to estimate the density and temperature of mixtures of real fluids for all three mixture options. We assume that n fluids are mixing in the ith node. At node i, pressure (p) and molar concentrations (x_k) are known. The problem is to calculate the density (ρ), temperature (T), specific heat (C_p), specific heat ratio (γ), and viscosity (μ) of the mixture at the ith node.

Density by Amagat's Model. GFSSP's default mixture model uses Amagat's Law of Partial Volumes. It is suitable for liquids and ideal gas mixtures. The density of the mixture in the node (ρ_{mix}) is a function of the densities of the individual components evaluated at the temperature and total pressure of the node:

$$\frac{1}{\rho_{mix}} = \sum_{k=1}^{k=n} \frac{x_k}{\rho_k} \ . \tag{11}$$

Density by Dalton's Model. GFSSP offers the option of evaluating mixture densities using Dalton's Law of Partial Pressures for gas mixtures. This option would be used when at least one species of the gas mixture is at or below its saturation temperature and would be a liquid at the full pressure of the node, but a gas at the partial pressure of the species. The density of the mixture in the node is a function of the densities of the individual components evaluated at the temperature of the node and the partial pressure of the species:

$$\rho_{mix} = \sum_{k=1}^{k=n} \rho_k \ . \tag{12}$$

Other Properties. The compressibility factor of the mixture (Z_{mix}) is calculated from the equation of state:

$$Z_{mix} = \frac{P}{\rho_{mix} R_{mix} T} \ . \tag{13}$$

The gas constant of the mixture (R_{mix}) is the universal gas constant divided by the molecular weight of the mixture:

$$R_{mix} = \frac{R_{univ}}{\sum\limits_{k=1}^{k=n} \bar{x}_k M_k} \ . \tag{14}$$

The viscosity, specific heat, and specific heat ratio of the mixture are calculated as the molar averages of the component properties as shown in equations (15)–(17).

$$\mu_{mix} = \sum_{k=1}^{k=m} \bar{x}_k \mu_k \ , \tag{15}$$

$$C_{p,mix} = \frac{\sum\limits_{k=1}^{k=n} C_{p,k}\,\bar{x}_k\,M_k}{\sum\limits_{k=1}^{k=n} \bar{x}_k\,M_k} \quad , \tag{16}$$

and

$$\gamma_{mix} = \sum_{k=1}^{k=n} \bar{x}_k \gamma_k \quad . \tag{17}$$

3.1.6.1 Mixture Temperature Option. The default method for calculating the temperature of a mixture is the Temperature Option. The unsteady formulation of the energy equation (eq. (4)) is rewritten, where the enthalpy is replaced by the product of the specific heat and temperature:

$$\left(T_i\right)_{\tau+\Delta\tau} = \frac{\sum\limits_{j=1}^{j=n}\sum\limits_{k=1}^{k=n_f} C_{p,k,j}\,x_{k,j}\,T_j\,\mathrm{MAX}\!\left[-\dot{m}_{ij},0\right] + \left(C_{V,i}\,m_i T_i\right)_{\tau}/\Delta\tau + Q_i}{\sum\limits_{j=1}^{j=n}\sum\limits_{k=1}^{k=n_f} C_{p,k,j}\,x_{k,j}\,\mathrm{MAX}\!\left[\dot{m}_{ij},0\right] + \left(C_{V,i}\,m\right)_{\tau+\Delta\tau}/\Delta\tau} \quad , \tag{18}$$

where $C_{p,k}$ is the molar specific heat and x_k is the mole-fraction of the kth species. It is stressed that this formulation does not handle phase-change and assumes that the specific heat is approximately constant over the temperature range of interest. It may also be noted that that work input and viscous work were neglected in this formulation of the energy equations.

3.1.6.2 Enthalpy-1 Option. The temperature is calculated by an iterative method where temperature is calculated from the following equation:

$$\sum_{k=1}^{k=n_f} x_{i,k} h_{i,k}\left(p_i, T_i\right) - h_i = 0 \quad . \tag{19}$$

It may be noted that equation (19) assumes Amagat's model. For Dalton's model equation (19) can be rewritten as:

$$\sum_{k=1}^{k=n_f} x_{i,k} h_{i,k}\left(p_{i,k}, T_i\right) - h_i = 0 \quad . \tag{19a}$$

Note that partial pressure has been used in equation (19a).

Once the temperature is calculated, the properties of individual species will be calculated from pressure and enthalpy of the individual species. For a gaseous mixture, the mixture properties will then be calculated by taking molar average of species properties as shown in equations (15)–(17).

3.1.6.3 Enthalpy-2 Option. Temperature and other properties of individual species are calculated from node pressure and the enthalpy of the species:

$$
\begin{aligned}
T_{ik} &= f\left(p_i, h_{ik}\right) \\
\rho_{ik} &= f\left(p_i, h_{ik}\right) \\
\mu_{ik} &= f\left(p_i, h_{ik}\right) \\
K_{ik} &= f\left(p_i, h_{ik}\right) \\
C_{p_{ik}} &= f\left(p_i, h_{ik}\right) .
\end{aligned}
\tag{20}
$$

The nodal properties are calculated by averaging the properties of species as shown in equations (15)–(17).

The temperature of the node is calculated from the following relation:

$$
T_i = \frac{\displaystyle\sum_{k=1}^{n_f} \bar{c}_{ik} \, C_{p_{ik}} \, T_{ik}}{C_{p_i}} .
\tag{21}
$$

3.1.7 Friction Calculations

It was mentioned earlier in this TM that the friction term in the momentum equation is expressed as a product of K_f, the square of the flow rate and the flow area. Empirical information is necessary to estimate K_f. Several options for flow passage resistance are listed in table 5. In this subsection, the expression of K_f for all resistance options is described.

3.1.7.1 Pipe Flow (Branch Option 1). Figure 23 shows the pipe resistance option parameters that are required by GFSSP. This option considers that the branch is a pipe with length (L), diameter (D), and surface roughness (ε). For this option, K_f can be expressed as:

$$
K_f = \frac{8 \, fL}{\rho_u \, \pi^2 D^5 g_c} ,
\tag{22}
$$

where ρ_u is the density of the fluid at the upstream node of a given branch. The derivation of K_f for pipe flow is covered in appendix A.

Detail A   Detail A

D = Pipe Diameter
L = Pipe Length
ε = Absolute Roughness
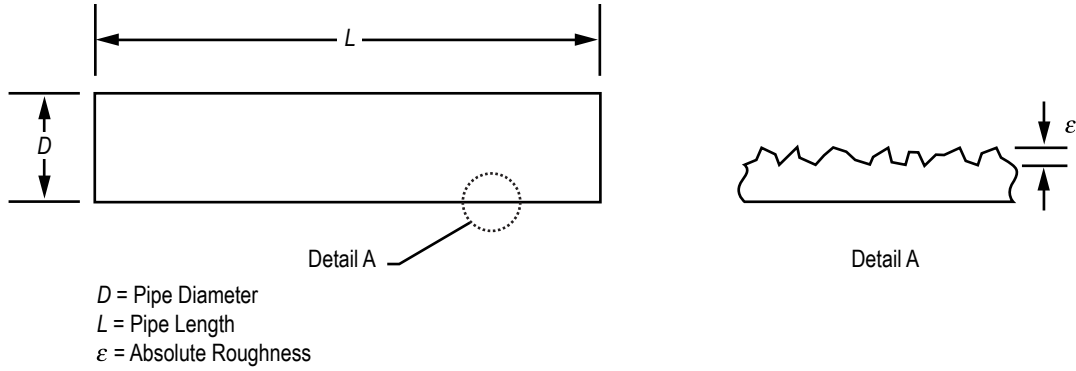
Figure 23. Pipe resistance option parameters.

The Darcy friction factor (f) is determined from the Colebrook equation,[16] which is expressed as:

$$\frac{1}{\sqrt{f}} = -2\log\left[\frac{\varepsilon}{3.7D} + \frac{2.51}{\mathrm{Re}\sqrt{f}}\right] \ ,$$ (23)

where ε/D and Re are the surface roughness factor and Reynolds number, respectively. It may be mentioned that all pipe flow options assume fully developed flow.

3.1.7.2 Flow Through a Restriction (Branch Option 2). This option regards the branch as a flow restriction with a given flow coefficient (C_L), and area (A). For this option, K_f can be expressed as:

$$K_f = \frac{1}{2g_c \rho_u C_L^2 A^2} \ .$$ (24)

In classical fluid mechanics, head loss is expressed in terms of a nondimensional 'K factor':

$$\Delta h = K\frac{u^2}{2g} \ .$$ (25)

K and C_L are related as:

$$C_L = \frac{1}{\sqrt{K}} \ .$$ (26)

3.1.7.3 Noncircular Duct (Branch Option 3). This option considers a duct with a non-circular cross section. Four different types of cross sections can be modeled as shown in figure 24.
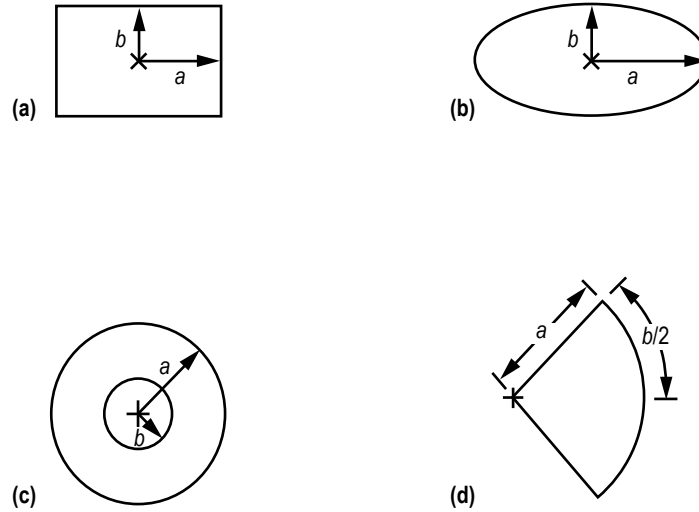
Figure 24. Noncircular duct cross section: (a) Rectangle, (b) ellipse, (c) concentrate annulus, and (d) circular sector.

White describes a procedure to estimate the friction factor in a noncircular duct.[17] This procedure consists of the following steps:

(1) Estimate the hydraulic diameter of the cross section: $D_h = (4)(\text{area})/\text{perimeter}$.

(2) Estimate the Poiseuille number (Po) for a particular cross section. The Poiseuille number can be expressed as a polynomial function of aspect ratio as shown in equation (27). Table 6 provides the coefficients for different geometries:

$$\text{Po} = A_0 + A_1\left(\frac{b}{a}\right) + A_2\left(\frac{b}{a}\right)^2 + A_3\left(\frac{b}{a}\right)^3 + A_4\left(\frac{b}{a}\right)^4 . \tag{27}$$

Table 6. Poiseuille number coefficients for noncircular duct cross sections.

Coefficients	Rectangle	Ellipse	Concentric* Cylinder	Circular Section
A_0	23.9201	19.7669	22.0513	11.9852
A_1	−29.436	−4.53458	6.44473	3.01553
A_2	30.3872	−11.5239	−7.35451	−1.09712
A_3	−10.7128	22.3709	2.78999	–
A_4	–	−10.0874	–	–

* For $b/a < 0.2508$ $P_0 = A_0\left(\frac{b}{a}\right)^{A_1}$, where, $A_0 = 24.8272$, $A_1 = 0.0479888$.

(3) Calculate the friction factor for a noncircular pipe:

• Laminar flow (Re<2,300)

$$f = \frac{4\,\text{Po}}{\text{Re}} \; . \tag{28}$$

• Turbulent flow

— Compute the effective diameter:

$$D_{eff} = \frac{16 D_h}{\text{Po}} \; . \tag{29}$$

— Compute the effective Reynolds number:

$$\text{Re}_{eff} = \frac{\dot{m}}{\mu} \frac{D_{eff}}{A} \; . \tag{30}$$

— Compute the friction factor using the Colebrook equation (eq. (23)).

(4) Compute K_f from the following expression:

$$K_f = \frac{8\,fL}{\rho_u \pi^2 D_h^5 g_c} \; . \tag{31}$$

3.1.7.4 Pipe With Entrance and Exit Loss (Branch Option 4). Figure 25 shows the pipe with entrance and/or exit loss resistance option parameters that are required by GFSSP. This option is an extension of option 1. In addition to the frictional loss in the pipe, entrance and exit losses are also calculated. For this option, K_f can be expressed as:

$$K_f = \frac{8 K_i}{\rho_u \pi^2 D^4 g_c} + \frac{8\,fL}{\rho_u \pi^2 D^5 g_c} + \frac{8 K_e}{\rho_u \pi^2 D^4 g_c} \; , \tag{32}$$

where K_i and K_e are the entrance and exit loss coefficients, respectively.

D = Pipe Diameter K_i = Entrance Loss Coefficient
L = Pipe Length K_e = Exit Loss Coefficient
ε = Absolute Roughness
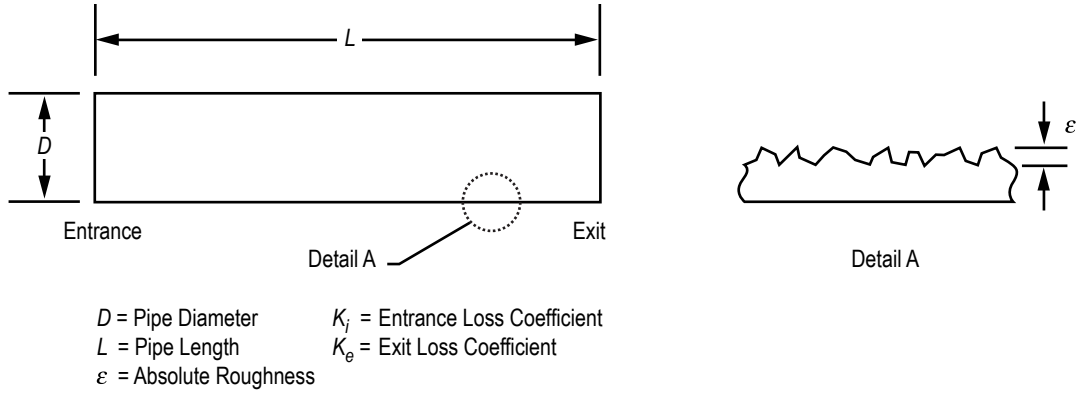
Figure 25. Pipe with entrance and/or exit loss resistance option parameters.

3.1.7.5 Thin, Sharp Orifice (Branch Option 5). Figure 26 shows the thin, sharp orifice resistance option parameters that are required by GFSSP. This option considers the branch as a thin, sharp orifice with a pipe diameter of D_1 and an orifice diameter of D_2. For this option, K_f can be expressed as:[18]

$$K_f = \frac{K_1}{2g_c \rho_u A^2} \; , \tag{33}$$

where $A = \dfrac{\pi D_1^2}{4}$.

For upstream Re ≤ 2,500,

$$K_1 = \left[2.72 + \left(\frac{D_2}{D_1} \right)^2 \left(\frac{120}{\text{Re}} - 1 \right) \right] \left[1 - \left(\frac{D_2}{D_1} \right)^2 \right] \left[\left(\frac{D_1}{D_2} \right)^4 - 1 \right] . \tag{34}$$

For upstream Re > 2,500,

$$K_1 = \left[2.72 - \left(\frac{D_2}{D_1} \right)^2 \left(\frac{4,000}{\text{Re}} \right) \right] \left[1 - \left(\frac{D_2}{D_1} \right)^2 \right] \left[\left(\frac{D_1}{D_2} \right)^4 - 1 \right] . \tag{35}$$

This option is recommended for subsonic and incompressible flow.
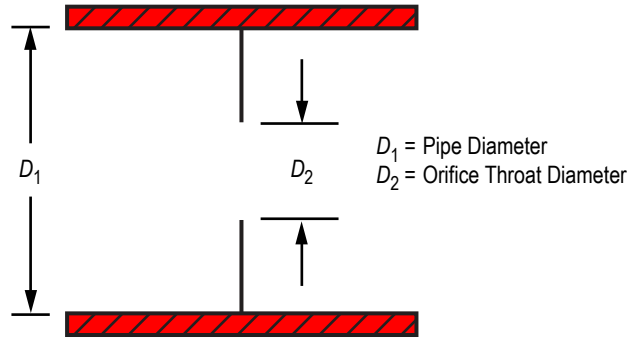
Figure 26. Thin sharp orifice resistance option parameters.

3.1.7.6 Thick Orifice (Branch Option 6). Figure 27 shows the thick orifice resistance option parameters that are required by GFSSP. This option models the branch as a thick orifice with a pipe diameter of D_1, an orifice diameter of D_2, and orifice length of L_{or}. This option should be used if $L_{or}/D_2 \leq 5$. If $L_{or}/D_2 > 5$, the user should use a square expansion, option 8, or a square reduction, option 7. For option 6, K_f can be expressed as in equation (33). However, the K_1 in equation (33) is calculated[18] from the following expressions:

For upstream Re ≤ 2,500,

$$K_1 = \left[2.72 + \left(\frac{D_2}{D_1} \right)^2 \left(\frac{120}{\text{Re}} - 1 \right) \right]\left[1 - \left(\frac{D_2}{D_1} \right)^2 \right]\left[\left(\frac{D_1}{D_2} \right)^4 - 1 \right]\left[0.584 + \frac{0.0936}{\left(L_{or}/D_2 \right)^{1.5} + 0.225} \right]. \quad (36)$$

For upstream Re > 2,500:

$$K_1 = \left[2.72 + \left(\frac{D_2}{D_1} \right)^2 \left(\frac{4,000}{\text{Re}} \right) \right]\left[1 - \left(\frac{D_2}{D_1} \right)^2 \right]\left[\left(\frac{D_1}{D_2} \right)^4 - 1 \right]\left[0.584 + \frac{0.0936}{\left(L_{or}/D_2 \right)^{1.5} + 0.225} \right]. \quad (37)$$

This option is recommended for subsonic and incompressible flow.
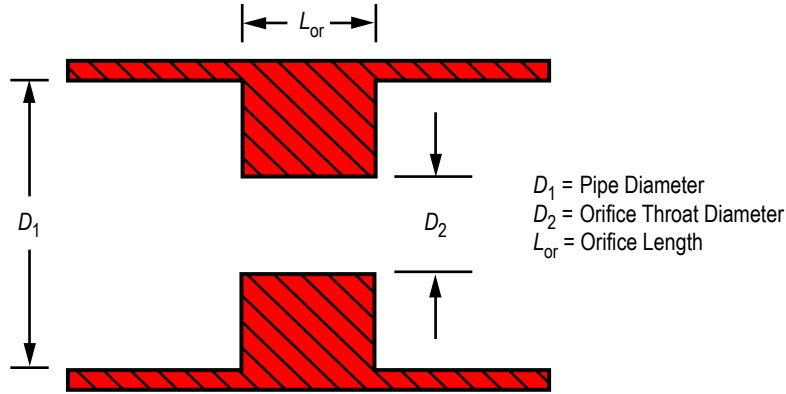
Figure 27. Thick orifice resistance option parameters.

3.1.7.7 Square Reduction (Branch Option 7). Figure 28 shows the square reduction resistance option parameters that are required by GFSSP. This option considers the branch as a square reduction. The diameters of the upstream and downstream pipes are D_1 and D_2, respectively. For this option, K_f can be expressed as in equation (33). However, the K_1 in equation (33) is calculated from the following expressions:[18]

For upstream Re ≤ 2,500,

$$K_1 = \left[1.2 + \frac{160}{\text{Re}}\right]\left[\left(\frac{D_1}{D_2}\right)^4 - 1\right].$$

(38)

For upstream Re > 2,500,

$$K_1 = \left[0.6 + 0.48 f\right]\left(\frac{D_1}{D_2}\right)^2 \left[\left(\frac{D_1}{D_2}\right)^2 - 1\right]^2.$$

(39)

The Reynolds number and friction factor that are utilized within these expressions are based on the upstream conditions. The user must specify the correct flow direction through this branch. If the model determines that the flow direction is in the reverse direction, the user will have to replace the reduction with an expansion and rerun the model.
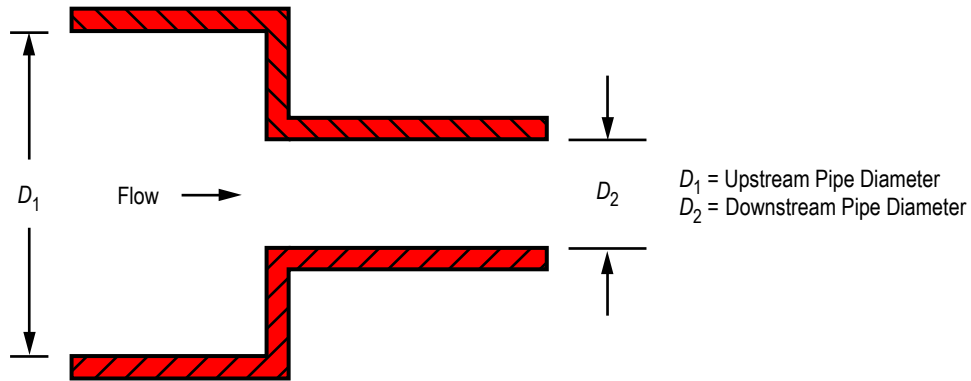
Figure 28. Square reduction resistance option parameters.

3.1.7.8 Square Expansion (Branch Option 8). Figure 29 shows the square expansion resistance option parameters that are required by GFSSP. This option considers the branch as a square expansion. The diameters of the upstream and downstream pipes are D_1 and D_2, respectively. For this option, K_f can be expressed as in equation (33). However, the K_1 in equation (33) is calculated from expressions (40) and (41).[18] The Reynolds number and friction factor that are utilized within these expressions are based on the upstream conditions. If the flow direction is opposite to what is specified in the model input (i.e., the flow rate becomes negative), this option will automatically switch to branch option 8, the Square Expansion (sec. 3.1.7.8).

For upstream Re ≤ 4,000,

$$K_1 = 2\left[1 - \left(\frac{D_1}{D_2}\right)^4\right].$$ (40)

For upstream Re > 4,000:

$$K_1 = \left[1 + 0.8f\right]\left[1 - \left(\frac{D_1}{D_2}\right)^2\right]^2.$$ (41)

This option is recommended for subsonic and incompressible flow.
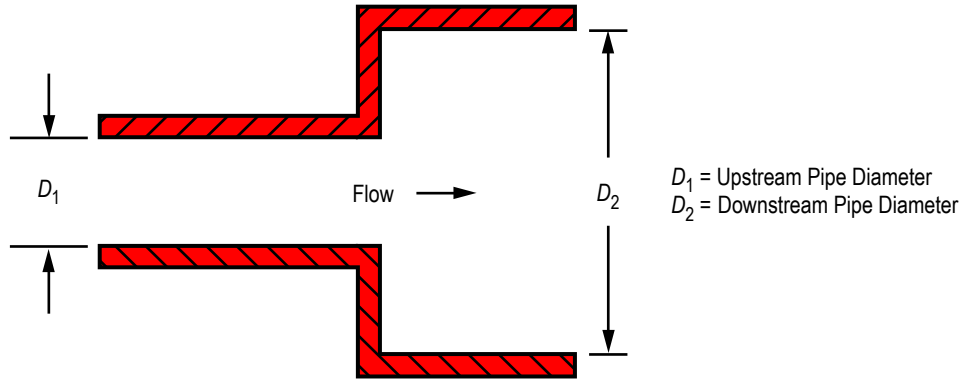
Figure 29. Square expansion resistance option parameters.

3.1.7.9 Rotating Annular Duct (Branch Option 9). Figure 30 shows the rotating annular duct resistance option parameters that are required by GFSSP. This option considers the branch as a rotating annular duct. The length, outer radius, and inner radius of the annular passage are L, r_o, and r_i, respectively. The inner surface is rotating at N rpm ($N = 30\,\omega/\rho$). For this option, K_f can be expressed as:

$$K_f = \frac{fL}{\rho_u \pi^2 A^2 g_c \left(r_o - r_i\right)} \ .$$

(42)

The friction factor (f) in equation (42) was calculated from the following expressions:[19]

$$f_{0T} = 0.077 \left(Ru\right)^{-0.24} \ ,$$

(43)

where

$$Ru = \frac{\rho_u u \ 2\left(r_0 - r_i\right)}{\mu}$$

(44)

and u is the mean axial velocity, therefore:

$$\frac{f}{f_{0T}} = \left[1 + 0.7656 \left(\frac{\omega r_i}{2u}\right)^2\right]^{0.38} \ .$$

(45)
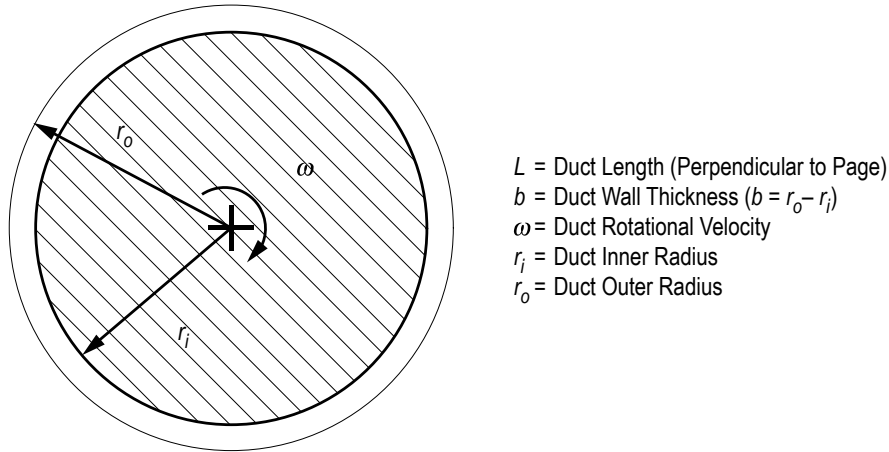
Figure 30. Rotating annular duct resistance option parameters.

3.1.7.10 Rotating Radial Duct (Branch Option 10). Figure 31 shows the rotating radial duct resistance option parameters that are required by GFSSP. This option considers the branch as a rotating radial duct. This option accounts only for the frictional losses encountered with this type of flow. Since centrifugal effects are also important in a rotating radial duct, the user must select this option and activate the rotational term in the momentum conservation equation (eq. (2)). The activation of the rotational term is explained in section 5.3.2 (fig. 68).
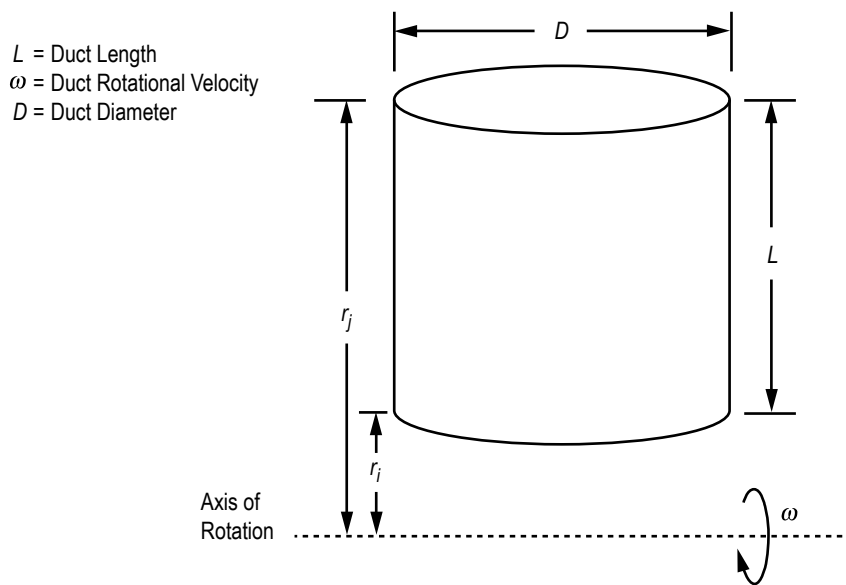
Figure 31. Rotating radial duct resistance option parameters.

The length and diameter of the duct are L and D, respectively. The rotational speed is ω rad/s. For this option, K_f can be expressed as:

$$K_f = \frac{8\,fL}{\rho_u \pi^2 D^5 g_c} \ .$$

(46)

The friction factor (f) in equation (46) was calculated from the following equations:[20]

$$f_{0T} = 0.0791(\text{Re})^{-0.25}$$

(47)

and

$$\frac{f}{f_{0T}} = 0.942 + 0.058 \left[\left(\frac{\omega D}{u} \right) \left(\frac{\omega D^2}{v} \right) \right]^{0.282} \ .$$

(48)

3.1.7.11 Labyrinth Seal (Branch Option 11). Figure 32 shows the labyrinth seal resistance option parameters that are required by GFSSP. This option considers the branch as a labyrinth seal. The number of teeth, clearance, and pitch are n, c, and m, respectively. For this option, K_f can be expressed as:

$$K_f = \frac{\left(1/\varepsilon^2 + 0.5 \right) n + 1.5}{2 g_c \, \rho_u \alpha^2 A^2} \ ,$$

(49)

where the carryover factor (ε) is expressed as:

$$\varepsilon = \sqrt{ \frac{1}{1 - \dfrac{(n-1)\ c\,/\,m_p}{n(c\,/\,m + 0.02)}} } \ .$$

(50)

For a straight labyrinth seal, α should be set to unity. For a stepped labyrinth seal, α should be less than unity. A value of 0.9 has been recommended for many rocket engine turbopump applications.
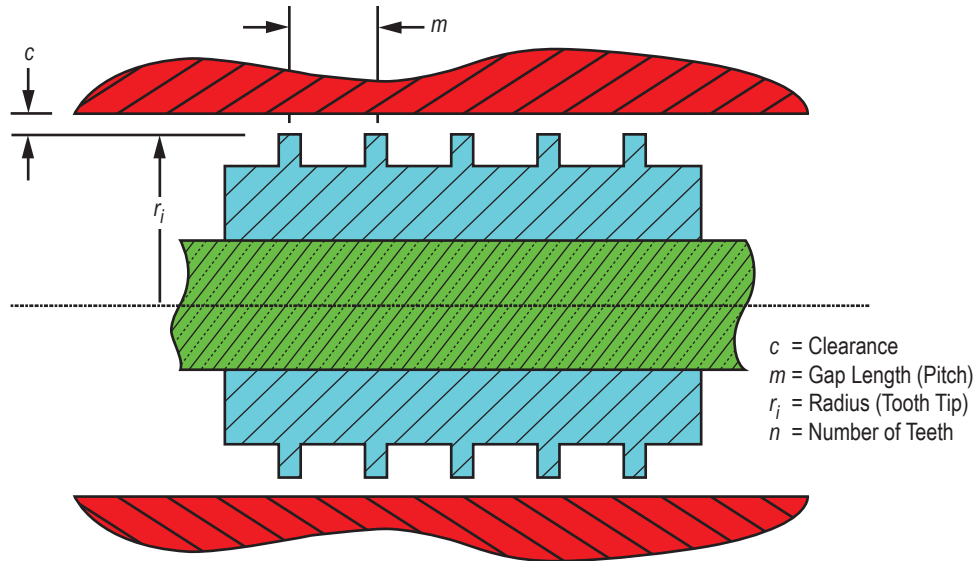
Figure 32. Labyrinth seal resistance option parameters.

3.1.7.12 Flow Between Parallel Plates (Branch Option 12). Figure 33 shows the parallel flat plate resistance option parameters that are required by GFSSP. This option considers the branch as having laminar flow between parallel flat plates. A face seal can be modeled using this option. The flow is assumed to occur between two parallel plates separated by a distance equal to the clearance between the shaft and the housing. The effect of curvature is neglected. The length, inner diameter, and clearance of the seal are L, D, and c, respectively. For this option, K_f can be expressed as:[21]

$$K_f = \frac{12\mu\,L\rho}{\pi g_c D_c^3 |\dot{m}|} \,.$$

(51)
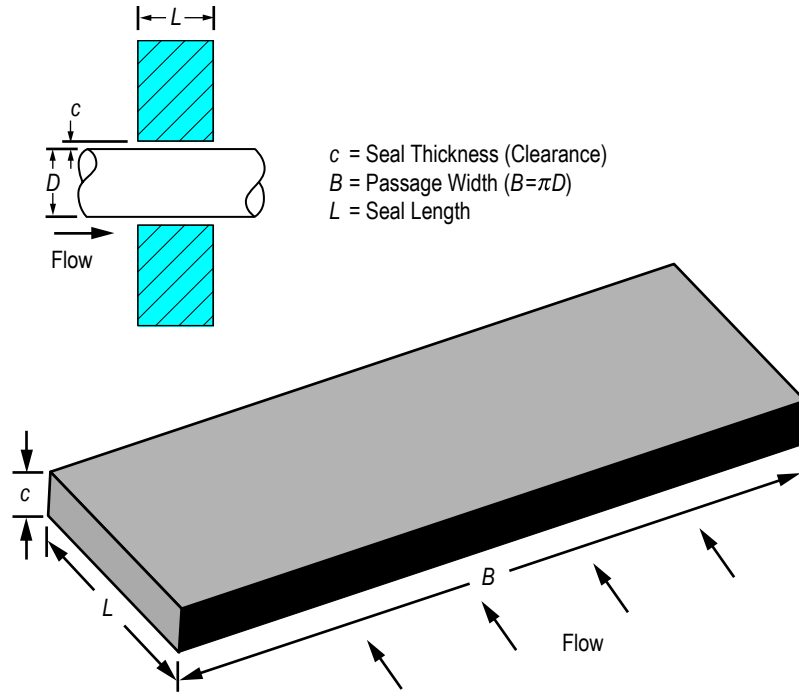
Figure 33. Parallel flat plate resistance option parameters.

3.1.7.13 Common Fittings and Valves (Branch Option 13). This option considers the branch as common fittings or valves. The resistance in common fittings and valves can be computed by the two-K method.[22] For this option, K_f can be expressed as:

$$K_f = \frac{K_1 / \mathrm{Re} + K_\infty (1 + 1/D)}{2 g_c \rho_u A^2} \quad,$$

(52)

where

K_1 = K for the fitting at Re = 1
K_∞ = K for the fitting at Re = ∞
D = internal diameter of attached pipe (in).

The constants K_1 and K_∞ for common fittings and valves are listed in table 7.

Table 7. Constants for two-K method of hooper for fittings/valves[22]
(GFSSP resistance option 13).

Fitting Type			K_1	K_∞
90° elbows	Standard (R/D = 1), screwed		800	0.4
	Standard (R/D = 1), flanged or welded		800	0.25
	Long radius (R/D = 1.5), all types		800	0.2
	Mitered (R/D = 1.5)	1 weld (90° angle)	1,000	1.15
		2 welds (45° angle)	800	0.35
		3 welds (30° angle)	800	0.3
		4 welds (22.5° angle)	800	0.27
		5 welds (18° angle)	800	0.25
45° elbows	Standard (R/D = 1), all types		500	0.2
	Long radius (R/D = 1.5), all types		500	0.15
	Mitered, 1 weld, 45° angle		500	0.25
	Mitered, 2 weld, 22.5° angle		500	0.15
180° elbows	Standard (R/D = 1), screwed		1,000	0.6
	Standard (R/D = 1), flanged or welded		1,000	0.35
	Long radius (R/D = 1.5), all types		1,000	0.3
Tee, used as elbow	Standard, screwed		500	0.7
	Long radius, screwed		800	0.4
	Standard, flanged or welded		800	0.8
	Stub-in-type branch		1,000	1
Tee, flow through	Screwed		200	0.1
	Flanged or welded		150	0.5
	Stub-in-type branch		100	–
Valves	Gate, ball, plug ($\beta = d_{orifice}/d_{pipe}$)	Full line size, $\beta = 1$	300	0.1
		Reduced trim, $\beta = 0.9$	500	0.15
		Reduced trim, $\beta = 0.8$	1,000	0.25
	Globe, standard		1,500	4
	Globe, angle or Y-type		1,000	2
	Diaphragm, dam type		1,000	2
	Butterfly		800	0.25
	Check	Lift	2,000	10
		Swing	1,500	1.5
		Tilting disk	1,000	0.5

3.1.7.14 Pump Characteristics (Branch Option 14). This option considers the branch as a pump with given characteristics. The pump characteristics must be expressed as a curve fit of pressure rise versus flow rate:

$$\Delta p = A_0 + B_0 \dot{m} + C_0 \dot{m}^2 , \tag{53}$$

where

Δp = pressure rise (lbf/ft^2).
$\dot{m}$ = flow rate (lbm/s).

The user must input the intercept A_0, and the first and second order terms, B_0 and C_0, of the curve fit. The momentum source (S) in equation (2) is then expressed as:

$$S = \Delta p \, A . \tag{54}$$

3.1.7.15 Pump Horsepower (Branch Option 15). This option considers the branch as a pump with a given horsepower (P) and efficiency (η). The momentum source (S) in equation (2) is then expressed as:

$$S = \frac{550 \rho_u P \eta \, A}{\dot{m}} . \tag{55}$$

3.1.7.16 Valve With a Given Loss Coefficient (Branch Option 16). This option considers the branch as a valve with a given flow coefficient (C_v). The flow coefficient is the volume (in gallons) of water at 60 °F that will flow per minute through a valve with a pressure drop of 1 psi across the valve. The recommended formula for C_v determination with water is:

$$C_v = Q \sqrt{\frac{1}{\Delta p}} , \tag{56}$$

where Q is the volumetric flow rate in gpm of water at 60 °F and Δp is the pressure drop in psia. For this option, K_f can be expressed as:

$$K_f = \frac{4.68 \times 10^5}{\rho_u C_v^2} . \tag{57}$$

3.1.7.17 Joule-Thompson Device (Branch Option 17). This option considers the branch as a Viscojet,[23] which is a specific type of flow resistance with relatively large flow passages with very high pressure drops. The flow rate through the Viscojet is given by:

$$w = 10{,}000 \, k_v \frac{V_f}{L_\Omega} \sqrt{\Delta p \, S} \, (1 - x) , \tag{58}$$

where w is the flow rate in lbm/hr, L_Ω is the resistance of the fluid device, k_v is an empirical factor, and V_f is the viscosity correction factor.

For this option, K_f can be expressed as:

$$K_f = \frac{18.6624}{S} \left(\frac{L_\Omega}{V_f k_v (1 - x)} \right)^2 . \tag{59}$$

3.1.7.18 Control Valve (Branch Option 18). This is an exclusively transient option that considers the branch as a control valve that monitors the pressure at some arbitrary point downstream

of the valve and opens and closes to maintain that pressure within a prescribed tolerance. This option was originally developed for use with the pressurization option to model on/off, or 'bang-bang,' pressurization systems as shown in figure 34. The valve is regarded as a flow restriction with a given flow coefficient (C_L) and area (A), and uses equation (24) to calculate K_f for the valve.
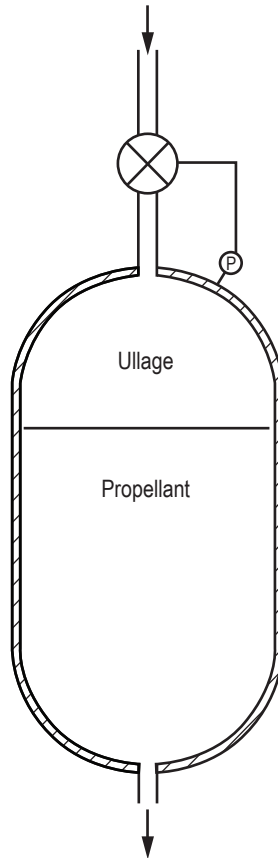
Figure 34. Control valve in a pressurization system.

The remaining inputs for the control valve option define its open/close characteristics. The suboption determines the type of open/close response of the valve (instantaneous, linear, or nonlinear) and the valve initial position describes whether the valve is initially open or closed. The control node defines the location in the model where the control valve option is to monitor and maintain the pressure while the pressure tolerance file name provides the code with the name of the file containing the pressure tolerance data for the control valve.

For a linear open/close response, the time to open/close and the number of time steps needed to complete that response are provided as additional inputs. If the length of the valve open/close time steps does not match the global time step, the program will temporarily change the global time step until the valve has fully opened/closed. The area of the valve will be linearly interpolated at each time step.

Finally, for a nonlinear open/close response, the file names for the open and close characteristics of the valve are required as additional inputs. The files detail the valve C_L and area as functions of time from the start of valve opening/closing.

3.1.7.19 User-Defined Resistance (Branch Option 19). This option allows users to create a new resistance option that is not available in the GFSSP library. Once this option is chosen, the user is required to supply the coding for calculating K_f for this option in the User Subroutine to be described in the following section. In the preprocessor, the user is required to supply the cross-sectional area of the branch.

3.1.7.20 Heat Exchanger Core (Branch Option 20). This option considers branch as a heat exchanger core. In a typical heat exchanger core (fig. 35), the fluid goes past a tube bank to allow heat transfer between fluids in the main duct and fluids within the tubes. The free flow area is reduced and there is a larger surface area of contact between the fluid and solid walls. Sections 1 and 2 in figure 35 represent inlet and outlet of the heat exchanger core, respectively. The pressure drop through the heat exchanger core can be expressed as:[24]

$$\Delta p = \frac{G^2}{2g_c} v_1 \left[\left(K_c + 1 - \sigma^2 \right) + 2 \left(\frac{v_2}{v_1} - 1 \right) + f \frac{A_s}{A_c} \frac{v_m}{v_1} - \left(1 - \sigma^2 - K_e \right) \frac{v_2}{v_1} \right], \tag{60}$$

where $G \left(= \dfrac{\dot{m}}{A} \right)$ is the mass flux.

Figure 35. Heat exchanger core.

Equation (60) can be rewritten as:

$$\Delta p = \frac{1}{2\rho_1 g_c A_c^2} \left[\left(K_c + 1 - \sigma^2 \right) + 2 \left(\frac{\rho_1}{\rho_2} - 1 \right) + f \frac{A_s}{A_c} \frac{\rho_1}{\rho_{avg}} - \left(1 - \sigma^2 - K_e \right) \frac{\rho_1}{\rho_2} \right] \dot{m}^2 . \tag{61}$$

Therefore, K_f can be expressed as:

$$K_f = \frac{\left(K_c + 1 - \sigma^2 \right) + 2 \left(\frac{\rho_1}{\rho_2} - 1 \right) + f \frac{A_s}{A_c} \frac{\rho_1}{\rho_{avg}} - \left(1 - \sigma^2 - K_e \right) \frac{\rho_1}{\rho_2}}{2\rho_1 g_c A_c^2} . \tag{62}$$

3.1.7.21 Parallel Tube (Branch Option 21). This option considers the branch as a parallel tube where fluid flows through n number of tubes (fig. 36). The flow is assumed to be distributed uniformly in all tubes. This resistance option calculates the pressure drop in the parallel tube. For this option, K_f can be expressed as:

$$K_f = \frac{8 f L}{\rho_u \pi^2 D^5 g_c n^2} . \tag{63}$$

Figure 36. Parallel tube.

3.1.7.22 Compressible Orifice (Branch Option 22). This option considers the branch as an orifice for compressible flow. In this option, unlike other options, flow rate is calculated from a simplified momentum equation (eq. (3c)). There is no need to calculate K_f for this option. The input to this option is identical to option 2 (flow-through restriction).

3.1.7.23 Labyrinth Seal/Egli Correlation (Branch Option 23). This option provides an alternative formulation of flow resistances through the labyrinth seal (branch option 11). Egli developed a formulation of flow-through labyrinth seal based on the actual flow characteristics typical for a sharp-edged orifice.[25] Based on the general relations between leakage and number of throttling passages, flow resistance was calculated. The formulation also considered the effect of kinetic energy being carried from one throttling into the next. The input parameters (fig. 32) for this option are inner radius, clearance, pitch, number of teeth, and tooth tip width.

3.1.7.24 Fixed Flow (Branch Option 24). This option fixes flow rate in a given branch. The fixed flow branch can only be located adjacent to a boundary node. With this new option, the user can prescribe either pressure or flow rate as boundary condition.

The fixed flow rate option has been implemented by introducing pump characteristics (fig. 37) of a positive displacement pump.
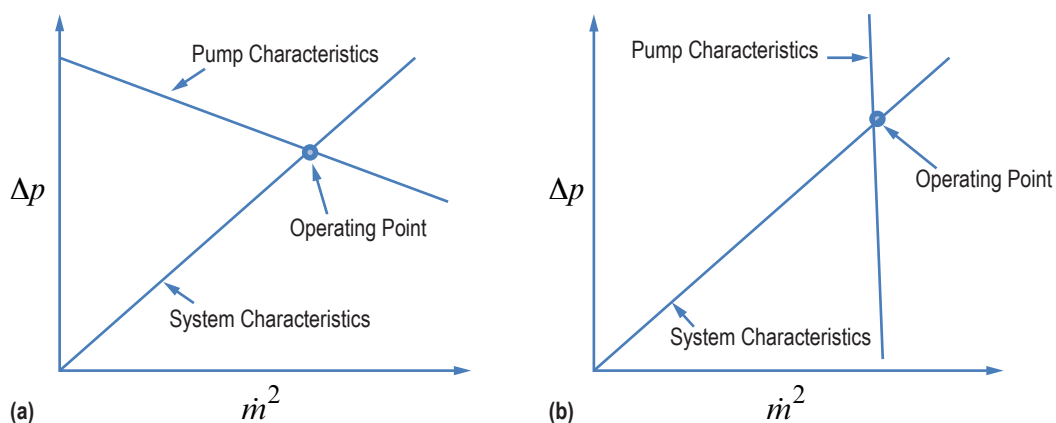
Figure 37. Implementation of fixed flow rate in a branch using pump characteristics: (a) Centrifugal pump and (b) positive displacement pump.

The pump characteristics of a positive displacement pump are expressed as: $\Delta p = A + C\dot{m}|\dot{m}|$, where $A = \alpha\,\dot{m}|\dot{m}|$; $C = -\alpha$, where $\alpha = 1 \times 10^{25}$. Substituting A and C, one gets $\dot{m} = \dot{m}|\dot{m}| / \dot{m}$.

There is a word of caution about the use of this option. The calculated pressure field may not be realistic as flow rate is calculated independent of pressure differential.

3.2 Solution Procedure

GFSSP numerically solves the governing equations described in the previous section to compute pressure, temperature, flow rate, and other fluid properties in a given flow circuit. The mathematical closure is described in table 8 where each variable and the designated governing equation to solve that variable are listed. It may be noted that the pressure is calculated from the mass conservation equation although pressure does not explicitly appear in equation (1). This is, however, possible in the iterative scheme where pressures are corrected to reduce the residual error in mass conservation equation. This practice was first implemented in a semi-implicit pressure linked equation (SIMPLE) algorithm proposed by Patankar and Spalding[26] and commonly referred to as 'pressure based' algorithm in computational fluid dynamics literature. The momentum conservation equation (eq. (2)) which contains both pressure and flow rate is solved to calculate the flow rate. The strong coupling of pressure and flow rate requires that mass and momentum conservation equations are solved simultaneously.

Table 8. Mathematical closure.

Variable Number	Variable Name	Designated Equation to Solve the Variable
1	Pressure	Mass conservation
2	Flow rate	Momentum conservation
3	Fluid enthalpy or entropy	Energy conservation of fluid
4	Solid temperature	Energy conservation of solid
5	Species concentration	Species conservation
6	Fluid mass	Thermodynamic state

The energy conservation equation can either be expressed in terms of enthalpy or entropy. The temperature, density, and other thermophysical properties, such as viscosity, conductivity, and specific heats are computed from pressure and enthalpy or entropy using thermodynamic property programs, GASP[10]/WASP[11] or GASPAK.[12] In flow circuits where solid to fluid heat transfer is present, the energy conservation equation for solids is solved to calculate the solid temperature. The rate of heat transfer between solid to fluid appears as a source or sink term in the energy conservation equations of fluid and solid.

For a mixture, the conservation equations of species are solved to compute the mass fraction of species. There are three options for solving the energy equation for a mixture as discussed in section 3.1.3.2. The method of calculating thermodynamic and thermophysical peoperties of all three options has been described in section 3.1.6.

For a transient problem, fluid mass is required in mass and momentum conservation equations (eqs. (1) and (2)). GFSSP uses the thermodynamic equation of state (eq. (10)) to calculate resident mass in an internal node where density, compressibility factor, and temperature are computed.

There are two types of numerical methods available to solve a set of nonlinear coupled algebraic equations: the successive substitution method and the Newton-Raphson method. In the successive substitution method, each equation is expressed explicitly to calculate one variable. The previously calculated variable is then substituted into the other equations to calculate another variable. In one iterative cycle, each equation is visited. The iterative cycle is continued until the difference in the values of the variables in successive iterations becomes negligible. The advantages of the successive substitution method are its simplicity to program and its low code overhead. The main limitation, however, is finding an optimum order for visiting each equation in the model. This visiting order, which is called the information flow diagram, is crucial for convergence. Underrelaxation (partial substitution) of variables is often required to obtain numerical stability. Details of the successive substitution method appear in appendix B.

In the Newton-Raphson method, the simultaneous solution of a set of nonlinear equations is achieved through an iterative guess and correction procedure. Instead of solving for the variables directly, correction equations are constructed for all of the variables. The intent of the correction equations is to eliminate the error in each equation. The correction equations are constructed in two steps: (1) The residual errors in all of the equations are estimated and (2) the partial derivatives of all of the equations, with respect to each variable, are calculated. The correction equations are then solved by the Gaussian elimination method. These corrections are then applied to each variable, which completes one iteration cycle. These iterative cycles of calculations are repeated until the residual error in all of the equations is reduced to a specified limit. The Newton-Raphson method does not require an information flow diagram; therefore, it has improved convergence characteristics. The main limitation to the Newton-Raphson method is its requirement for a large amount of computer memory. Details of the Newton-Raphson method appear in appendix C.

In GFSSP, a combination of the successive substitution method and the Newton-Raphson method is used to solve the set of equations. This method is called SASS (simultaneous adjustment with successive substitution) (fig. 38). In this scheme, the mass and momentum conservation equations are solved by the Newton-Raphson method. The energy and specie conservation equations are solved by the successive substitution method. The underlying principle for making such a division was that the equations that are more strongly coupled are solved by the Newton-Raphson method. The equations that are not strongly coupled with the other set of equations are solved by the successive substitution method. Thus, the computer memory requirement can be significantly reduced while maintaining superior numerical convergence characteristics.

Iteration Loop	Governing Equations	Variables
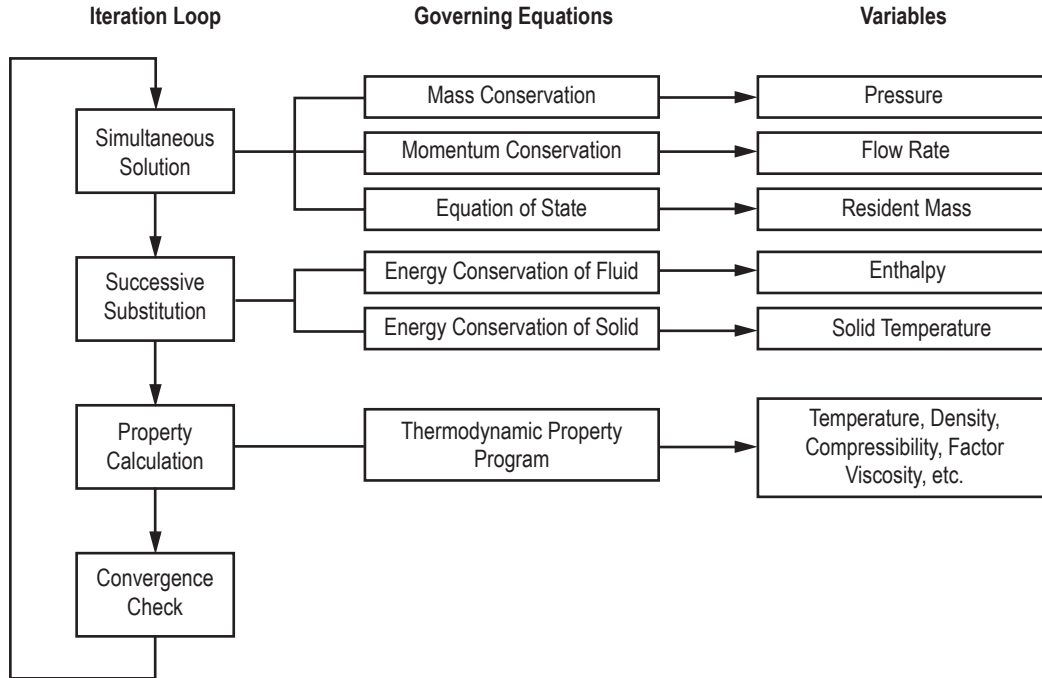
Figure 38. SASS scheme for solving governing equations.

SASS has two options available. In one option, there are two iterative loops—inner and outer. In the inner iterative loop, mass and momentum conservation equations are solved by the Newton-Raphson scheme. For unsteady formulation, the equation of state is also solved by the Newton-Raphson scheme in addition to mass and momentum conservation equations. In the outer loop, the energy and specie conservation equations are solved by the successive substitution method. The outer loop also calculates the density and other thermodynamic and thermophysical properties and the flow resistance coefficient (K_f) which is a function of density. This option is called nonsimultaneous option. The total number of iterations in this option can be expressed as:

$$N_{total} = \sum_{i=1}^{n_o} n_i \ , \tag{64}$$

where n_o is the number of outer iterations and n_i is the number of inner iterations. The inner iterative cycle is terminated when the normalized maximum correction (Δ_{max}) is less than the convergence criterion (C_c). Δ_{max} is determined from

$$\Delta_{max} = \text{MAX} \left| \sum_{i=1}^{N_E} \frac{\Phi'_i}{\Phi_i} \right| \ , \tag{65}$$

where

N_E = the total number of equations solved by the Newton-Raphson scheme (N_E = number of nodes + number of branches (steady flow)

N_N = number of nodes × 2 + number of branches (unsteady flow).

The outer iteration is terminated when Δ_{max}^o is less than the convergence criterion (C_c). Δ_{max}^o is determined from

$$\Delta_{max}^o = \text{MAX}\left|\Delta K_f, \Delta_\rho, \Delta_h, \text{ or } \Delta_s\right| , \tag{66}$$

where

$$\Delta K_f = \text{MAX}\left|\sum_{i=1}^{N_B} \frac{K'_f}{K_f}\right|$$

$$\Delta_\rho = \text{MAX}\left|\sum_{i=1}^{N_N} \frac{\rho'}{\rho}\right|$$

$$\Delta_h = \text{MAX}\left|\sum_{i=1}^{N_N} \frac{h'}{h}\right|$$

$$\Delta_s = \text{MAX}\left|\sum_{i=1}^{N_N} \frac{s'}{s}\right| . \tag{67}$$

N_B and N_N are the number of branches and nodes, respectively, in a flow circuit.

In the second option, there is only one iterative loop. During the iterative cycle mass, momentum and the equation of state are first solved by the Newton-Raphson scheme. Then the energy and specie conservation equations are solved by the successive substitution method. The iterative cycle is terminated when the normalized maximum correction (Δ_{max}) is less than the convergence criterion (C_c).

This option is called the simultaneous option and is more efficient than the nonsimultaneous option. The nonsimultaneous option, however, is more numerically stable. With the help of a logical variable, SIMUL, the user can switch between the first and second options. More detailed discussion of both options appears in section 4.

4. COMPUTER PROGRAM

The purpose of this section is to describe the structure and major functions of the program. The main objective of the computer program is to implement the numerical algorithm described in section 3 in a way which is easy to follow, modular to allow for future extension, robust, and free of errors. There are seven major functions of the computer program:

(1) Development of a flow circuit with fluid and solid nodes with branches and conductors.
(2) Development of an indexing system or data structure to define a network of fluid and solid nodes with branches and conductors.
(3) Generation of conservation equations of fluid mass, momentum, energy and species and solid temperatures in respective nodes and branches.
(4) Calculation of thermodynamic and thermophysical properties of the fluid and solid in nodes.
(5) Numerical solution of conservation equations.
(6) Input/output.
(7) User-defined modules.

GFSSP consists of three major modules: Graphical User Interface (GUI module), Solver and Property (SP) module, and User Subroutine (US) module (fig. 39). Functions (1) and (6) are done in the GUI; functions (2) through (6) are done in SP module, and function (7) is done in US module. A distinct boundary is maintained among GUI, SP, and US modules. The GUI and US modules supply the information to the SP module through an input data file and User Subroutines. The SP module returns an output data file and plot files for graphical and text display of results. The maintenance of a strict boundary among the three modules is a key feature of GFSSP that makes the code easy to use, maintain, and upgrade. Users are not required to know the details of the computational method to become a proficient user of the code. The modularity also helps the developer to add new capabilities with minimum impact to the existing code. This section describes the SP and US modules. The GUI module is described in section 5.
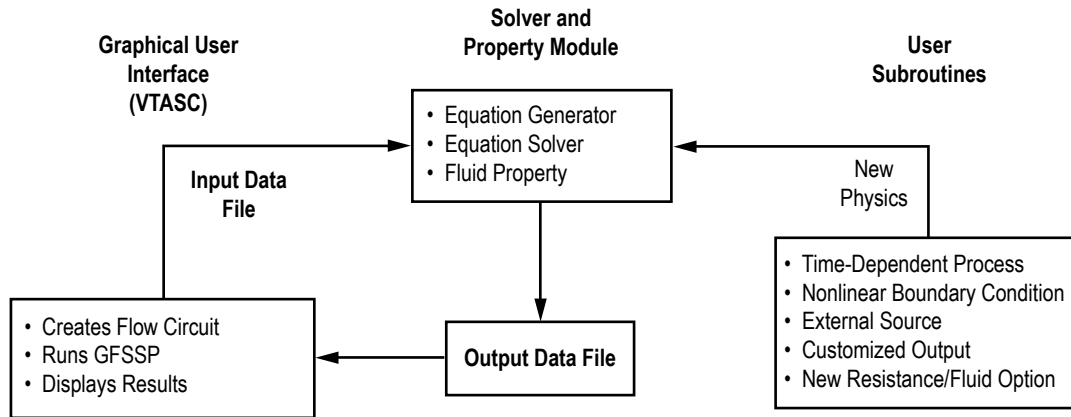
Figure 39. GFSSP process flow diagram showing interaction among three modules.

4.1 Process Flow Diagram

Figure 39 shows GFSSP's process flow diagram to describe the interaction among GUI, SP, and US modules. Users create a flow circuit in the GUI, visual thermofluid analyzer for system and components (VTASC) by a 'point, drag, and click' method. VTASC creates an input data file that is read by the SP module. The user runs the SP module from VTASC, which also reads the output data file generated by the SP module to display the results in the GUI. The VTASC also allows users to plot time-dependent results in Winplot.[13] Specialized input to the model can be supplied through User Subroutines that also interact with the SP module. Such specialized input includes time-dependent processes, nonlinear boundary conditions, and external mass, momentum, and energy sources, customized output and new resistance and fluid options.

4.2 Solver and Property Module

The main routine and the associated set of subroutines perform seven major functions that include:

(1) Reading of the input data file generated by VTASC, GFSSP's graphical user interface.
(2) Generation of the trial solution based on the initial guess.
(3) Supply time-dependent boundary conditions for unsteady flow.
(4) Numerical solution of conservation equations by the SASS scheme.
(5) Interaction with thermodynamic property programs to calculate properties at nodes.
(6) Calculation of flow resistances in the branches.
(7) Create text output and plot files.

The flow charts of the SP module for nonsimultaneous and simultaneous schemes are shown in figures 40 and 41, respectively.
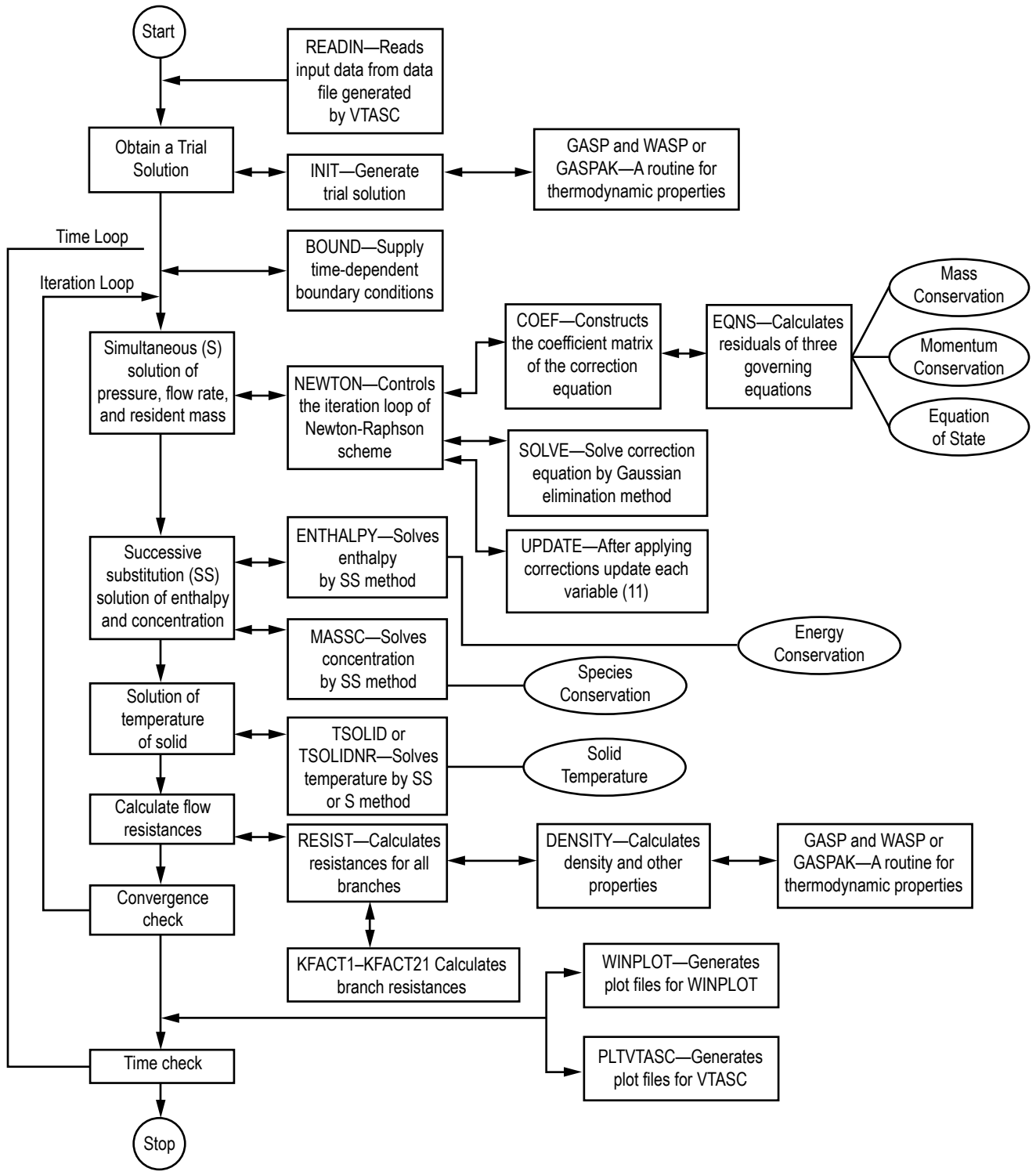
Figure 40. Flowchart of nonsimultaneous solution algorithm in SP module.
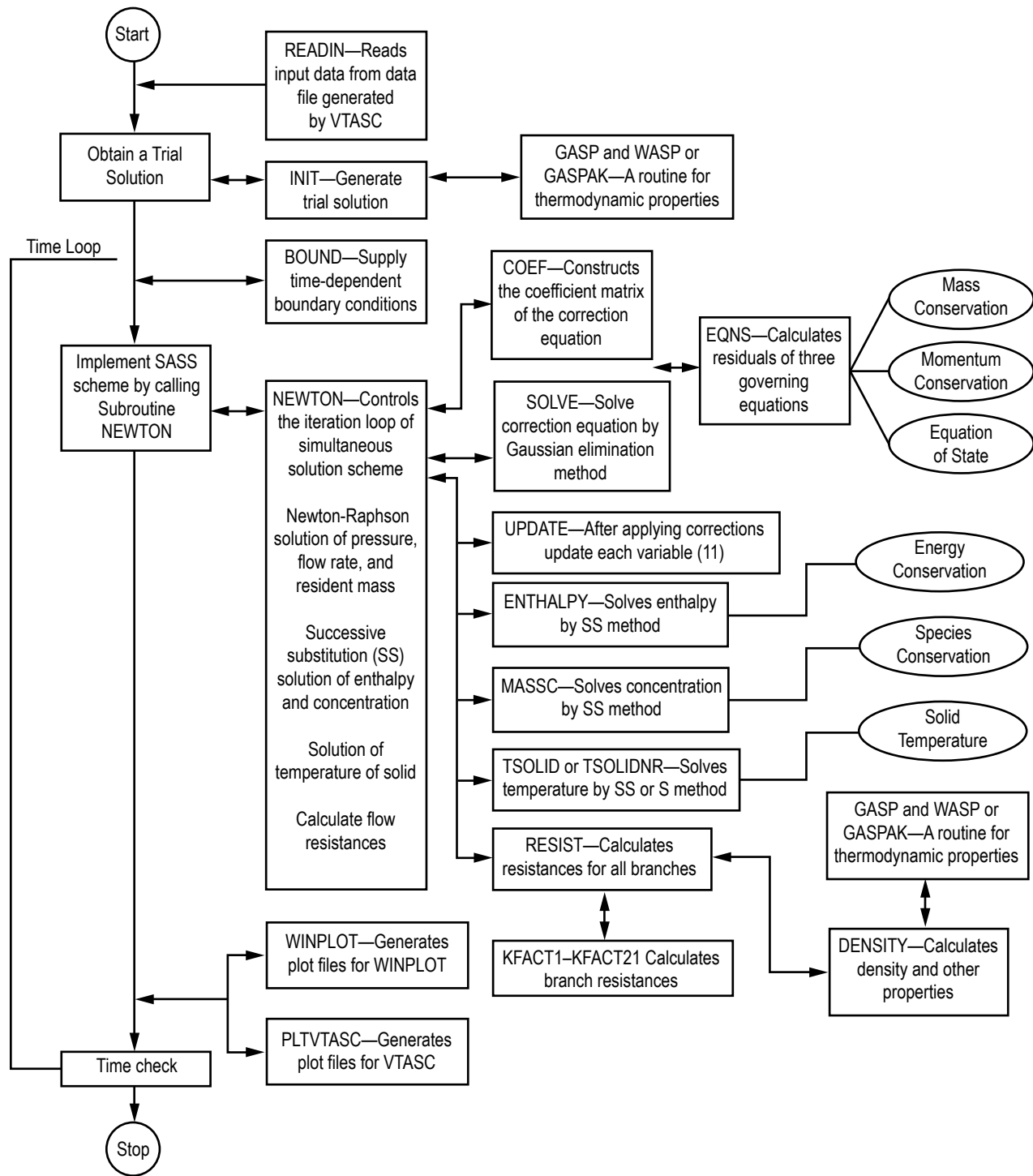
Figure 41. Flowchart of simultaneous solution algorithm in SP module.

4.2.1 Nonsimultaneous Solution Scheme

The flow chart of nonsimultaneous solution algorithm is shown in figure 40. In this scheme, there are two iteration cycles— inner and outer. In the inner iteration cycle, the mass and momentum conservation equation and resident equation of state are solved by the Newton-Raphson scheme until convergence. During this iteration cycle, property and resistance coefficients are not updated. In the outer iteration cycle, the energy and concentration equations are solved and density and resistance coefficients are updated and a new set of Newton-Raphson iteration is started. The outer iteration loop is repeated until the fractional change in density, resistance coefficient, and enthalpy is negligible.

The subroutine READIN reads the input data file. The subroutine INIT generates a trial solution by interacting with the thermodynamic property codes GASP, WASP, and GASPAK, or the property tables. Subroutine BOUND reads any applicable time-dependent boundary conditions from the model history files. Subroutine NEWTON conducts the Newton-Raphson solution of the mass conservation, flow rate, and energy conservation equations with the help of the subroutines EQNS, COEF, SOLVE, and UPDATE. The subroutine EQNS generates the equations. The coefficients of the correction equations are calculated in COEF. The correction equations are solved by the Gaussian Elimination method in SOLVE. After applying the corrections, the variables are updated in subroutine UPDATE. This cycle of calculations is repeated until the corrections are negligible. The energy conservation equation is then solved in subroutine ENTHALPY or ENTROPY by the successive substitution scheme. For problems involving fluid mixture subroutine, MASSC is called to solve species conservation equations. For conjugate heat transfer problems, the energy conservation equations for solid nodes are solved in subroutine TSOLID or TSOLIDNR. The resistance for each branch is calculated in RESIST following the calculation of fluid densities at each node in the subroutine DENSITY. The flow resistance coefficients (K_f) for each branch are computed in subroutines KFACT1 through KFACT24 depending upon the resistance option selected for a particular branch. The convergence of the numerical scheme is checked to determine if the cycle of calculation needs to be repeated. The solver module also calls 23 User Subroutines from various subroutines as described in section 4.3.

4.2.2 Simultaneous Solution Scheme

The flowchart of the simultaneous solution algorithm is shown in figure 41. The functionality of subroutine READIN, INIT, and BOUND is identical to the nonsimultaneous scheme. In this scheme, there is only one iteration loop. The enthalpy (or entropy), concentrations, density, and resistance coefficient are updated in each Newton-Raphson iteration. Therefore, in each Newton-Raphson iteration, subroutine ENTHALPY or ENTROPY, MASSC, TSOLID or TSOLIDNR, RESIST, and DENSITY are called to compute and update all variables. The iteration loop is controlled in subroutine NEWTON. The interaction of the SP and US modules is identical to the nonsimultaneous scheme.

64

4.2.3 Conjugate Heat Transfer

GFSSP can model solid to fluid heat transfer which is commonly known as conjugate heat transfer. There are two solution options for solving the conservation equation for solid nodes: successive substitution and Newton-Raphson. The successive substitution scheme is implemented in subroutine TSOLID and the flowchart of the subroutine is shown in figure 42. Subroutine TSOLID calls STCOND, CONVHC, RADHCF, RADHCSA, RADHCSSR, and SLDCP to estimate different terms of the conservation equation. STCOND determines the thermal conductivity from the property table. CONVHC determines the heat transfer coefficient from GFSSP's built-in correlation. CONVHC also calls subroutine USRHCF to allow the user to provide a problem-specific heat transfer coefficient. RADHCF, RADHCSA, and RADHCSSR are three subroutines called from TSOLID to compute radiation heat transfer from solid to fluid, solid to ambient, and solid to solid, respectively. SLDCP determines the specific heat from the property table for computing the transient term. Subroutine QDOTSSCR calculates solid to solid conduction and radiation heat transfer.
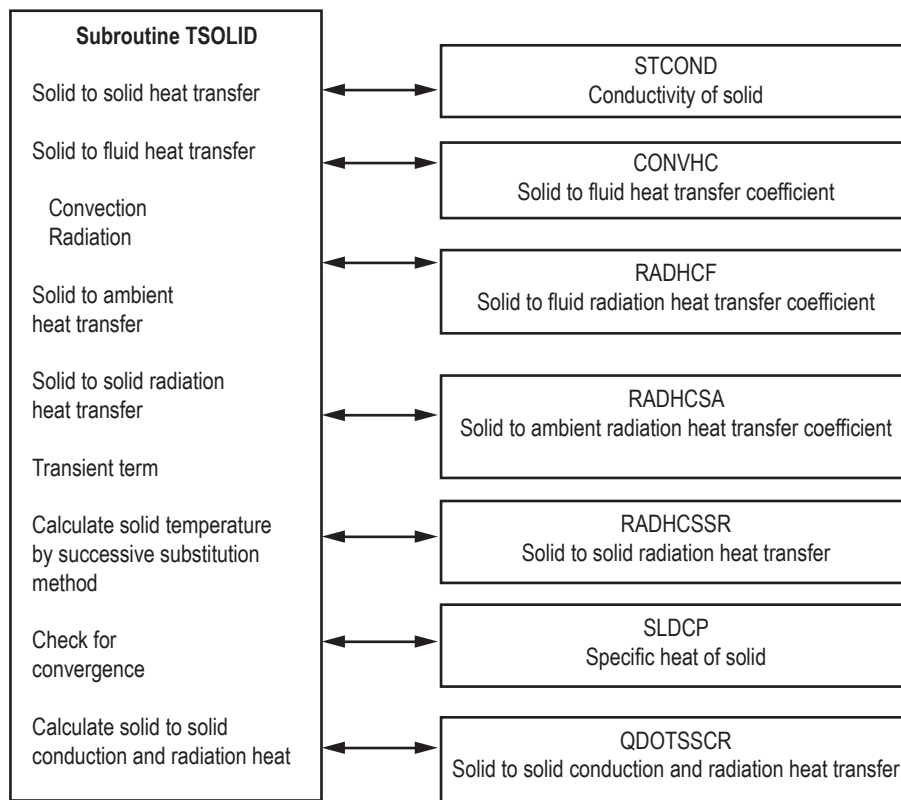
Figure 42. Solid temperature calculation by successive substitution method.

The Newton-Raphson scheme is implemented in subroutine TSOLIDNR. The flowchart of this subroutine is shown in figure 43. TSEQNS, TSCOEF, and GAUSSY are three subroutines that perform the major functions of the Newton-Raphson scheme. The residuals are calculated in TSEQNS. The flowchart of subroutine TSEQNS is shown in figure 44. TSCOEF calculates the coefficient matrix. The correction equations are solved in GAUSSY. After convergence of the numerical scheme, subroutine QDOTSSCR calculates solid to solid conduction and radiation heat transfer.
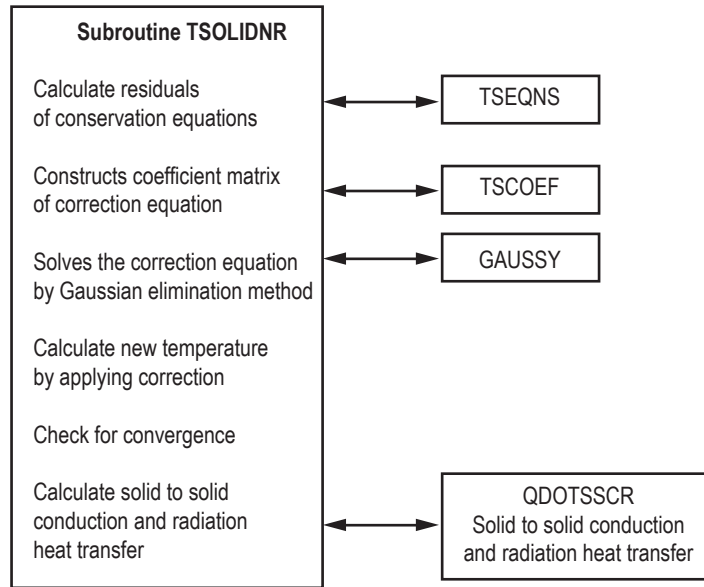
Figure 43. Solid temperature calculation by Newton-Raphson method.
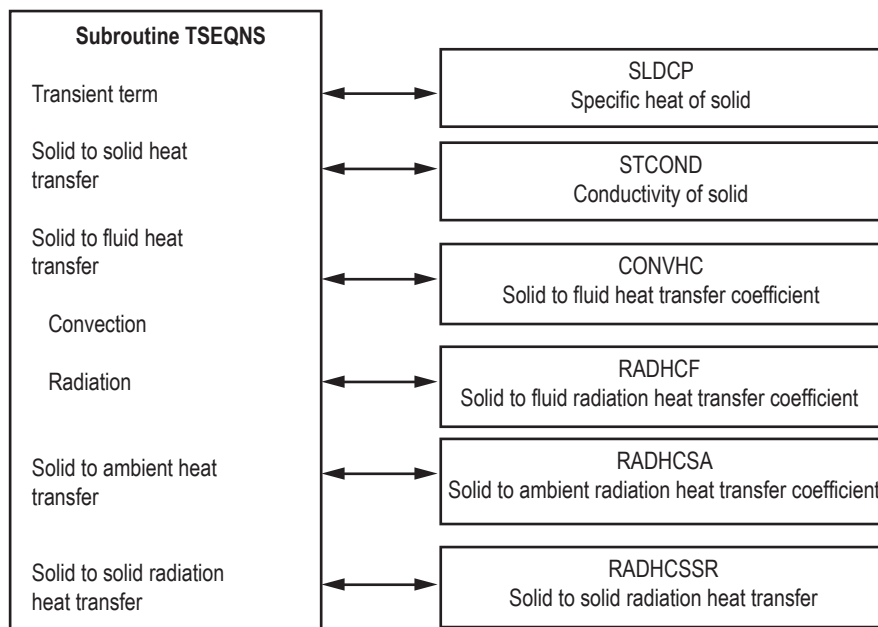
Figure 44. Calculation of residuals of energy conservation equation for Newton-Raphson method.

4.2.4 Thermodynamic Property Package

The thermodynamic property packages included in GFSSP consist of three separate programs: GASP,[10] WASP,[11] and GASPAK[12]; it also includes tabulated data for RP-1. The GASP and WASP programs consist of a number of subroutines. GASP provides the thermodynamic properties for 10 fluids: helium, methane, neon, nitrogen, carbon monoxide, oxygen, argon, carbon dioxide, fluorine, and hydrogen. WASP provides the thermodynamic properties of water. RP-1 properties are provided in the form of tables. Subroutine RP1 searches for the required property values from these tables. GASPAK provides thermodynamic properties for helium, methane, neon, nitrogen, carbon monoxide, oxygen, argon, carbon dioxide, hydrogen, parahydrogen, water, isobutane, butane, deuterium, ethane, ethylene, hydrogen sulfide, krypton, propane, xenon, R-11, R-12, R-22, R-32, R-123, R-124, R-125, R-134A, R-152A, nitrogen trifluoride, and ammonia.

The thermodynamic property subroutines are called from two GFSSP subroutines: INIT and DENSITY. In subroutine INIT, enthalpies and densities are computed from given pressures and temperatures at the boundary and internal nodes. In subroutine DENSITY, density, temperatures, specific heats, and specific heat ratios are calculated from given pressures and enthalpies at each node.

4.3 User Subroutines

Experienced users have the ability to introduce additional capability into the code through User Subroutines. Twenty-three User Subroutines are called from various locations of the solver module. The caller and called subroutines are shown in figure 45. All necessary GFSSP variables are available through COMMON BLOCK and subroutine arguments. Users can develop many additional capabilities by developing their own code in User Subroutines. These may include the following:

(1) Heat or mass transfer model in any node of a circuit.
(2) External forces applied on the fluid in any branch of the circuit. Users also have the ability to modify the existing formulation of various forces already existing in the code.
(3) Variable time step, geometry, and boundary conditions for a time-dependent problem.
(4) New resistance or fluid options.
(5) Develop customized output and/or plot file.

Appropriate use of User Subroutines requires some familiarity with GFSSP variables and indexing practice. Common block variables are explained in appendix D. GFSSP indexing practice and User Subroutines are explained in the following sections. The use of User Subroutines has also been demonstrated in example 10.
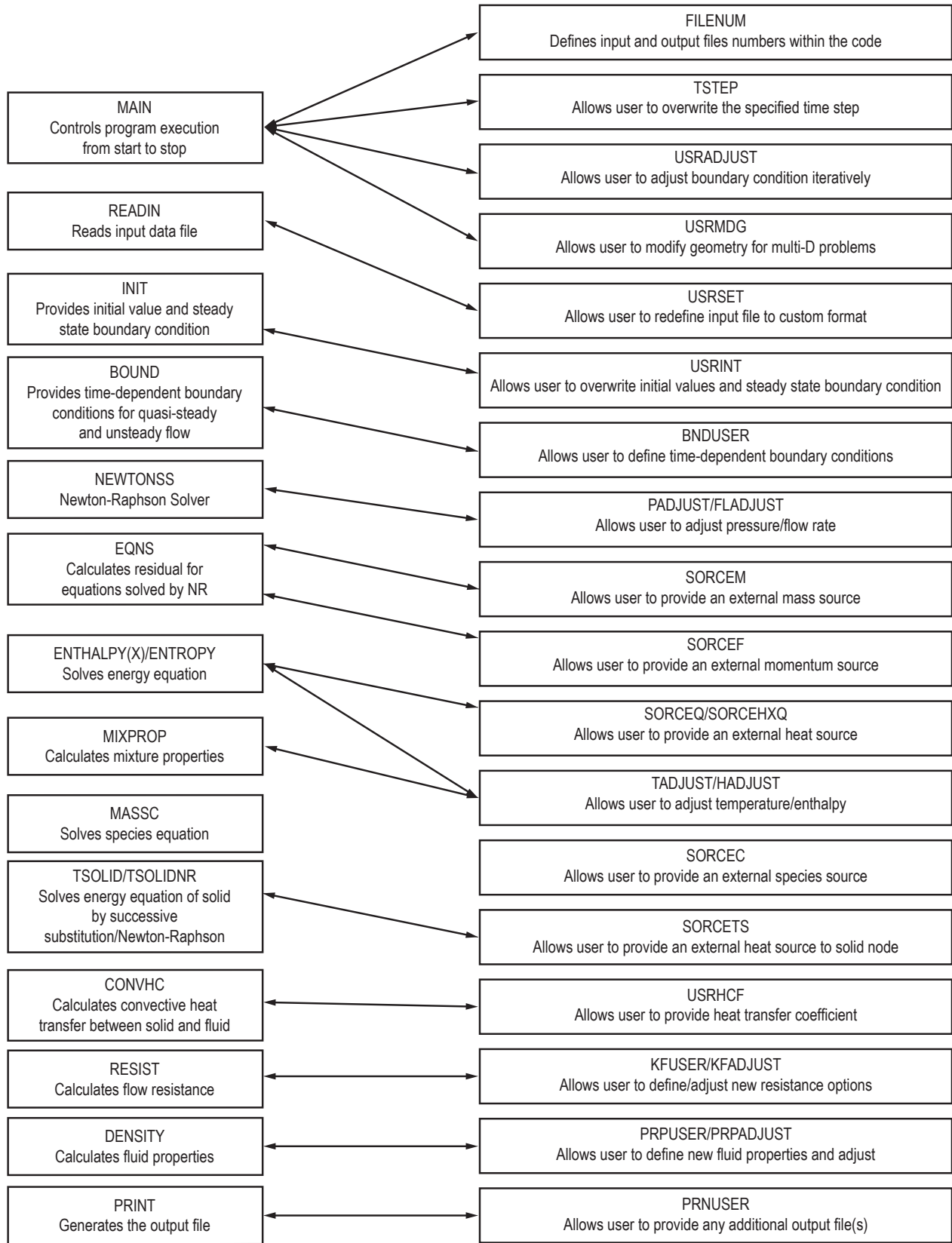
Figure 45. Interaction of User Subroutine with solver module.

4.3.1 Indexing Practice

Users who will be using User Subroutines to add new features into the code need to understand the indexing practice. In order to develop the coding to incorporate new features, users need to access different variables at nodes and branches. All variables are stored in one-dimensional arrays. The description of all node variables appears in sections 2 and 3 of appendix D. Section 4 describes all branch variables.

4.3.1.1 Fluid Node. User-defined fluid node names are stored in NODE-array. NODE-array includes both internal and boundary nodes. The total number of elements in NODE-array is NNODES, which represents the total number of nodes in a given model. Subroutine INDEXI finds the address location for a given node.

SUBROUTINE INDEXI (NUMBER, NODE, NNODES, IPN)

Input variables:
NUMBER: *Node* Number
NODE: Array for storing *Node*
NNODES: Number of *Nodes*

Output variable:
IPN: Location of *Node* in Array (Pointer)

4.3.1.2 Branch. User-defined branch numbers are stored in IBRANCH-array. The total number of elements in the IBRANCH-array is NBR, which represents the total number of branches in a given model. Subroutine INDEXI is also used to find the address location for a given branch.

SUBROUTINE INDEXI (NUMBER, IBRANCH, NBR, IB)

Input variables:
NUMBER: *Branch* Number
IBRANCH: Array for storing *Branch* Number
NBR: Number of *Branches*

Output variable:
IB: Location of *Branch* in Array (Pointer)

4.3.1.3 Solid Node. User-defined solid node numbers are stored in NODESL-array. The total number of elements in the NODESL-array is NSOLIDX, which represents the total number of solid nodes in a given model. Subroutine INDEXS finds the address location for a given solid node.

SUBROUTINE INDEXS (NUMBER, NODESL, NSOLIDX, IPSN)

Input variables:
NUMBER: Solid Node number
NODESL: Array for storing Solid Node Number
NSOLIDX: Number of Solid Nodes

Output variable:
IPSN: Location of Solid Node in Array (Pointer)

4.3.1.4 Ambient Node. User-defined ambient node numbers are stored in NODEAM-array. The total number of elements in the NODEAM-array is NAMB, which represents the total number of ambient nodes in a given model. Subroutine INDEXA finds the address location for a given ambient node.

SUBROUTINE INDEXA (NUMBER, NODEAM, NAMB, IPAN)

Input variables:
NUMBER: Ambient Node Number
NODEAM: Array for storing Ambient Node Number
NAMB: Number of Ambient Nodes

Output variable:
IPAN: Location of Ambient Node in Array (Pointer)

4.3.1.5 Solid to Solid Conductor. User-defined solid to solid conductor numbers are stored in an ICONSS-array. The total number of elements in the ICONSS-array is NSSC, which represents the total number of solid to solid conductor in a given model. Subroutine INDEXSSC finds the address location for a given solid to solid conductor.

SUBROUTINE INDEXSSC (NUMBER, ICONSS, NSSC, ICSS)

Input variables:
NUMBER: Solid to Solid Conductor Number
ICONSS: Array for storing Solid to Solid Conductor Number
NSSC: Number of Solid to Solid Conductors

Output variable:
ICSS: Location of Solid to Solid Conductor in Array (Pointer)

4.3.1.6 Solid to Fluid Conductor. User-defined solid to fluid conductor numbers are stored in ICONSF-array. The total number of elements in the ICONSF-array is NSFC, which represents the total number of solid to fluid conductors in a given model. Subroutine INDEXSFC finds the address location for a given solid to fluid conductor.

SUBROUTINE INDEXSFC (NUMBER, ICONSF, NSFC, ICSF)

Input variables:
NUMBER: Solid to Fluid Conductor number
ICONSF: Array for storing Solid to Fluid Conductor Number
NSFC: Number of Solid to Fluid Conductors

Output variable:
ICSF: Location of Solid to Fluid Conductor in Array (Pointer)

4.3.1.7 Solid to Ambient Conductor. User-defined solid to ambient conductor numbers are stored in an ICONSA-array. The total number of elements in the ICONSA-array is NSAC, which represents the total number of solid to ambient conductors in a given model. Subroutine INDEX-SAC finds the address location for a given solid to ambient conductor.

SUBROUTINE INDEXSAC (NUMBER, ICONSA, NSAC, ICSA)

Input variables:
NUMBER: Solid to Ambient Conductor Number
ICONSA: Array for storing Solid to Ambient Conductor Number
NSAC: Number of Solid to Ambient Conductors

Output variable:
ICSA: Location of Solid to Ambient Conductor in Array (Pointer)

4.3.1.8 Solid to Ambient Conductor. User-defined solid to solid radiation conductor numbers are stored in ICONSSR-array. The total number of elements in the ICONSSR-array is NSSR, which represents the total number of solid to solid conductors in a given model. Subroutine INDEXSAC finds the address location for a given solid to ambient conductor.

SUBROUTINE INDEXSSRC (NUMBER, ICONSSR, NSSR, ICSSR)

Input variables:
NUMBER: Solid to Solid Radiation Conductor Number
ICONSSR: Array for storing Solid to Solid Radiation Conductor Number
NSSR: Number of Solid to Solid Radiation Conductors

Output Variable:
ICSSR: Location of Solid to Solid Radiation Conductor in Array (Pointer)

4.3.2 Description of User Subroutines

Additional capabilities can be added to the code by the utilization of User Subroutines. Twenty-three blank subroutines are provided with a common block of variables. These subroutines are called from various locations in the SP module as shown in figure 45. A few subroutines

also pass pertinent local variables through arguments. Users can develop their code to add new capabilities through use of these variables. A short description of each subroutine is now provided. A description of all common block variables appears in appendix D and listing of blank User Subroutines are provided in appendix E.

4.3.2.1 Subroutine FILNUM. This subroutine is called from the main program at the beginning of computation to assign file numbers to integer names. All file numbers are assigned integer variable names in the MAIN routine of GFSSP. All file numbers are listed in this subroutine for users to make them aware what file numbers are already in use. It also includes 10 additional file numbers for possible use in User Subroutines. Users need to make sure they do not use the existing file numbers.

4.3.2.2 Subroutine USRINT. This subroutine is called from subroutine INIT. This allows the user to assign different initial values and steady state boundary conditions and overwrite the values assigned in subroutine INIT.

4.3.2.3 Subroutine SORCEM. This subroutine is called from Subroutine EQNS. It has two arguments:

IPN – Address location of node
TERMU – Transient term of mass conservation equation.

In this subroutine, users can define any additional mass sources, EMS (IPN), at any internal node. An alternative form of the transient term in the mass conservation equation (eq. (1)) can be used by overwriting the existing TERMU.

4.3.2.4 Subroutine SORCEF. This subroutine is called from subroutine EQNS. This subroutine allows users the ability to redefine each term in the momentum equation and provides an opportunity to add external momentum sources to any branch. It has 13 arguments:

I – Address location of branch
TERM0 – Unsteady term in momentum conservation equation
TERM1 – Longitudinal inertia
TERM2 – Pressure gradient
TERM3 – Gravity force
TERM4 – Friction force
TERM5 – Centrifugal force
TERM6 – External momentum source due to pump
TERM7 – Momentum source due to transverse flow (multidimensional model)
TERM8 – Momentum source due to shear (multidimensional model)
TERM9 – Variable geometry unsteady term
TERM10 – Normal stress
TERM100 – User-supplied momentum source

The first argument is the address location of the branch. The other12 arguments represent the various terms of the momentum equation. The algebraic form of each term is described in

equation (2). However, GFSSP's Newton-Raphson scheme solves the equation in the following form:

$$F\left(x_1, x_2, x_3, \ldots x_{nl}\right) = 0 . \tag{68}$$

The momentum equation in subroutine EQNS, therefore, appears as:

TERM0 + TERM1 – TERM2 – TERM3 + TERM4 – TERM5 – TERM6
+ TERM7 – TERM8 + TERM9 – TERM10 – TERM100 = 0. (69)

4.3.2.5 Subroutine SORCEQ. This subroutine is called from subroutine ENTHALPY (if SECONDL is false) or from subroutine ENTROPY (if SECONDL is true). It has two arguments:

IPN – Address location of node
TERMD – Component of linearized source term appearing in the denominator of the enthalpy or entropy equation.

This subroutine allows the user to introduce a heat source or sink at any internal node. In numerical calculation it is often necessary to linearize the heat source to ensure numerical stability. Suppose one needs to account for heat transfer from the wall at a given temperature (say T_{wall}) to the fluid at T_F in the energy conservation equation. The additional heat source can be expressed as:

$$\dot{Q}_{wall} = hA\left(T_{wall} - T_F\right) . \tag{70}$$

In a linearized formulation of the energy conservation equation, $h_c A T_{wall}$ appears in the numerator and $h_c A / C_p$ appears in the denominator of the equation as shown below:

$$h_i = \frac{\displaystyle\sum_{j=1}^{j=n} a_j h_j + h_c A T_{wall}}{\displaystyle\sum_{j=1}^{j=n} a_j + h_c A / C_p} . \tag{71}$$

Example of coding:

```
SORCEH(IPN) = HC*HAREA*TWALL
TERMD = HC*HAREA/CPNODE(IPN)
```

where

HC = heat transfer coefficient (Btu/ft^2-s-R)
HAREA = heat transfer area (ft^2)
TWALL = wall temperature (°R)
CPNODE(IPN) = specific heat of fluid at IPN (Btu/lb$_m$–R).

An example of adding a heat source is shown in example 8 in section 6.

4.3.2.6 Subroutine SORCEC. This subroutine is called from subroutine MASSC. This subroutine allows users to introduce a source or sink of species at any internal node. An example of uses of this subroutine appears in example 10 of section 5.

4.3.2.7 Subroutine SORCETS. This subroutine is called from subroutine TSOLID or TSOLIDNR depending upon whether the energy conservation equation for solid is solved by successive substitution or the Newton-Raphson method. This subroutine allows users to introduce heat source or sink at any solid node.

4.3.2.8. Subroutine KFUSER. This subroutine is called from subroutine RESIST. In this subroutine, users can introduce a new resistance option in any branch. It has four arguments:

I – Address location of branch
RHOU – Upstream node density
EMUU – Upstream node viscosity
XVU – Upstream node vapor quality
RHOUL – Upstream node liquid density
EMUL – Upstream node liquid viscosity
AKNEW – K_f for the branch in consideration.

Users must provide all input data to calculate K_f for the branch in this subroutine.

4.3.2.9 Subroutine PRPUSER. This subroutine will be used when users want to integrate a separate thermodynamic property package instead of built-in thermodynamic property packages, GASP, WASP, and GASPAK.

4.3.2.10 Subroutine TSTEP. This subroutine is called from the main program at the start of each time step. In this subroutine the user has the opportunity to overwrite and prescribe a new time step.

4.3.2.11 Subroutine BNDUSER. This subroutine is called from subroutine BOUND. In this subroutine users can modify boundary conditions and geometry at each time step for an unsteady model. This subroutine must be used when users want to integrate a separate thermodynamic property package instead of the built-in thermodynamic property packages, GASP and WASP. In an unsteady model, boundary conditions are specified at each time step. The thermodynamic properties at the boundary node must be calculated at the start of a new time step. Example 8 in section 6 demonstrates the use of this subroutine.

In order to modify the geometry, users can make use of six allocated arrays (BRPR1 through BRPR6) that store the branch parameters for all resistance options. Table 9 describes the allocation of branch parameters of all resistance options to these six arrays.

Table 9. Description of branch parameters for all resistance options.

Branch Option	BRPR1	BRPR2	BRPR3	BRPR4	BRPR5	BRPR6
Pipe	Length	Diameter	ε/D			
Restriction	C_L					
Noncircular duct	Length	Height	Width	Type (1–4)		
Pipe with entrance and exit losses	Length	Diameter	ε/D	K_i	K_e	
Thin, sharp orifice	D_1	D_2				
Thick orifice	Length	D_1	D_2			
Square reduction	D_1	D_2				
Square expansion	D_1	D_2				
Rotating annular duct	Length	r_o	r_i	rpm		
Rotating radial duct	Length	Diameter	rpm			
Laby seal	Radius	Clearance, c	Pitch, m	Number of teeth, n	Multiplier, α	
Parallel plates (face seal)	Radius	Clearance, c	Length			
Fittings and valves	Diameter	K_1	K_∞			
Pump characteristics	A_0	B_0	C_0			
Pump Hp	Power, P	Efficiency, η				
Valve with C_v	C_v					
Viscojet	L_Ω	V_f	k_v			
Control valve	C_L	Control Node				
User-defined	User-defined	User-defined	User-defined	User-defined	User-defined	User-defined
Heat exchanger core	Frontal area	Free-flow area	Heat transfer area	Coefficient of control	Coefficient of Exp.	Length
Parallel tubes	Length	Diameter	ε/D	Number of tubes, n		
Compressible orifice	C_L					
Laby seal, Egli correlation	Radius	No. of teeth	Tooth width	Pitch	Clearance	
Fixed flow	Flow rate					

4.3.2.12 Subroutine PRNUSER. This subroutine is called from subroutine PRINT. In this subroutine users can add additional information in GFSSP output files or can create new output files.

4.3.2.13 Subroutine USRSET. This subroutine is called from subroutine READIN if USETUP is set to true. This subroutine allows users to set up their own model instead of using the GFSSP preprocessors. When this option is activated, GFSSP reads the title, input, and output filenames from the data file. The user must provide other necessary information for the model. Only experienced users may have a need to use this subroutine.

4.3.2.14 Subroutine USRHCF. This subroutine is called from subroutine CONVHC. It has two arguments:

NUMBER – Address location of solid to fluid conductor
HCF – Heat transfer coefficient in Btu/s-ft^2-°R.

This subroutine allows users to calculate heat transfer coefficient by a correlation provided by the user to overwrite the heat transfer coefficient calculated by GFSSP's solver module.

4.3.2.15 Subroutine USRADJUST. This subroutine is called from MAIN. Users can adjust the boundary condition and introduce additional iterative cycle to achieve any desired design goal.

4.3.2.16 Subroutine SORCEHXQ. This subroutine is called from subroutine ENTHAL-PYX, which solves the energy equation for individual species (enthalpy-2 option for mixture, sec. 3.1.3.2). It has three arguments:

IPN – Address location of node
TERMD – Component of linearized source term appearing in the denominator of the species enthalpy equation
K – Index of the fluid species in the mixture.

This subroutine allows the user to introduce a heat source or sink at any internal node for a given species. The use of TERMD has been explained in the context of subroutine SORCEQ.

4.3.2.17 Subroutine KFADJUST. This subroutine is called from subroutine RESIST to allow users to modify the K_f value of a particular branch. It has nine arguments:

I – Address location of branch
RHOU – Upstream node density
EMUU – Upstream node viscosity
RHOUL – Upstream node liquid density
EMUL – Upstream node liquid viscosity
RHOUV – Upstream node vapor density
EMUV – Upstream node vapor viscosity
ISATU – Index to designate saturation condition (ISATU = 1) of upstream node
AKNEW – K_f for the branch in consideration.

4.3.2.18 Subroutine PRPADJUST. This subroutine is called from subroutine DENSITY and allows users to adjust any thermodynamic or thermophysical properties, if necessary.

4.3.2.19 Subroutine TADJUST. This subroutine is called from subroutine MIXPROP which calculates mixture properties and temperature for the enthalpy-2 option of modeling fluid mixture. This subroutine allows users to adjust node temperature, if necessary.

4.3.2.20 Subroutine PADJUST. This subroutine is called from subroutine NEWTONSS which controls the Newton-Raphson scheme. This subroutine allows users to adjust node pressure, if necessary.

4.3.2.21 Subroutine FLADJUST. This subroutine is called from subroutine NEWTONSS which controls the Newton-Raphson scheme. This subroutine allows users to adjust mass flow rate in the branch, if necessary.

4.3.2.22 Subroutine HADJUST. This subroutine is called from subroutine ENTHALPY, which solves for enthalpy of single fluid, and subroutine ENTHALPYX, which solves for enthalpies of all species in the mixture (enthalpy-2 option). This subroutine allows users to adjust enthalpies, if necessary.

4.3.2.23 Subroutine USRMDG. This subroutine is called from MAIN. This subroutine allows users to modify geometrical parameters for a multidimensional grid.

4.3.2.24 Utility Subroutines and Functions. User Subroutine code will sometimes use linear interpolation, for example, when calculating boundary conditions as a function of time. The user may choose to use GFSSP's built-in subroutine INTERPOL:

CALL INTERPOL(XVALUE, N, XARRAY, YARRAY, YVALUE)

Input:
> XVALUE – x value at which y will be interpolated
> N – size of arrays XARRAY and YARRAY
> XARRAY(N) – array of increasing x values
> YARRAY(N) – array of y values corresponding to XARRAY

Output:
> YVALUE – y value interpolated at XVALUE

Subroutine INTERPOL is for linear interpolation only. It does not extrapolate. Out-of-range values will be set equal to the first or last value in YARRAY, as appropriate.

User Subroutines may also make use of various unit conversion functions to convert between SI and English units. For example:

REAL FUNCTION KW_BTUS(VALUE)

will convert the power in kW stored in variable VALUE to BTU/s. These conversion functions are described in the User Subroutine write-up of appendix E.

5. GRAPHICAL USER INTERFACE

This section introduces the visual thermofluid dynamics analyzer for systems and components (VTASC), a unique GUI designed to simplify the model building process for GFSSP. VTASC allows the user to design GFSSP models using an interactive 'point and click' paradigm. The program seeks to eliminate some of the more tedious, error prone, and time-consuming operations associated with the model building process such as the selection of unique numbers for nodes and branches, and the explicit specification of the upstream and downstream nodes for every branch. The models may be easily modified both in terms of additional nodes and branches and the model specific data. Figure 46 shows the main VTASC window that consists of menu and toolbar options and a blank canvas.
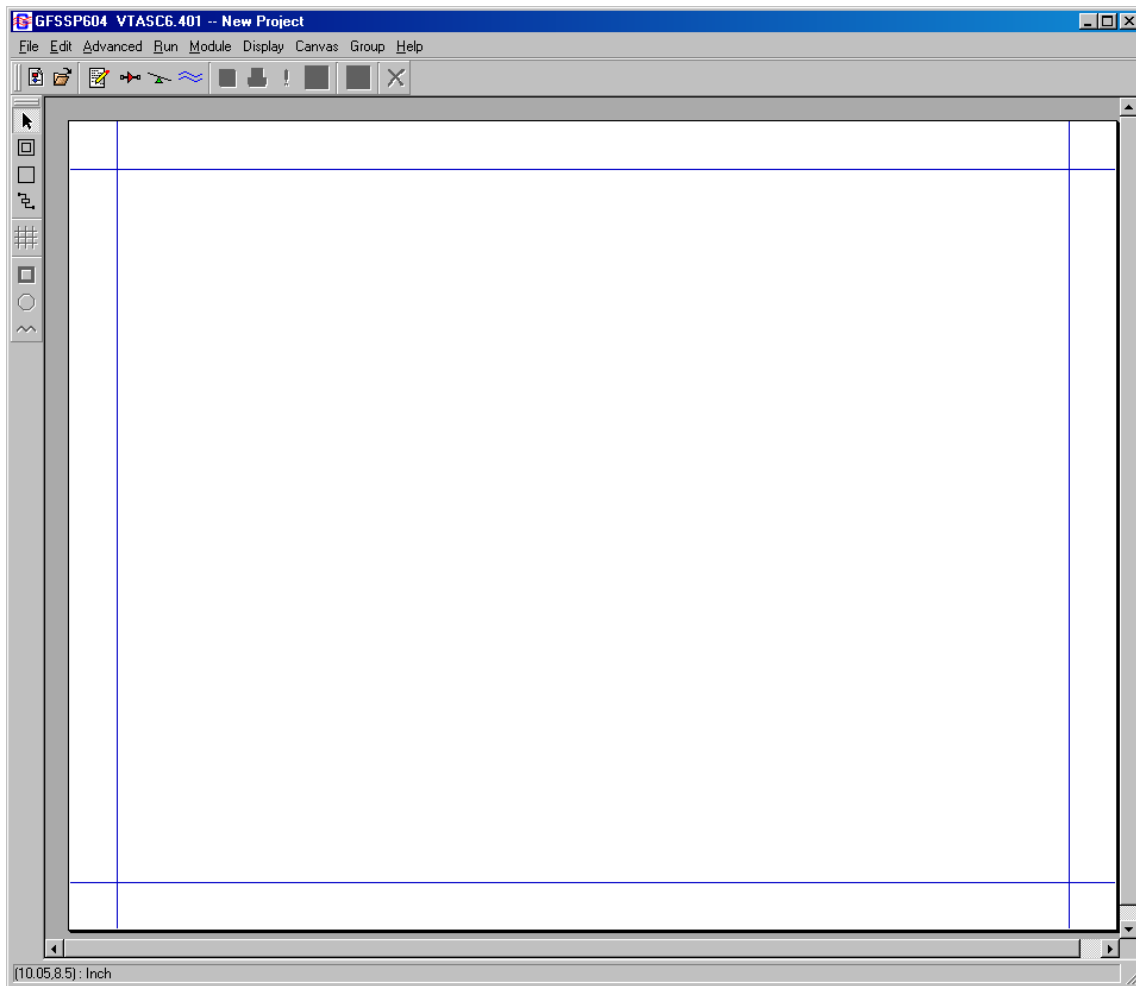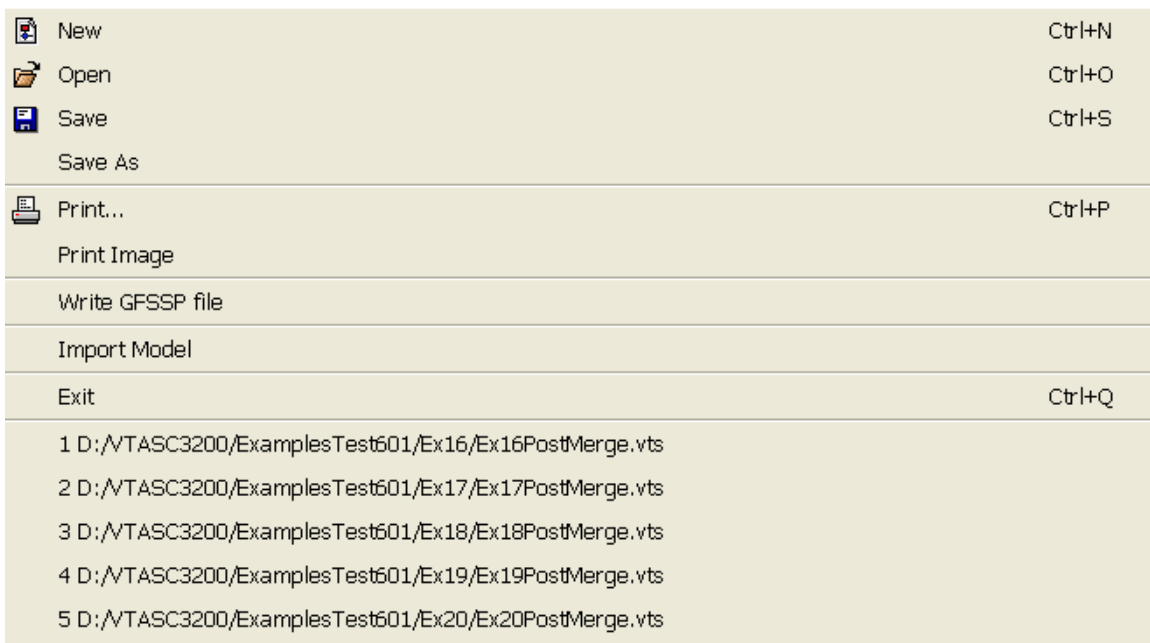
Figure 46. Main VTASC window.

5.1 Menus

5.1.1 File Menu

The File pulldown menu, shown in figure 47, contains the functions to begin a new model, open an existing model, save the model, save the model with an alternate location and name, print the model to a printer, print an image of the model to a bitmap (.bmp extension) file, write an input file for GFSSP based on the current model, import a second model into an existing model, and exit the application. The most commonly used of these functions are available, as shortcuts, from the file input/output toolbar. In addition, the File menu contains a listing of the nine most recently saved models.

📄 New		Ctrl+N
📂 Open		Ctrl+O
💾 Save		Ctrl+S
Save As		
🖨 Print...		Ctrl+P
Print Image		
Write GFSSP file		
Import Model		
Exit		Ctrl+Q
1 D:/VTASC3200/ExamplesTest601/Ex16/Ex16PostMerge.vts		
2 D:/VTASC3200/ExamplesTest601/Ex17/Ex17PostMerge.vts		
3 D:/VTASC3200/ExamplesTest601/Ex18/Ex18PostMerge.vts		
4 D:/VTASC3200/ExamplesTest601/Ex19/Ex19PostMerge.vts		
5 D:/VTASC3200/ExamplesTest601/Ex20/Ex20PostMerge.vts		

Figure 47. VTASC File menu.

The new model function will reinitialize the application to a clean state without having to exit and then restart the application; if model data are present, then the user will be prompted to continue the operation. The open model function will present a file dialog to allow the user to select a previous model; note that all model files have a '.vts' extension. The VTASC model files are not synonymous with the GFSSP input files; GFSSP compatible input files may be generated, based on the current model, as described below. The Save function allows the user to save the current model to a desired location. In the case where the model has not been previously read or saved, a file dialog will appear and allow the user to save the current model to a given location. The Save As function works identically except that a file dialog will appear in all instances. The Print function produces a Postscript file, which allows the user to print the current circuit to a printer or to a file. The Print Image function will save an image of the circuit in a bitmap file. The Write GFSSP

file function will become active once the user has input the required data; this is covered in the following section. The Import Model function allows the user to insert additional models to an existing model. This function will become active once the user has created a branch. The Save and Print functions are not available until at least one node is present.

5.1.2 Edit Menu

The Edit pulldown menu, shown in figure 48, contains the functions to delete a selected item(s), activate the Global Options dialog, open an existing GFSSP output or input file using the desired editor (see Global Options User Information tab, fig. 51c), and to select all elements on the canvas. The Delete function, which is not available until at least one node is present, also appears as a shortcut from the file input/output toolbar. The Global Options dialog is discussed in detail in section 5.2.
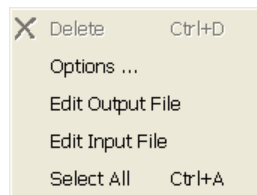
Figure 48. VTASC Edit menu.

5.1.3 Advanced Menu

The Advanced pulldown menu, shown in figure 49, contains the functions to activate dialogs for GFSSP's advanced options such as Transient Heat, Heat Exchanger, Tank Pressurization, Turbopump, Valve Open/Close, Fluid Conduction (not active), Pressure Regulator, Flow Regulator, Pressure Relief Valve, Enable/Disable Grid Generation, Enable/Disable Conjugate Heat Transfer, and Select/Deselect SI Units. With the exception of Enable/Disable Grid Generation, Enable/Disable Conjugate Heat Transfer, and Select/Deselect SI Units, these functions are not available unless the advanced option has been activated through the Global Options dialog. The advanced option dialogs are discussed in detail in section 5.4. Selecting Enable Grid Generation enables the grid node on the left side of the main window. Selecting Enable Conjugate Heat Transfer will enable the three Conjugate Heat Transfer related icons on the left side of the main window. Selecting Enable SI Units will change all of the VTASC data input field labels to SI units.
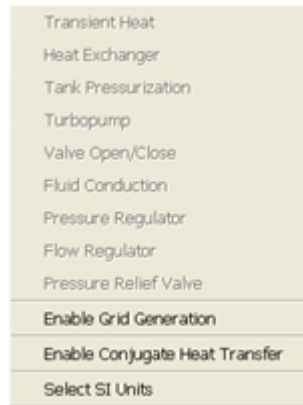
Figure 49. VTASC Advanced menu.

5.1.4 Run and Module Menus

The Run pulldown menu contains the functions to call and run GFSSP, Winplot, and GFSSP and Winplot together. These functions also appear as shortcuts from the file input/output toolbar. The third function will call Winplot and then start GFSSP. Note that Winplot is not part of the GFSSP installation package and must be obtained separately. Also note that Winplot is not made available by VTASC unless a model is defined as an unsteady model. The Module menu contains the function to activate the User Executable Build dialog. The User Executable Build dialog is discussed in detail in section 5.6.

5.1.5 Display, Canvas, Group, and Help Menus

The Display pulldown menu contains the functions to activate the Display Results/Properties dialog, clear any results/properties displayed on the canvas, and enable Quicklook. The Display Results/Properties dialog is discussed in detail in section 5.8.2. The Canvas pulldown menu contains the functions to toggle between a 1-, 2-, or 4-page canvas. The double page canvas configuration is useful for larger models that will not easily fit on a single page canvas. Note that in multiple page configurations, the user should allow for margins for printing. The Group pulldown menu provides the ability to perform move/align operations on a group of network objects that have been selected. The Help pulldown menu contains the function to activate a popup window with additional information about that particular version of VTASC, open the GFSSP User Manual, and disable/enable tooltips.

5.2 Global Options

From the Edit menu, select the menu option labeled Options… to display the Global Options dialog shown in figure 50. A left mouse click on items listed to the extreme left allows access to the desired information within the right pane. As shown, selecting the 'Instructions' option gives general instructions on the use of this dialog.
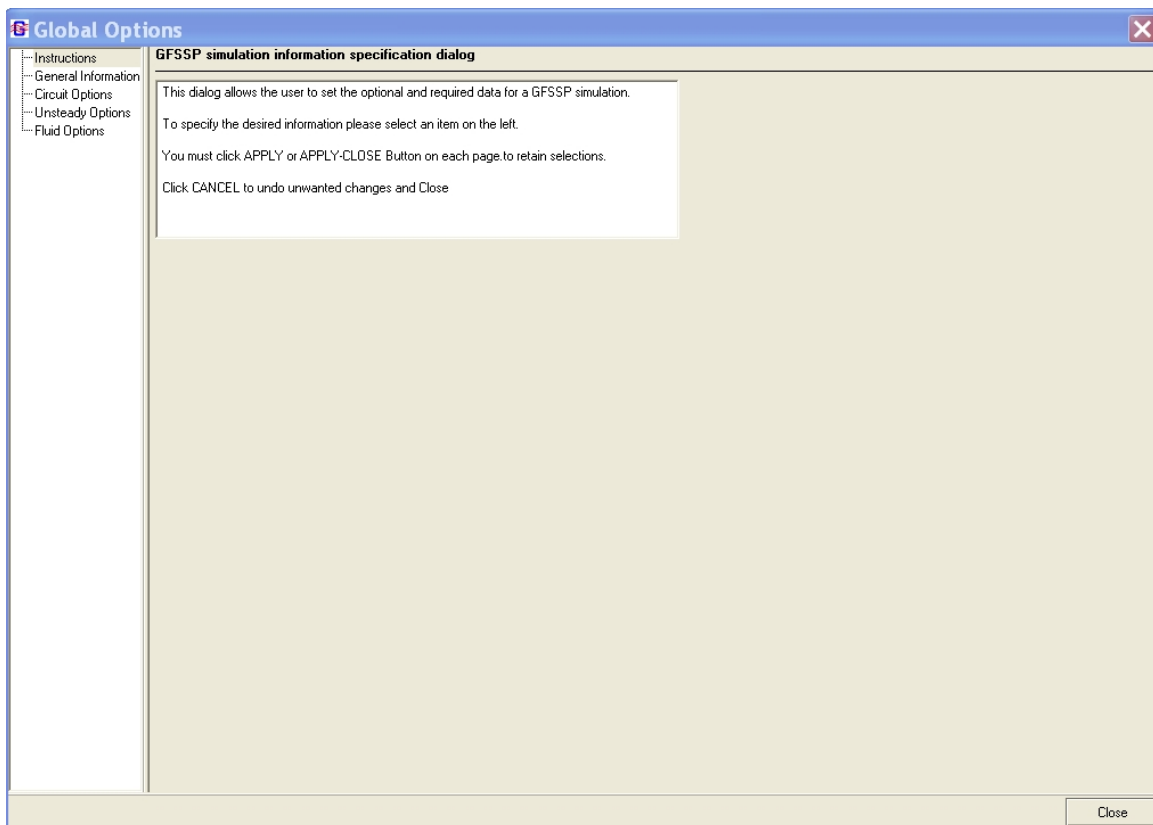
Figure 50. Global Options dialog.

5.2.1 General Information

Selecting the General Information item, or toolbar alternative ⊞ , displays the following dialog (fig. 51) to access User Information, Solution Control, and Output Control. The User Information tab shown in figure 51(a) allows specification of the title, name of the analyst, working directory, GFSSP compatible input file, the output file to be generated by GFSSP, and the name and location of the GFSSP executable that will be used to run the model. The working directory, which is the directory where VTASC will write the GFSSP input and output files associated with the model, is assigned by specifying a file path for the GFSSP input file. The installation version of GFSSP is the default executable defined for any new model. Note that the file menu option Write GFSSP File will become active only when the input and output GFSSP files have been specified. The Compiler radio button allows the user to select either Compaq or Intel Fortran to compile user subroutines. The default text editor for input, output, and history files is Notepad. The user may switch to another editor (e.g. Textpad) by browsing to the executable in the Editor box.

(a)

Global Options

Setup for specifying general information

(b)

User Information | Solution Control | Output Control

Solver Information
☑ Simultaneous Solution

Solution Methods

Energy for Fluid
⦿ Energy by First Law
○ Energy by Second Law

Energy for Solid
⦿ Newton Raphson
○ Successive Substitution

Differencing Scheme
⦿ First Order
○ Second Order

Nonlinear Solver
⦿ Newton - SS
○ Broyden - SS

Convergence Information
Convergence Criteria 0.0001   Relax K 1   Relax H 1   Relax HC 1
Maximum Iterations 500   Relax D 0.5   Relax NR 1   Relax TS 1

Initial Guess
○ User Specified
○ Successive Substitution
⦿ Internally Specified

☐ Save Information   Node Restart Save/Read File : FNODE.DAT
☐ Read Information   Branch Restart Save/Read File : FBRANCH.DAT

Default   Cancel   Apply-Close   Apply

Close

Global Options

Setup for specifying general information

(c)

User Information | Solution Control | Output Control

Output Information
☑ Network Information
☑ Extended Print Information   ☐ Extended Plot Information
☐ Initial Field
☐ Check Values   ☐ Debug Solver
☐ Winplot Data   Plot frequency: 1   ○ CSV file ⦿ Binary file   Binary file write frequency 50   Buffer 1.1
☐ Plot User Specified Variables   Number User Variables (max 20) 1
☐ Tecplot Data   Plot frequency: 1
☐ Disable GFSSP Run Information.
☑ Disable PLT file creation.

Default   Cancel   Apply-Close   Apply

Close

Figure 51. General information dialogs: (a) User Information tab, (b) Solution Control tab, and (c) Output Control tab.

84

The Solution Control tab shown in figure 51(b) allows specification of certain characteristics of the solution procedure of a particular model. The user can choose either a Simultaneous Solution procedure or the original Hybrid Solution scheme. The user can also choose between the First Law and Second Law of thermodynamics based solution procedures for the energy equation. For the solid energy equation, the user can select either a Newton Raphson or Successive Substitution solution scheme. For the differencing scheme, the user may select either the First Order or Second Order. For the Nonlinear Solver, the user may select either the Newton-Successive Substitution or Broyden-Successive Substitution method. In addition, the user can specify the Convergence Criteria, Maximum Iterations, Relaxation parameters, and choose the method by which the initial guess is made. The user can also specify if they wish to use restart files by checking the appropriate box. Checking the Save Information box indicates the user wishes to save the final solution to use as an initial guess in another model. Checking the Read Information box indicates the user wishes to read in a previously saved solution as an initial guess. Note that if both boxes are checked, GFSSP will overwrite the initial restart files during the simulation. Two restart files are used by GFSSP for both saving and reading information. One file is used for node information and one for branch information. The user may name the restart files using the designated text boxes.

The Output Control tab shown in figure 51(c) allows specification of the type of data to output during the GFSSP simulation as well as requesting certain values to be checked for reasonableness, and requesting various messages for model debugging purposes. The solution check can be activated by choosing Check Values. The output options consist of inclusion (default) or suppression of: (1) Network Information (print), (2) Extended Thermodynamic and Thermophysical Information at the nodes (print/plot), (3) Initial Flow Field (print), and (4) Winplot data (plot). For Winplot data, the user can define whether and how often GFSSP writes output to the Winplot data files (the default is 1, which writes output at every time step) as well as the type of file that is generated (comma separated value files or a binary file formatted for the Winplot plot program). If Binary file is selected, the user may select the number of plot records to buffer for each write to the binary file. The Buffer entry (default 1.1) is a way to give allowance for possible increase in the data (decrease in time step) to be written. Checking CSV will cause GFSSP to write a comma-separated variables file, which can be processed by Excel.

GFSSP provides a way for a user to define variables to be plotted. User-defined variables are defined in one of the subroutines in the User Subroutine file, found in the installation folder. Checking Tecplot Data will cause GFSSP to write data to a file that is suitable for input into the Tecplot program.

Checking Disable GFSSP Run Information will stop GFSSP from writing back to VTASC in the GFSSP Run Dialog. This has been found to be helpful for long-running models. GFSSP will create <modelname>.PLT files, which provide output data that can be used to present results within VTASC. Disabling (default) will prevent these files from being created.

Note that the Apply or Apply-Close button must be pressed to accept modifications to the data in any of the tabs. To reset the data to default on all three tabs press Default and then the Accept buttons. The Cancel button closes the General Information item without accepting any changes.

5.2.2 Circuit Options

Selecting the Circuit Options item, or toolbar alternative ▪▪▫, displays the pane in figure 52, allowing access to Circuit Options and Initial Values. The Circuit tab shown in figure 52(a) allows specification of the options that will be activated for this circuit. These options include: Axial Thrust; Cyclic Boundary; Dalton's Law (only active for multiple fluid models); Enthalpy Formulation (Stagnation or Static); Fluid Conduction (activates Fluid Conduction on Advanced Menu, but is not active); Fluid Mass Injection; Gravity (enables Buoyancy & Reference Node); Heat Exchanger (activates Heat Exchanger Advanced Menu); Heat Source (with optional units); Inertia (Note that while this option allows you to choose to supply relative angles between adjacent branches, no mechanism currently exists to define those angles using VTASC. The user must manually edit the GFSSP input file to supply those angles (also discussed in sec. 5.5.13); Mixture (with option to select method of calculations); MD-Grid; Momentum Source; Moving Boundary; Normal Stress; Phase Separation Model; Rotation; Shear; Solid-Fluid Heat Transfer Coefficient with the ability to choose method of calculation (only active for models using conjugate heat transfer); Transient Term Active; Transverse Momentum; and Turbopump (activates Turbopump Advanced Menu). Note that activating these options may require additional inputs in other areas of VTASC.

The Initial Values tab shown in figure 52(b) allows the user to set initial values of Pressure, Temperature, etc., for both boundary and interior nodes as well as solid nodes. The initial value for nodes may be changed, after the model has been built, by modifying the desired data and pressing the Apply to All button or the Apply to Selected button. For multiple fluids the option to set default concentrations will become visible. Node ID Sequence Start gives the user the ability to specify the starting Node ID. This may be changed at any time during model creation.

(a)
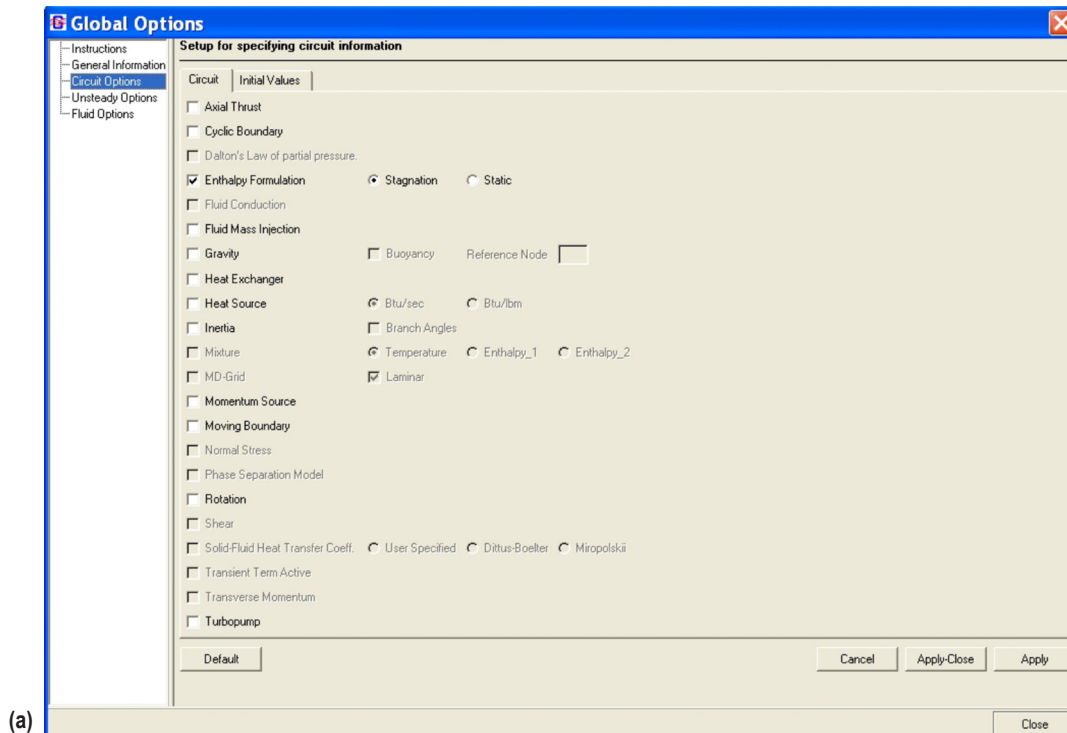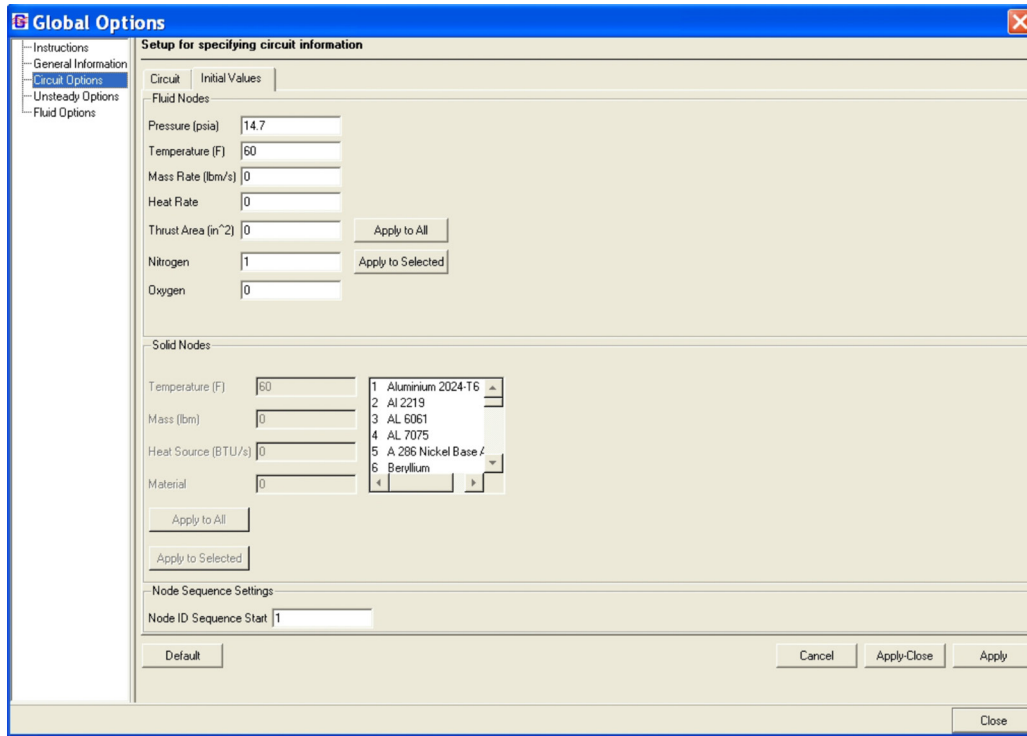
(b)

Figure 52. Circuit Options dialogs: (a) Circuit tab and (b) Initial Values tab.

5.2.3 Unsteady Options

Selecting the Unsteady Options item, or toolbar alternative ⤬, displays the dialog shown in figure 53. The Unsteady Options dialog allows users to choose from various levels of unsteady modeling. The options in this window include: Steady (default); Quasi-Steady; Time Step; Start Time (a relative time—does not have to be zero); End Time; Print Frequency (controls print interval to all output files except Winplot files); Unsteady; Variable Rotation (user specifies variable rotation file name); Variable Heat Load (activates Transient Heat Advanced Menu); Variable Geometry (user specifies the variable geometry file name); Tank Pressurization (activates Tank Pressurization Advanced Menu); Valve Open/Close (activates Valve Open/Close Advanced Menu); Pressure Regulator (activates Pressure Regulator on Advanced Menu); Flow Regulator (activates Flow Regulator on Advanced Menu); and Pressure Relief Valve (activates Pressure Relief Valve on Advanced Menu). Note that Pressure Regulator Option 1 and Flow Regulator Option 1 are mutually exclusive in a circuit. Note that activating the unsteady options may require additional inputs in other areas of VTASC.
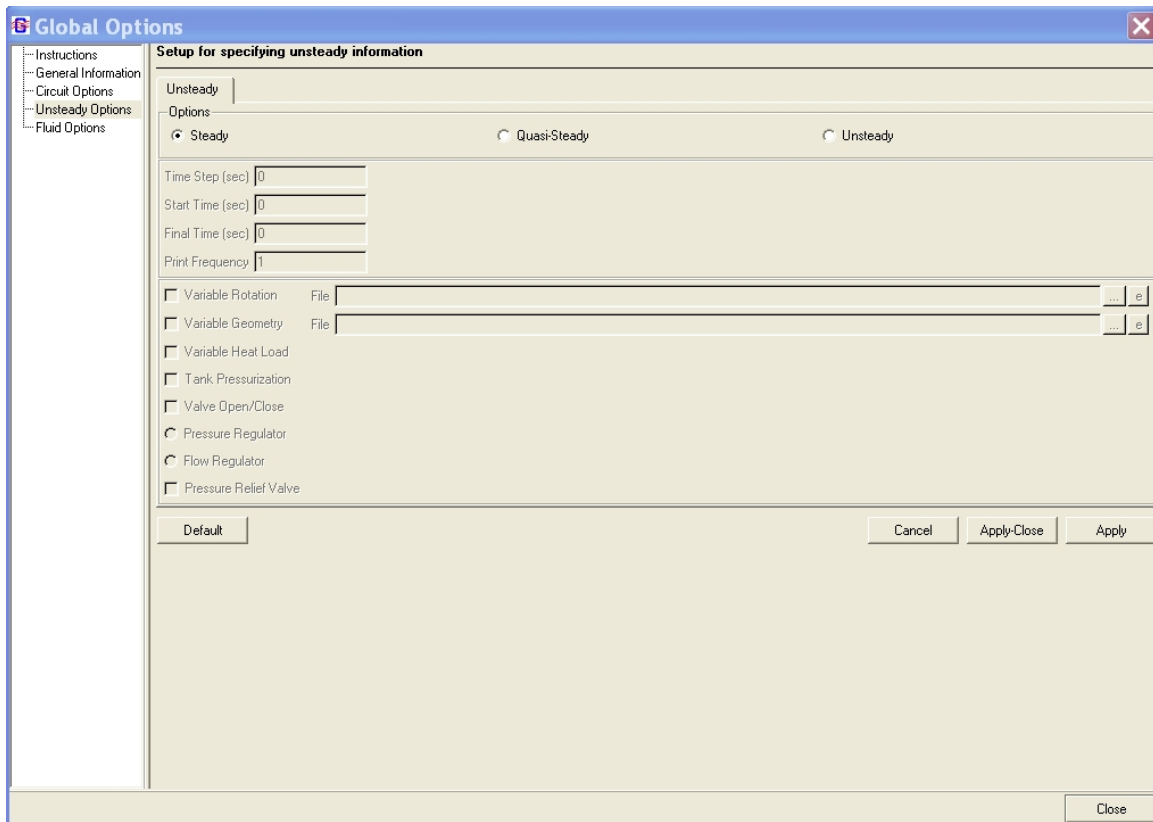
Figure 53. Unsteady Options dialog.

5.2.4 Fluid Options

Finally, selecting the Fluid Options item, or toolbar alternative ≈, displays the dialog shown in figure 54. The Fluid Options dialog allows users to choose the thermodynamic property approach used in the model. The user can choose from the embedded thermodynamic property packages (1) GASP and WASP or (2) GASPAK. Additionally, the user can choose a constant density fluid (the energy equation is not calculated with this option and this option cannot be used with fully unsteady modeling); an ideal gas by specifying the fluid gas constant, specific heat, viscosity, thermal conductivity, reference pressure, reference temperature, reference enthalpy, and reference entropy; hydrogen peroxide, with the capability to define the mole fraction of water present in the fluid; and user-defined fluids through user-defined property files. Note that choosing the constant density fluid option causes the program to set the flow to steady and all unsteady options to be turned off.
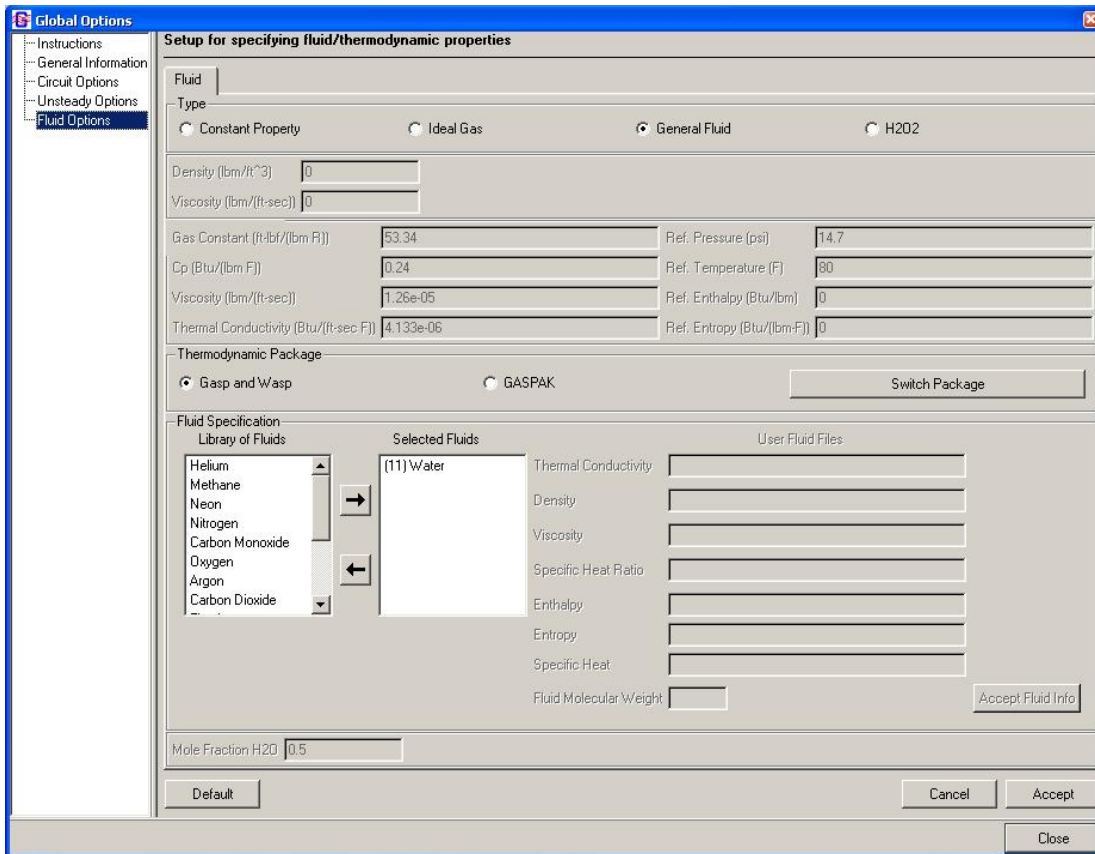
Figure 54. Fluid Options dialog.

Desired fluids from the Library of Fluids may be selected and added to the Selected Fluids list by pressing the ➡ button. Fluids may be deleted from the Selected Fluids list by selecting the unwanted fluids and pressing the ⬅ button. Note that a number enclosed in parentheses appears by each selected fluid. This is the GFSSP index number for that fluid. If a user fluid is selected, VTASC will prompt the user to double-click the fluid name to supply the fluid property file names and fluid molecular weight. Note that all user-defined property files must reside in the model's working directory. The Switch Package button allows the user to switch between the two available thermodynamic property packages. The Switch Package button will only work when all selected fluids are common to both fluid libraries. Note that manually switching between the two thermodynamic property packages will delete all fluids from the selected fluids list and node properties.

5.3 Fluid Circuit Design

5.3.1 Boundary and Internal Node Properties

The boundary node addition tool 🔲 located on the left border of VTASC is used to place boundary nodes on the drawing area, henceforth called the canvas. Upon selection of this tool the user may add boundary nodes by moving the mouse to the canvas and pressing the left mouse button. Placing a node will activate the delete function in the toolbar, edit menu, or by typing

CTRL+D. Similarly, interior nodes may be added by selecting the interior node addition tool ☐ .
Note that the nodes are automatically given unique numeric identifiers. Figure 55 shows a canvas
with a number of boundary and interior nodes.
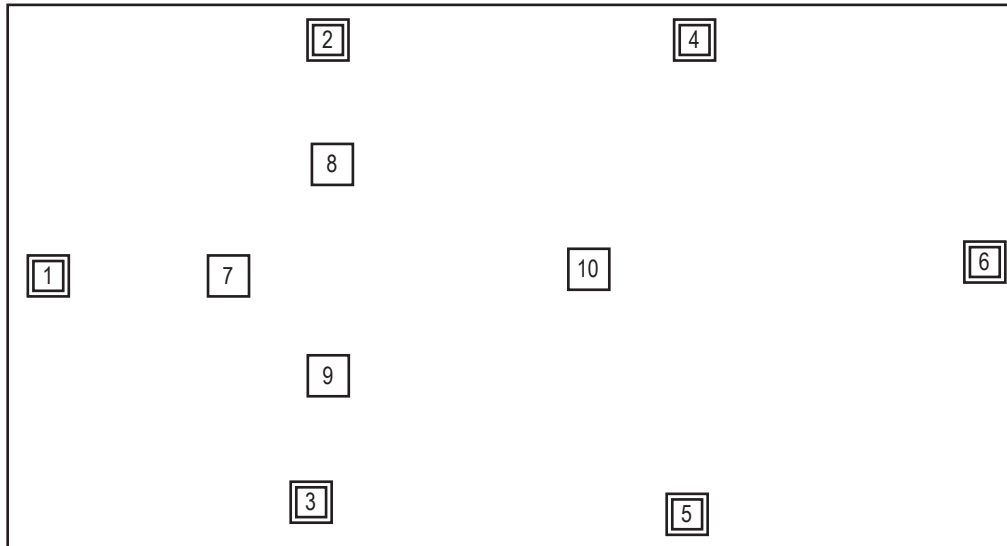
Figure 55. Boundary and interior nodes on canvas.

The selection tool ▶ is used to select the desired node and either modify its location or
enable its deletion. Positioning the mouse and pressing the left button performs the selection;
upon selection the selected node will be shown with a red border. Repositioning a node is simply
performed by pressing and holding the left mouse button over a node, moving the mouse to the
desired location, and releasing the left mouse button. Multiple nodes may be selected for deletion
by using the Ctrl keyboard button in conjunction with the mouse. The nodes may then be deleted
using the Delete toolbar button as long as they are not attached to any branches. A left mouse click
within the canvas, away from any nodes, will deselect any previously selected nodes.

A right mouse click upon a node will select the node and present a popup menu (fig. 56)
allowing the user to delete the node (will not work on multiple node selections and is not available
if the node is connected to any branch), set the properties for the indicated node, save the node
properties to a node property buffer, get the properties from the node property, or align the node
either horizontally or vertically with its neighboring elements. (Horizontal alignment aligns all
elements to the right of the selected node, while vertical alignment aligns all elements below the
selected node.) Once a model has been run, the internal node popup menu also allows the user
to activate the results dialogs, which will be discussed in section 5.8.
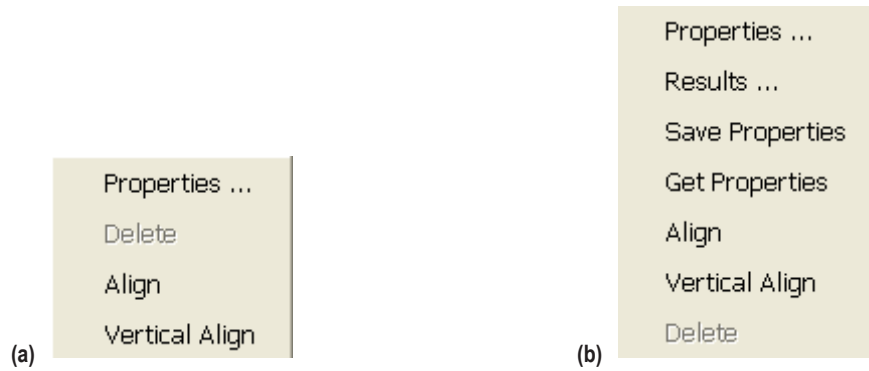
Figure 56. Node popup menus: (a) Boundary node and (b) internal node.

Choosing the Properties … option will present the dialog shown in figure 57. The appropriate inputs will be activated dependent upon the choices present within the Global Options dialog and the type of the selected node, whether boundary or interior. The user may modify the desired data within this dialog. To modify the concentration of a given fluid, select the desired fluid and type in the desired concentration. Note that directly upon selecting a fluid the user may type without having to reposition the mouse. If the user wants to change the numeric identifier for a node, simply type in the desired numeric identifier (maximum of five numbers). The user may enter any desired descriptive text into the Node Description input box. Pressing the OK button will accept and adjust the revised data and the Cancel button will reject the revised data. Also, in the case of an unsteady flow, each boundary node will be automatically assigned a unique Node History File name that is subject to user modification. Checkboxes allow for selection of the node as Moving Boundary, Phase Separation Model, or Cyclic Boundary. These are enabled on Circuit Options.
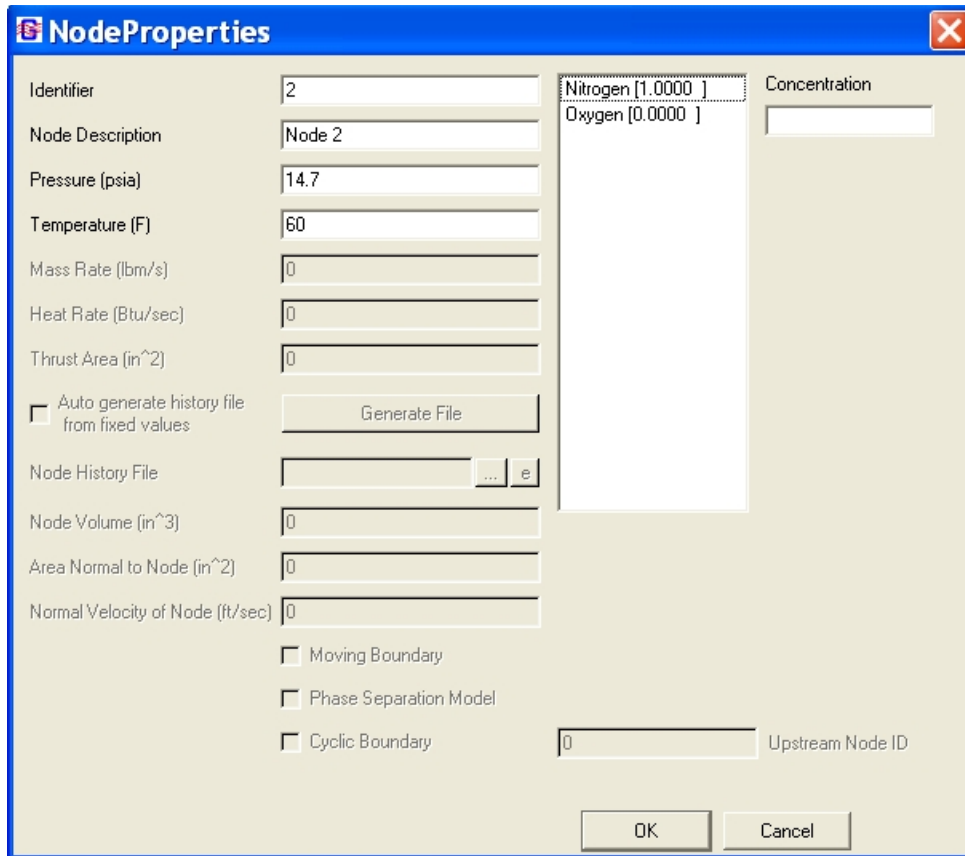
Figure 57. Node properties dialog.

5.3.2 Branch Properties

The branch addition tool ⬚ is used to specify the branches between the nodes. Selection of this tool immediately causes each interior and boundary node to be drawn with a series of 'handles' as denoted by the green squares in figure 58.
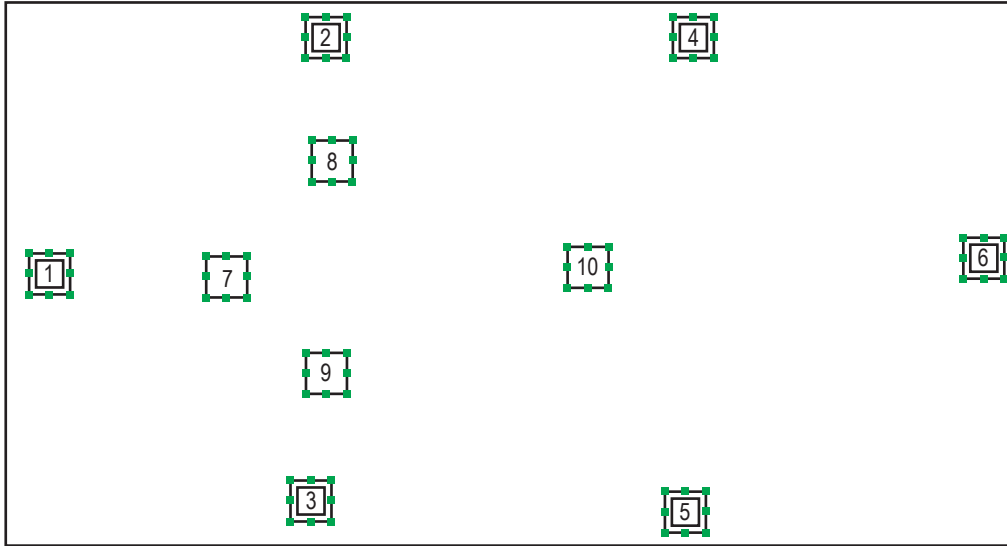
Figure 58. Nodes with branch 'handles.'

The handles serve to clearly identify eight possible locations on a node where the initial (upstream) or terminal (downstream) points of a branch may be located. Note that an unlimited number of branches may initiate or terminate at each handle. There are two different types of branches that may be created—a directed line segment between any two nodes and two possibly discontinuous line segments. In either case, a left mouse button click on a handle will initiate a branch. Once an initial handle (specifies the upstream node) has been selected, further movement of the mouse will draw a directed line segment from that handle to the current location of the mouse, as shown in figure 59(a). For the first type of branch, selecting another handle completes the branch, as this second handle effectively specifies the downstream node (fig. 59(b)).

Figure 59. Direct line segment branch: (a) Upstream node
and (b) downstream node.

The second type of branch is initiated identically; however, after selecting an initial handle, an additional anchor point may be set at any location on the canvas by a left mouse click at the desired location. The branch is then completed, as usual, by selecting another handle. This series of steps is shown in figures 60(a) and (b).
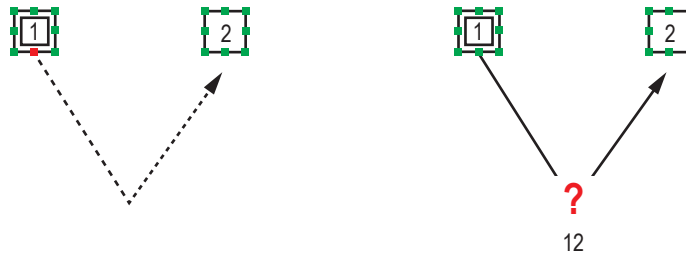
Figure 60. Two-line segment branch.

Completing a branch will activate the delete function in the toolbar, edit menu, or by typing Ctrl+D. As shown in figures 59(b) and 60(b), a unique numeric branch identifier formed by concatenating adjacent node numbers is automatically generated and the directed arrowhead visually defines the branch upstream and downstream node relationship. A click on the branch addition tool ⌁ may be used to clear a branch currently under design and reinitialize the process. The selection tool ▶ may be used to select the desired branch and either modify its location or enable its deletion. The Ctrl button may be used in conjunction with the mouse to select multiple branches and nodes for deletion using the Delete toolbar button. Note that a node can only be deleted using this method if all branches attached to it are deleted as well.

Figure 61 shows an example fluid circuit complete with nodes and branches.
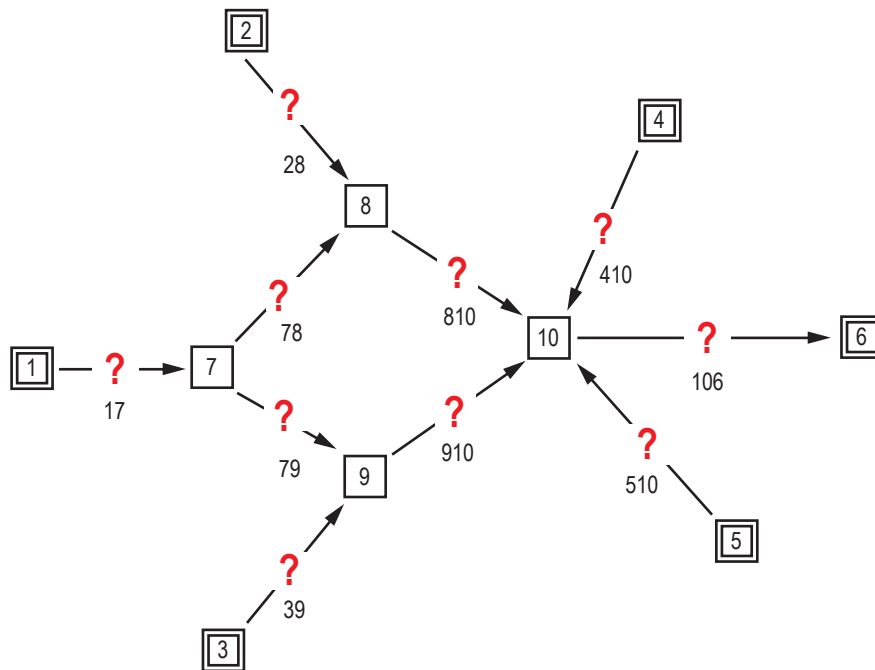
Figure 61. Example fluid circuit with complete branch connections.

94

Specification of a new branch shows an **?** image, which visually indicates that the resistance for this branch has not been specified. A right mouse click upon the image will present the popup menu shown in figure 62, which allows the user to specify the Components, i.e., the resistance for the branch; Align the branch either horizontally or vertically with its neighboring elements (horizontal alignment aligns all elements to the right of the selected branch, while Vertical Alignment aligns all elements below the selected branch); activate the Relocate ID dialog; activate the Change Branch connection dialog; or Delete the branch. The Properties option will be activated once a resistance has been selected for the branch and the Rotation/Momentum Data option will be activated in the case where either Rotation or Momentum is selected in the Global Options and for the specific branch under consideration. Once a model has been run, the Branch popup menu also allows the user to activate the results dialogs, which will be discussed in section 5.8.
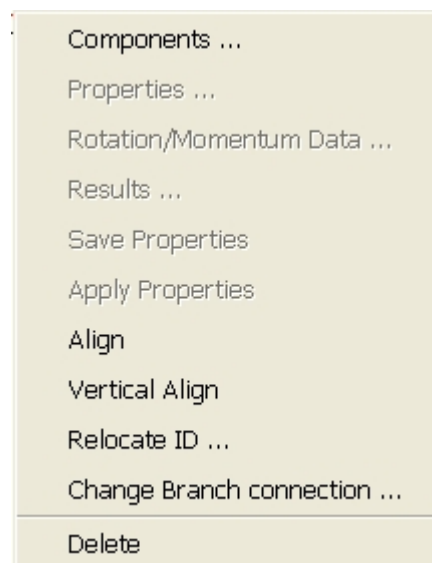
Figure 62. Branch popup menu.

Figure 63 shows the Relocate Branch ID dialog. This dialog gives the user a choice of eight locations where the branch identifier can be placed in relation to the branch element on the canvas. Figure 64 shows the Branch Connections dialog. This dialog allows the user to change the nodes that the branch connects and/or the handles where the branch attaches to each node. Note that changing the nodes connected to a branch does not automatically change the branch identifier.
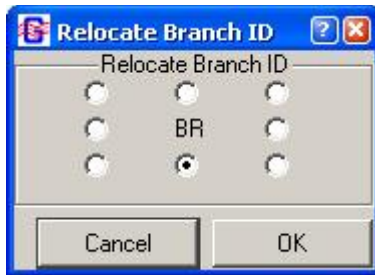
Figure 63. Relocate Branch ID dialog.
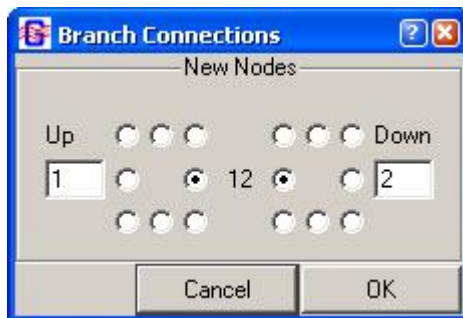
Figure 64. Change Branch Connections dialog.

Choosing the Component … option will present the Resistance Options dialog shown in figure 65. This dialog shows pictorial representations for each of the 27 (unsteady) branch resistance options currently allowed in GFSSP. Note that the Control Valve option will not be available for steady flows. The available resistance options are discussed in section 3.1.7. To assign a resistance option, left click on the desired component and click the Accept button. Figure 66 shows an example where each of the branches has been assigned a resistance option. The user may change the resistance option for a branch at any time without deleting the branch.
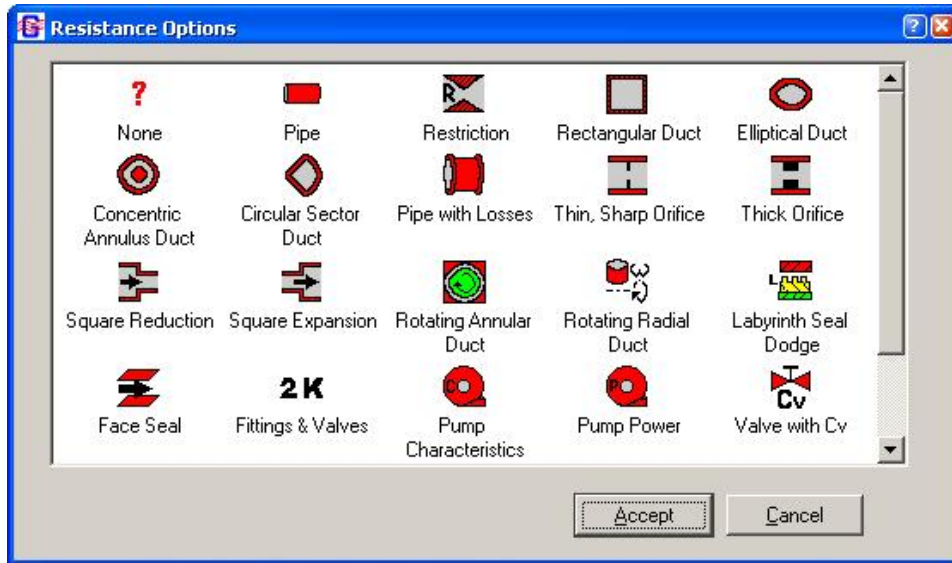
Figure 65. Branch Resistance Options dialog.
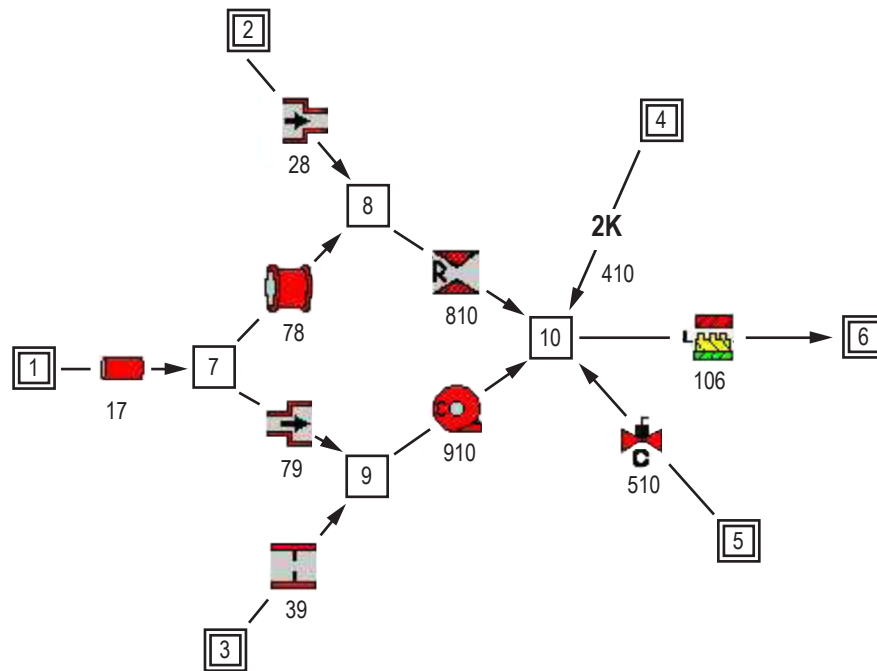
Figure 66. Example fluid circuit with Resistance options.

A right mouse click on a branch where the resistance has been specified will present the popup menu (fig. 62) with the Properties … option activated. Choosing the Properties … option will present a dialog that is specifically tailored to receive input for that Resistance option. In all instances, the properties specification dialogs behave in an identical fashion; however, the Fittings

and Valves - **2K** dialog is somewhat different and will be shown as an example. Choosing the Properties option for a fitting and valve will present the dialog shown in figure 67. The user can input the desired data or use the tree structure to the right to select a desired fitting or valve. Selection of a fitting or valve from the tree will load its specific data into the fields to the left; these data may then be edited as desired. The Accept button must be pressed to apply the data.
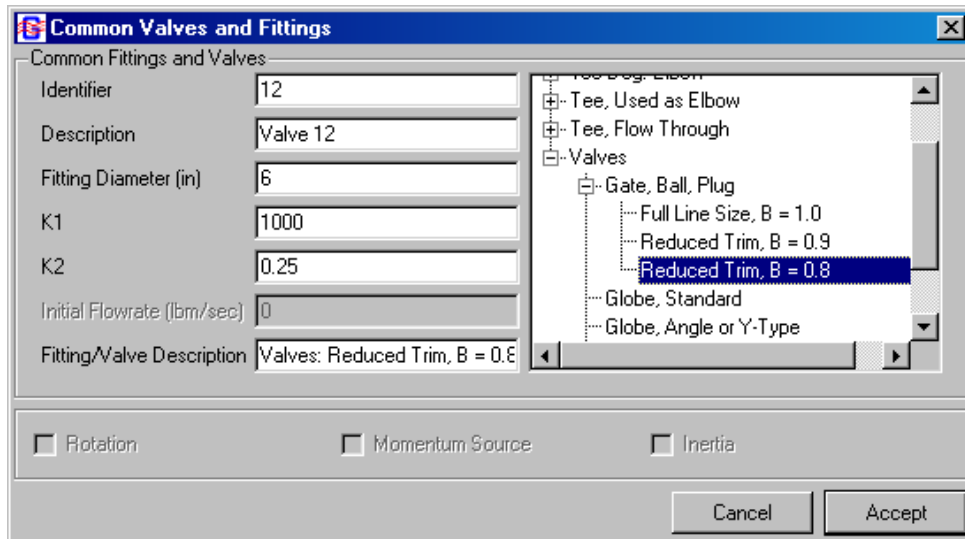
Figure 67. Fittings and valve resistance option properties dialog.

In general, the following applies to *every* resistance option. If the user wants to change the numeric identifier for a branch, simply type in the desired numeric identifier (maximum of five digits). For the unsteady case, the initial flow rate may be specified, and depending upon the selected global options, the Rotation, Momentum Source, and Inertia checkboxes may be active. Notice, in this case, that the Rotation checkbox is active and has been selected. If Rotation has been checked and Accepted, a right mouse click on the branch will present the popup menu with the Rotation/Momentum Source option active, and selecting this option will present the following dialog (fig. 68) to allow input of the relevant information.
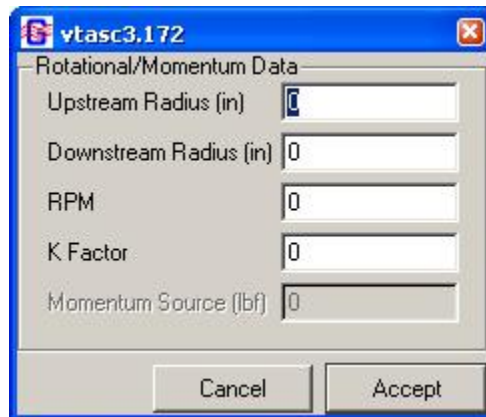
Figure 68. Rotation/Momentum dialog.

98

5.3.3 Conjugate Heat Transfer

The solid node ▢ , ambient node ○ , and conductor ∿ addition tools are used to specify the conjugate heat transfer portion of a GFSSP model. These tools are inactive until the user activates Conjugate Heat Transfer by selecting it from the Advanced menu. In practice, adding solid and ambient nodes to the VTASC canvas is analogous to adding internal and boundary fluid nodes, while adding conductors is an identical process to adding fluid resistance branches.

A right mouse click on an ambient node reveals a popup menu identical to the fluid boundary node popup menu shown in figure 56(a), while the Solid Node popup menu is similar to the Fluid Internal Node popup menu shown in figure 56(b). The Properties dialog for a solid node is shown in figure 69. For a solid node, all inputs are required at each node. A list of available materials is shown at the right of the dialog. When the user left mouse clicks the desired material from the list, the GFSSP index number for that material is automatically written to the Material input box. The Ambient Node Properties dialog shown in figure 70 requires a temperature and, optionally, a history file (unsteady) as a modeling input.
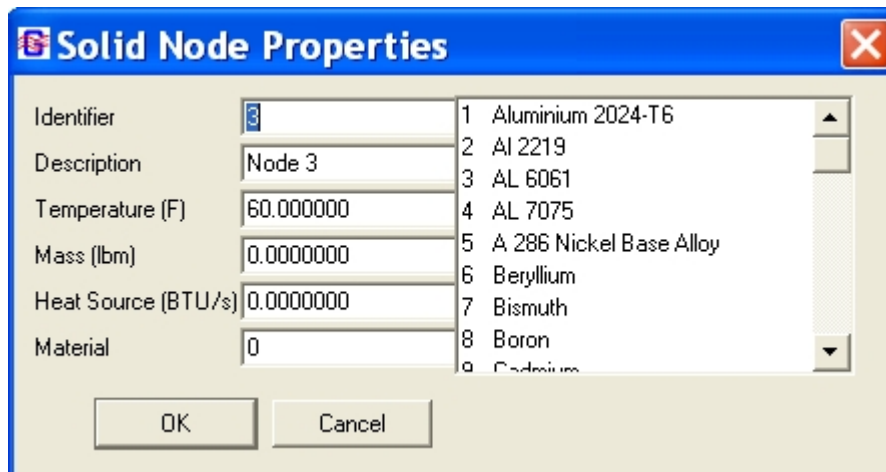
Figure 69. Solid Node Properties dialog.
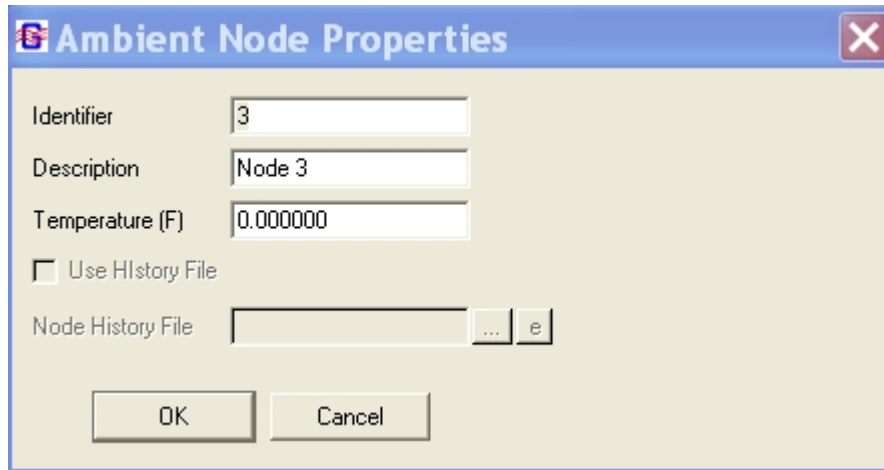
Figure 70. Ambient Node Properties dialog.

The Conductors popup menu (fig. 71) is very similar to the Fluid Branch popup menu (fig. 62). The only differences are that a Conductors option replaces the Components option; there is no need for a Rotation/Momentum option, and the Change Branch Connection option is here named the Change Conductor connection option. (The functionality is identical to the Change Branch Connection option).

Figure 71. Conductors popup menu.

When the user first adds a conductor to the canvas, VTASC indicates that the conductor type is undefined using this symbol ∿⚡. The user defines the type of conductor by selecting the Conductors option from the conductor popup menu. This opens the Conductors dialog shown in figure 72. The user must select the appropriate type of conductor and click the Accept button. Note that VTASC will not allow the user to select a conductor type that is inconsistent with the types of nodes attached to that particular conductor (e.g., a solid-ambient convection conductor type cannot be applied to a conductor connecting two solid nodes). As with fluid resistance branches, once the conductor type has been defined, the Properties option becomes active in the conductor popup
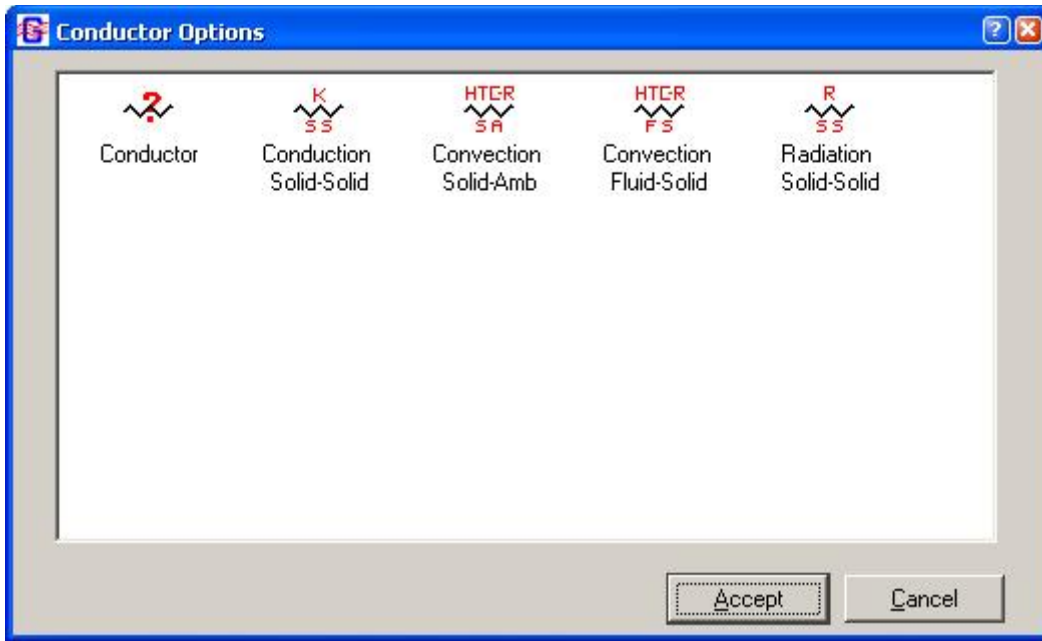
Figure 72. Conductors dialog.

menu. Selecting the Properties option activates a Properties dialog where the user supplies characteristics that are specific to that conductor type.

5.3.4 Import Model

VTASC provides the ability to easily combine two or more models using the Import Model function on the File menu (see fig. 47). Any two models (vts) can be combined in either order, with few exceptions/limitations. The imported model node and branch locations will be maintained. User information, solution control, and output control information will not be imported; only models with General Fluid specified will be imported; fluid conduction (inactive), fluid mass injection, momentum source, and normal stress will not be imported. Component ID conflicts will trigger a dialog giving the user an opportunity to resolve the conflict. Any inconsistencies will generate informational messages. Example 23 utilizes the file import feature.

5.4 Advanced Options

GFSSP contains many advanced features: Transient Heat Load, Heat Exchanger, Tank Pressurization, Turbopump, Valve Open/Close, Fluid Conduction (not active), Pressure Regulator, Flow Regulator, and Pressure Relief Valve. If any or all of the advanced features are selected (via Circuit Options or Unsteady Options), the user can input the appropriate information by selecting the option corresponding to the feature from the Advanced menu (see fig. 49).

The dialogs for each of the advanced options operate in an identical fashion. The user may add any number of components to that option by pressing the Add button. To modify the data for

a particular component, the user must select the component of interest, modify the data, and press the Accept button. To delete a component, press the Delete button after a component has been selected.

5.4.1 Transient Heat

The Transient Heat Load option dialog is shown in figure 73. This option is activated from the Circuit Options pane on the Global Options menu (see sec. 5.2.3). The user provides the fluid node where the heat load is applied and the name and location of a history file containing the heat load as a function of time.

Figure 73. Transient Heat Load option dialog.

5.4.2 Heat Exchanger

The Heat Exchanger option dialog is shown in figure 74. This option is activated from the Circuit Options pane on the Global Options menu (see sec. 5.2.2). The user has the option of modeling a Counter Flow or Parallel Flow heat exchanger. The user supplies the branch numbers that will be identified as the hot and cold branches as well as a value for heat exchanger effectiveness. If the user enters a heat exchanger effectiveness value between 0 and 1, GFSSP will perform calculations based on that effectiveness. If the user enters a value >1, this prompts GFSSP to internally calculate the effectiveness. In the latter case, the user must also supply a value of UA (the product of the overall conductance for heat transfer and the surface area on which that conductance is based).
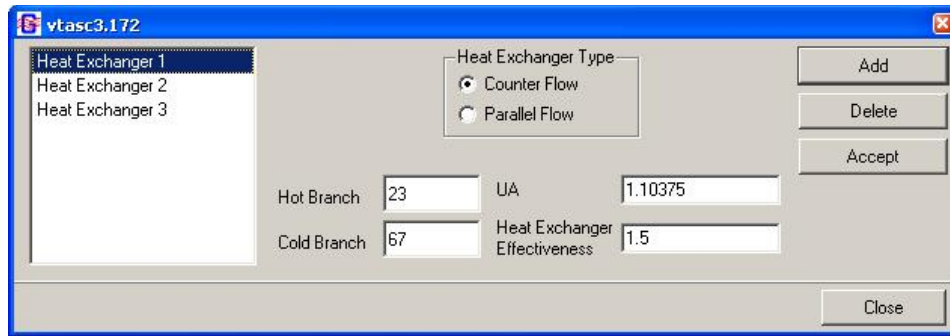
Figure 74. Heat Exchanger option dialog.

5.4.3 Tank Pressurization

The Tank Pressurization option dialog is shown in figure 75. This option is activated from the Unsteady Options pane on the Global Options menu (see sec. 5.2.4). The user has the option of modeling a vertically oriented cylindrical tank or a spherical tank. The user identifies the fluid nodes and fluid resistance branches in the model that represent the tank's ullage and propellant. The user is also asked to provide the initial surface areas where the ullage is interacting with (1) the propellant and (2) the tank wall. The user must also supply relevant tank characteristics (density, specific heat, and thermal conductivity of the tank wall material; tank wall thickness; and initial tank wall temperature). Finally, VTASC provides default values for the constants used in GFSSP's tank pressurization heat transfer calculations, but the user may modify these constants through this dialog if desired.
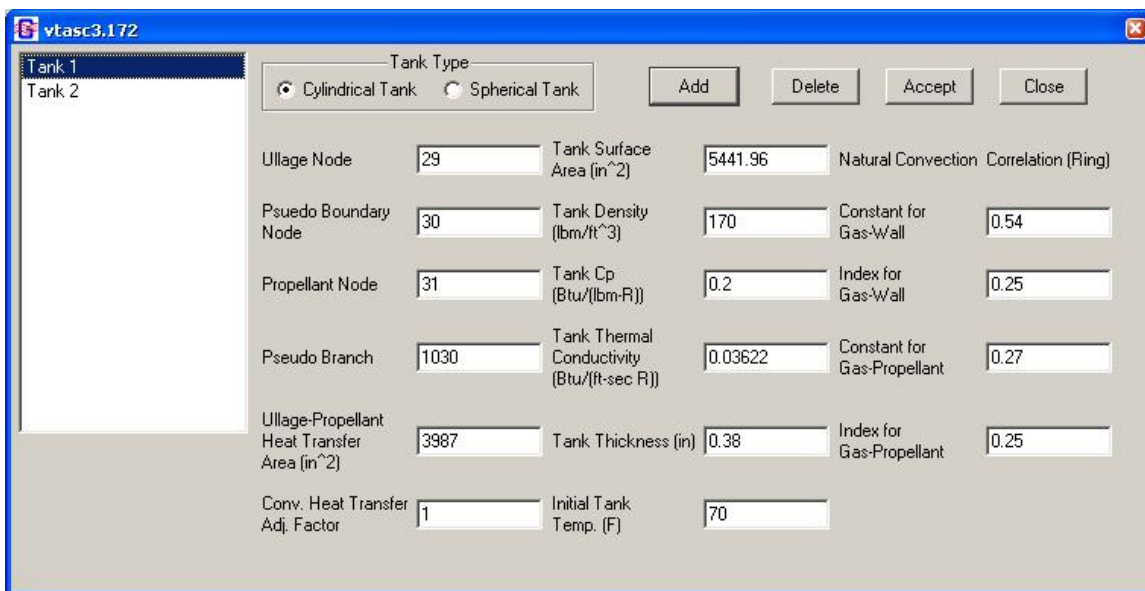
Figure 75. Tank Pressurization option dialog.

103

5.4.4 Turbopump

The Turbopump option dialog is shown in figure 76. This option is activated from the Circuit Options pane on the Global Options menu (see fig. 52(a)). The user supplies the fluid resistance branches that will represent the pump and the turbine. The user also supplies some characteristics of the turbine (speed, efficiency, diameter, and design point velocity ratio) as well as the name and location of a history file containing the pump characteristics (the quantities (Head/Speed^2) and (Torque/(Density*Speed^2)) as a function of (flow rate/speed).
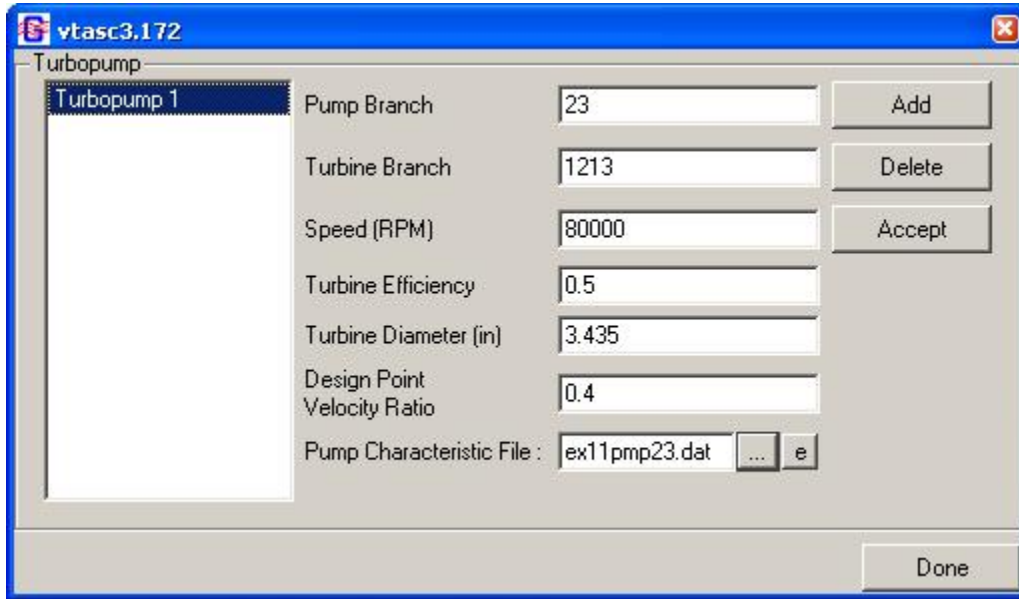
Figure 76. Turbopump option dialog.

5.4.5 Valve Open/Close

The Valve Open/Close option dialog is shown in figure 77. This option is activated from the Unsteady Options pane on the Global Options menu (see sec. 5.2.4). The user provides the fluid resistance branch that represents the valve and the name and location of a history file containing the cross-sectional flow area of the valve as a function of time.
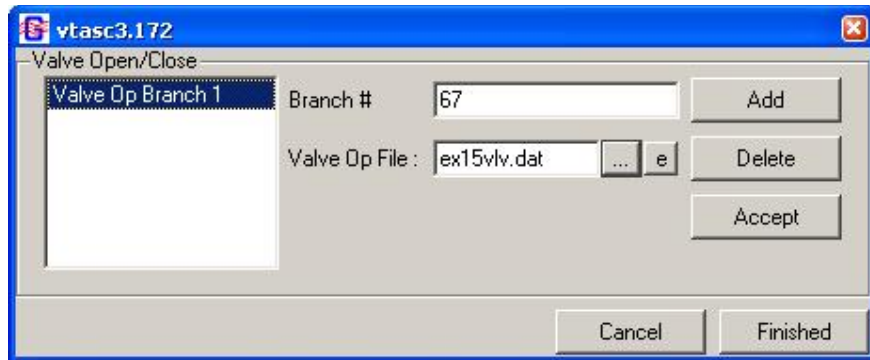
Figure 77. Valve Open/Close option dialog.

5.4.6 Fluid Conduction (Not Active)

The Fluid Conduction option dialog is shown in figure 78. This option is activated from the Circuit Options pane on the Global Options menu (see sec. 5.2.2). The user may populate the list of internal nodes in two ways. First, left mouse clicking the Load Nodes button will automatically populate the list with each internal node. Second, an individual internal node may be added to the list by typing the node identifier into the New Node input box and left mouse clicking the Add button. If the user wants to remove an internal node from the list, select that node from the list and left mouse click the Delete button. Selecting an internal node from the list reveals the list of upstream and downstream neighbors for that node. The user supplies the Area and Distance for each neighbor node by selecting that node from the Neighbor Nodes list.
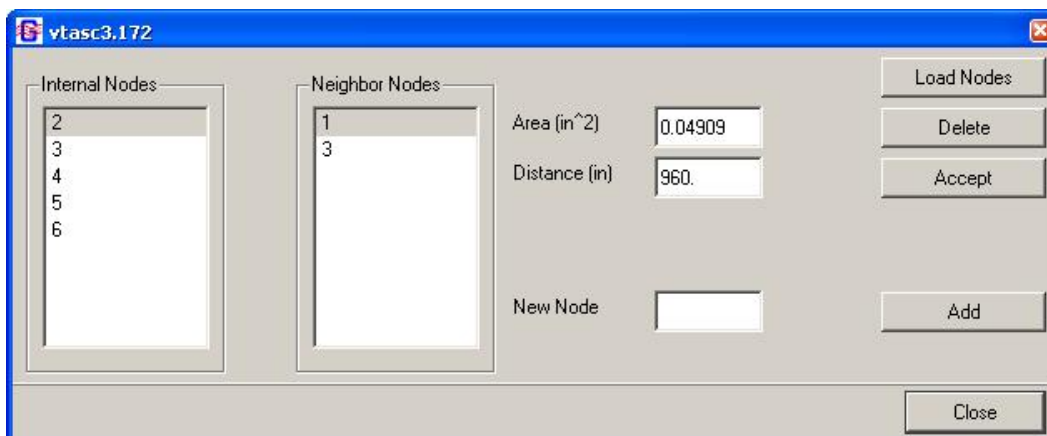
Figure 78. Fluid Conduction dialog.

5.4.7 Pressure Regulator

GFSSP offers a Pressure Regulator option to control the pressure downstream of a Restriction or Compressible Orifice branch. The Pressure Regulator option dialog is shown in figure 79.

105
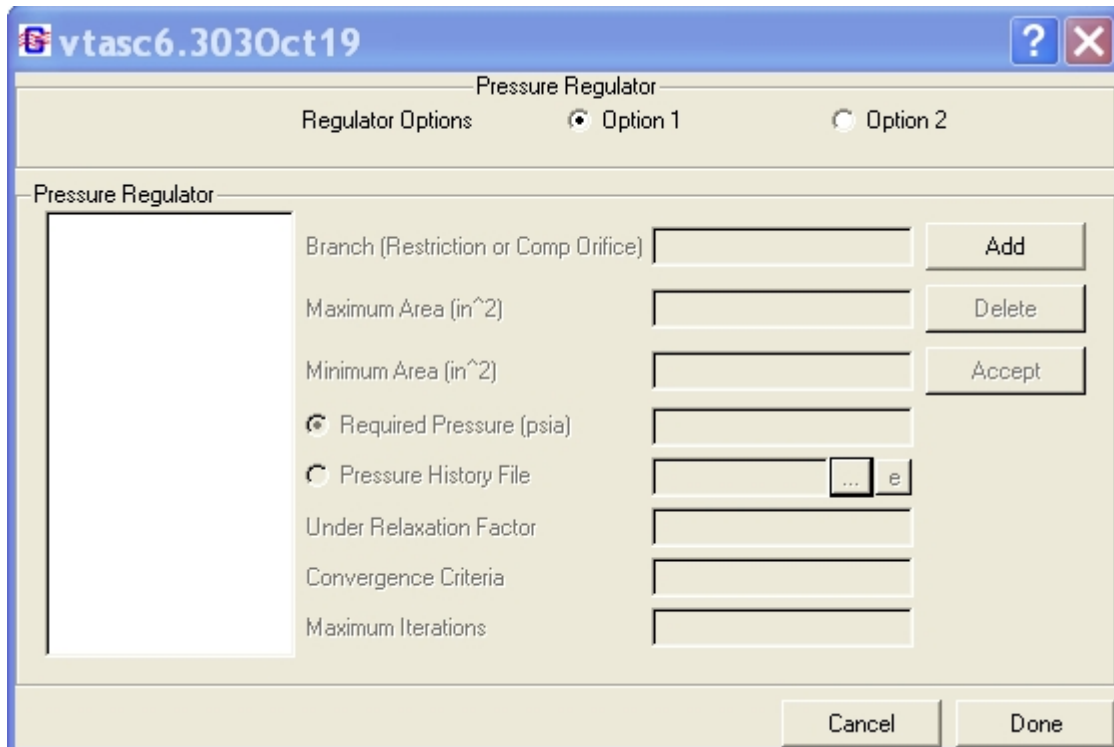
Figure 79. Pressure Regulator dialog.

This option is activated from the Unsteady Options pane on the Global Options menu (see sec. 5.2.4). The user is given two options: Iterative algorithm (Option 1) or Forward-looking algorithm (Option 2). With the Iterative algorithm, at each time step, the regulator area is adjusted until the desired pressure is achieved, or the maximum number of iterations is reached. This option behaves like an instantly responding regulator. Note that the many iterations per time step may slow running of the model and that only one Option 1 regulator is allowed per model. With the Forward-looking algorithm, the regulator area is adjusted just once per time step, based on an empirical relation developed by reference 27. This option may run faster, but the pressure may oscillate about the setpoint for several time steps until convergence is reached. Multiple Option 2 regulators are allowed.

The user may choose a constant required pressure, or define a pressure versus time profile in a history file. The format of the history file is:

```
NLINES
TIME1 P1
TIME2 P2
Etc….
```

NLINES is the number of TIME and P pairs in the file. TIME is given in seconds. Pressure is given in psia or kPa. The required pressure is then interpolated at each time step.

5.4.8 Flow Regulator

GFSSP offers a Flow Regulator option (fig. 80) to control the flow rate through a designated Restriction or Compressible Orifice branch. This option is activated from the Unsteady Options pane on the Global Options menu (see sec. 5.2.4).
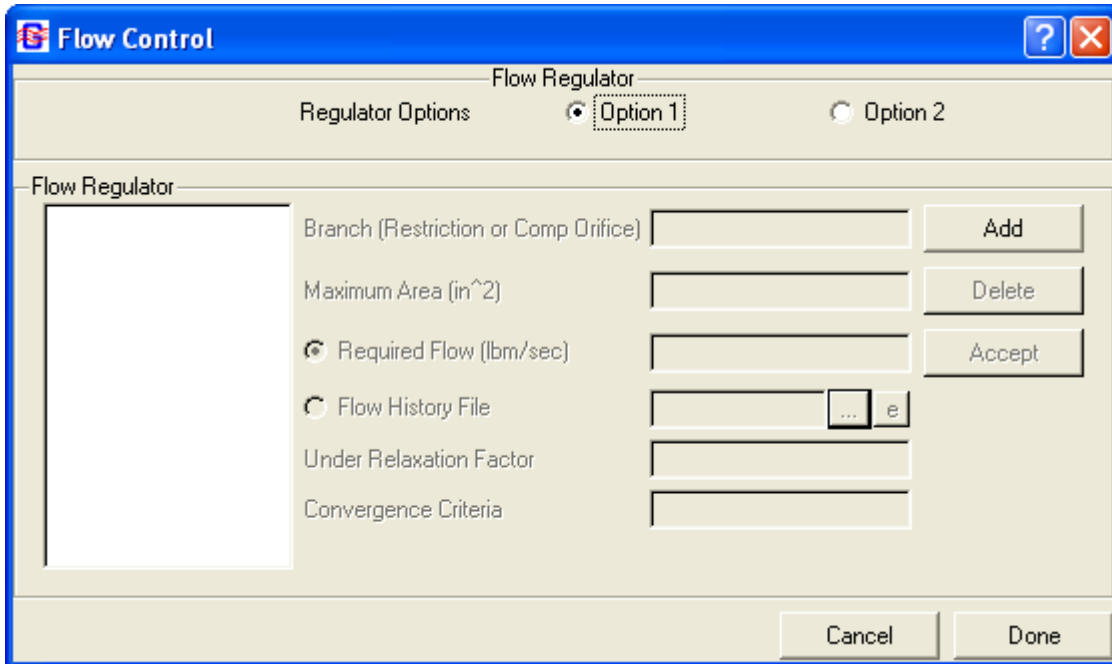
Figure 80. Flow Regulator dialog.

• Option 1—Iterative algorithm: In each time step, the regulator area is adjusted until the desired flow rate is achieved. This option acts like an instantly responding regulator. Note that the many iterations per time step may slow the model, and that only one regulator per model is allowed.

• Option 2—Marching algorithm: The regulator area is adjusted just once per time step. The area correction is based on the numerical derivative of flow rate versus area from the previous time steps. The flow rate may oscillate around the setpoint for several time steps until convergence is reached. Multiple regulators are allowed. Because other elements of the model may also affect the flow rate in the branch, the numerical derivative calculation is not exact, which may lead to improper area adjustment and an unstable model. Under-relaxation may improve the solution.

The user may choose a constant required flow rate, or one that varies with time based on a history file. The format of the history file is:

```
NLINES
TIME1 F1
TIME2 F2,
…
```

where NLINES is the number of history points, TIME is the time in seconds, and F is the desired flow rate in lb/s or kg/s.

5.4.9 Pressure Relief Valve

GFSSP allows the user to select a Restriction, Compressible Orifice, or Valve with Cv branch as a pressure relief valve. This normally closed branch will monitor the pressure differential between the upstream and downstream nodes, opening when the pressure differential exceeds a user-defined cracking pressure. The Pressure Relief Valve option dialog is shown in figure 81. This option is activated from the Unsteady Options pane on the Global Options menu (see sec. 5.2.4).
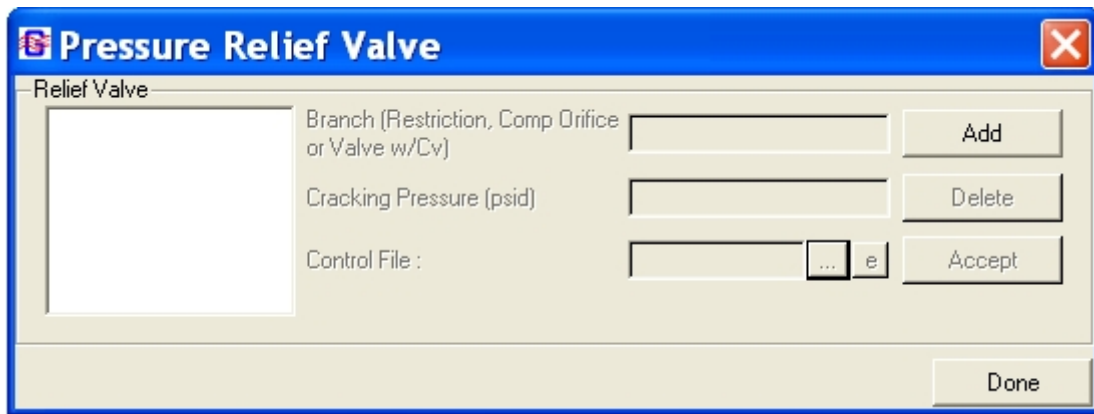
Figure 81. Pressure Relief Valve dialog.

Besides the cracking pressure (in psid or kPad), the user must supply a control file to describe the relief valve's operation. The format of the control file is:

```
NLINES
dP1 A or Cv1
dP2 A or Cv2
…
dPN A or CvN.
```

NLINES is the number of dP points in the file, and dP is the differential pressure across the valve, in psid or kPad. If using a Restriction or Compressible Orifice, A should be the area in in^2 or m^2. If using a Valve with Cv, Cv should be the value of Cv. The first pressure (dP1) is the reseating pressure, and is associated with a very small area (e.g., 1.0E-16 in^2) or small Cv (e.g. 1.0E-5).

The first pressure should be less than or equal to the cracking pressure. The last point (dPN) is the pressure at which the relief valve is fully open, and is associated with the fully open area/Cv. At pressures between the reseating pressure and the fully open pressure, the area or Cv of the valve will be interpolated. At pressures greater than the last point (dPN), the fully open area/Cv value is used.

5.4.10 Grid Generation

GFSSP provides the capability to generate a two-dimensional grid using Cartesian (X,Y) or Polar (R,Z) coordinates. After this feature is enabled on the Advanced Menu, the user selects the Grid Node on the left side of the main window (fig. 46). Left-clicking on the canvas will place a Grid Node on the canvas. This node is a placeholder substituting for the actual grid to be generated. Its properties will be the parameters of the generated grid. The grid is generated by right clicking on the Grid Node and selecting Generate Grid from the menu. This will produce the Grid Property dialog (see fig. 82), which is used to generate a grid, review the grid properties, and regenerate a grid with new parameters.
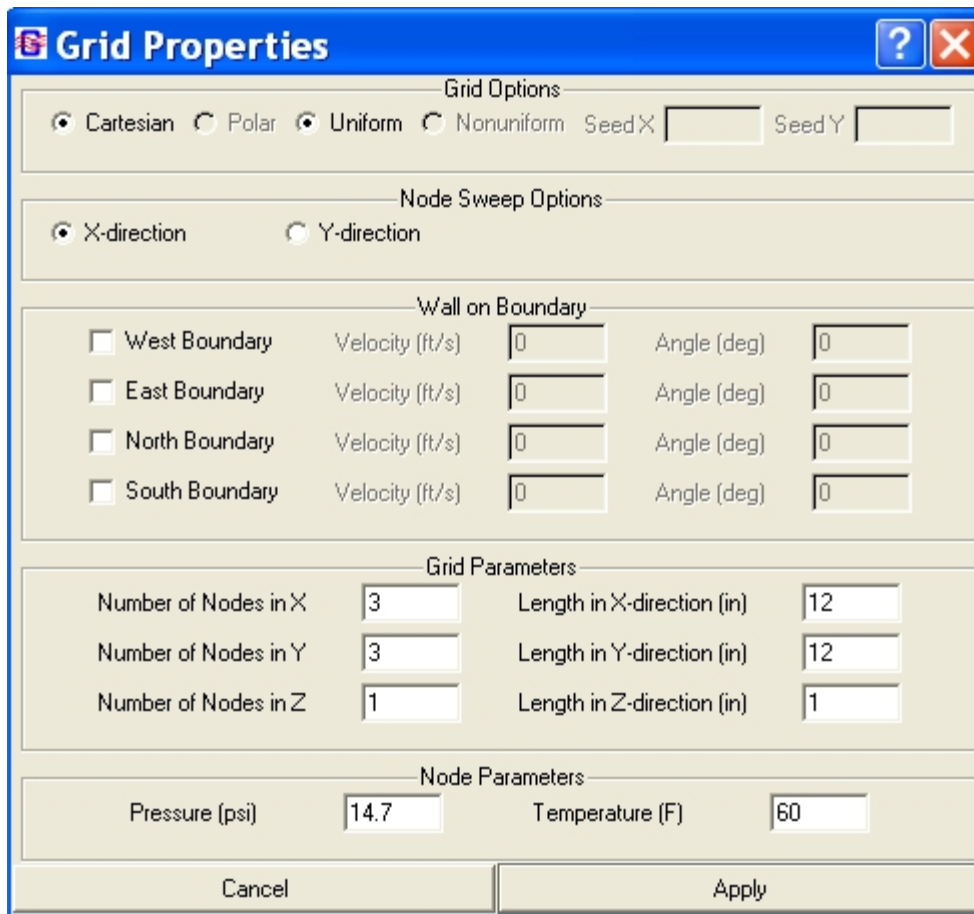
Figure 82. Grid Properties dialog.

This dialog is used to specify the parameters of the two-dimensional grid: Cartesian or Polar, Uniform or Nonuniform (requires seeds for X and Y distribution of nodes); direction of node sweep (increasing first in the *x*-direction or increasing first in the *y*-direction); whether boundary is a wall; velocity at the boundary; angle of the wall with the coordinate direction at boundary; overall length of the cavity, and number of nodes to distribute over that length; and default pressure and temperature for nodes.

When the grid generation is completed a message is displayed reminding that connectivity must be established between the grid and system components. The grid is not initially displayed on the canvas. It can be shown by right-clicking on the Grid Node and selecting Select Grid (fig. 83). Node and branch properties can be reviewed, but not modified. Right-click on the canvas to hide the grid. To establish connectivity between the generated grid and the system components, connect the Grid Node with desired branch option (fig. 84). Right click on that branch and select Resolve Grid Node Connection. This will produce the dialog shown in figure 85. Click on the desired generated node number, which will insert it into Connecting Node edit box, and click OK. The grid connecting node will have a green border when the generated grid is displayed.
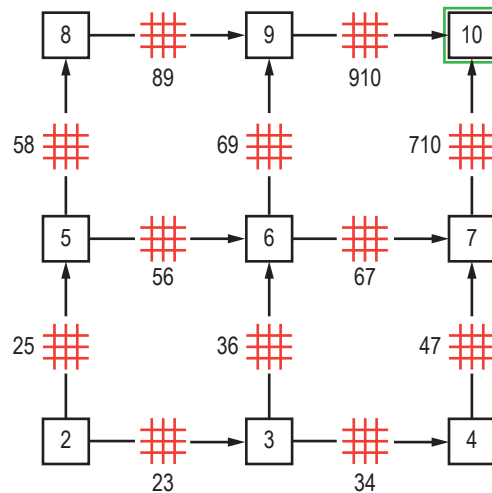
Figure 83. Generated grid (showing connecting node).

Figure 84. Circuit with grid node.

110
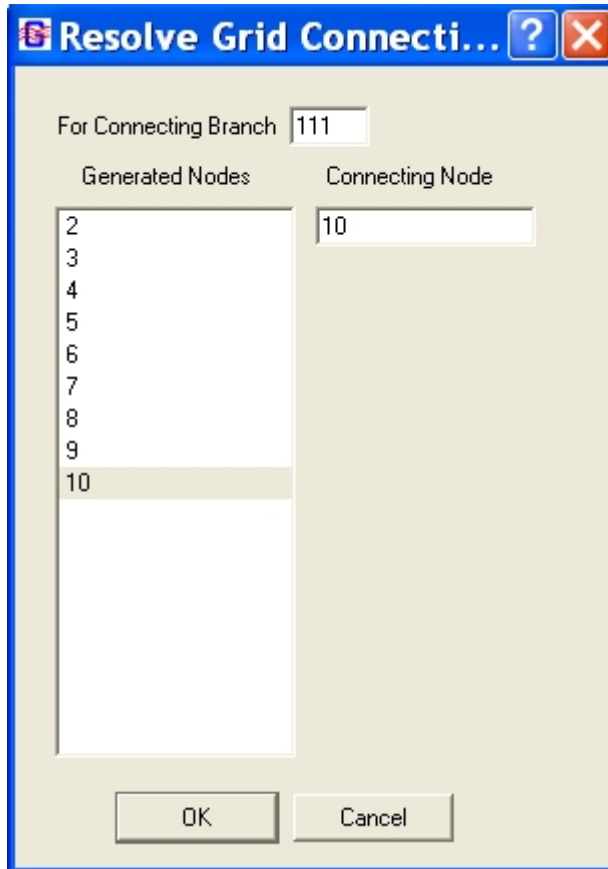
Figure 85. Resolve Grid Connectivity.

To review the generated grid parameters, right click on the Grid Node and select Properties, which will display the Grid Properties dialog. The grid can then be regenerated with new parameters by entering new parameters and clicking on Regenerate Grid. This will delete the existing grid and system connectivity, and create a new grid, which will need to be reconnected to the system components.

5.4.11 International Standard Units

VTASC provides the user the option to use SI units to specify values for properties being modeled. The SI values entered are converted to English units and written to the data file to be read by GFSSP. GFSSP output will be converted into SI units for reports and plotting. For users who use the User Subroutines, conversion (SI-English) functions are available to perform the unit conversions. A listing of the functions is found in userrtn604.for in appendix E.

5.5 GFSSP Input File

The primary interface between VTASC and GFSSP is the GFSSP input data file generated by VTASC. While it should not be necessary for the user to directly access the GFSSP input file

for most modeling activities, knowledge of the content and format of the GFSSP input data file may be helpful in some circumstances. This section is intended to provide the user with a helpful reference for better understanding the GFSSP input data file.

5.5.1 Title Information

The GFSSP input data file always begins with the title and documentary information for the model, which are shown below. The first two entries are internally defined by VTASC. The first, the GFSSP Version, tells GFSSP what features and formatting to expect while reading the input data file. The second entry defines where the user has installed GFSSP. The user defines the remaining four entries in VTASC. They include the analyst's name, the working directory path and input data file name, output data file name, and a descriptive title for the model.

```
GFSSP VERSION
604
GFSSP INSTALLATION PATH
C:\Program Files (x86)\GFSSP604\
ANALYST
ALOK MAJUMDAR
INPUT DATA FILE NAME
D:\GFSSP604Intel\ExamplesTest\Ex1\Ex1.dat
OUTPUT FILE NAME
Ex1.out
TITLE
Simulation of a Flow System Consisting of a Pump, Valve and Pipe Line
```

5.5.2 Option Variables

The GFSSP input data file includes all of the option variable values based on the user's choices in VTASC. The user is referred to appendix D for a specific definition of each logical variable.

```
USETUP
F
DENCON        GRAVITY   ENERGY    MIXTURE   THRUST    STEADY    TRANSV    SAVER
F             T         T         F         F         T         F         F
HEX           HCOEF     REACTING  INERTIA   CONDX     ADDPROP   PRINTI    ROTATION
F             F         F         F         F         F         T         F
BUOYANCY      HRATE     INVAL     MSORCE    MOVBND    TPA       VARGEO    TVM
F             T         F         F         F         F         F         F
SHEAR         PRNTIN    PRNTADD   OPVALVE   TRANSQ    CONJUG    RADIAT    Winplot
F             T         T         F         F         F         F         T
PRESS         INSUC     VARROT    CYCLIC    CHKVALS   WINFILE   DALTON    NOSTATS
F             F         F         F         F         T         F         F
NORMAL        SIMUL     SECONDL   NRSOLVT   IBDF      NOPLT     PRESREG   FLOWREG
F             T         T         F         1         T         0         0
TRANS_MOM     USERVARS  PSMG      ISOLVE    PLOTADD   SIUNITS   TECPLOT   MDGEN
F             F         F         1         F         F         F         F
NUM_USER_VARS IFR_MIX   PRINTD    SATTABL   MSORIN    PRELVLV   LAMINAR   HSTAG
1             1         F         F         F         F         T         T
```

112

5.5.3 Node, Branch, and Fluid Information

This section of the GFSSP input data file defines the basic scope of the model, including: (1) the total number of nodes, (2) the number of internal nodes, (3) the number of branches, and (4) the number of fluids:

```
NNODES   NINT   NBR    NF
 4       2      3      1
```

5.5.4 Solution Control Variables

The next section of the GFSSP input data file defines the numerical parameters chosen by the user, including the three under-relaxation parameters, the convergence criteria, and the maximum number of iterations:

```
RELAXK   RELAXD   RELAXH   CC        NITER   RELAXNR   RELAXHC   RELAXTS
 1       0.5      1        0.0001    500     1         1         1
```

5.5.5 Time Control Variables

This section of the GFSSP input data file is applicable only for unsteady models. It defines the time step, initial time, final time, output file print step, and the Winplot file print step.

```
DTAU    TIMEF   TIMEL   NPSTEP   NPWSTEP   WPLSTEP   WPLBUFF
1       0       200     25       1         50        1.1
```

5.5.6 Fluid Designation

This section of the GFSSP input data file lists the appropriate fluid definition information based on the user's selections in VTASC.

For a general fluid (GASP/WASP or GASPAK), the fluid designation lists the GFSSP index number for each selected fluid:

```
NFLUID(I),I =1,NF
 1   6   12
```

For a constant property fluid, the fluid designation lists the reference density and viscosity:

```
RHOREF    EMUREF
 62.4     0.00066
```

For an ideal gas fluid, the fluid designation lists the index number for an ideal gas and the reference properties associated with the ideal gas fluid:

```
NFLUID(I), I = 1, NF
 33
RREF    CPREF   GAMREF  EMUREF    AKREF     PREF    TREF    HREF   SREF
 53.34  0.24    1.3999  1.26e-05  4.133e-06 14.7    -459    0      0
```

For hydrogen peroxide, the fluid designation lists the index number and the mole fraction of water for the fluid:

```
NFLUID(I), I = 1, NF
34
MFRAC
0.5
```

Finally, for a user-defined fluid, the fluid designation lists the molecular weight of the fluid and the property table file names supplied by the user.

```
FLUID 1 PROPERTY FILES
28.0
AKFL1.DAT
RHOFL1.DAT
EMUFL1.DAT
GAMFL1.DAT
HFL1.DAT
SFL1.DAT
CPFL1.DAT
```

5.5.7 Node Numbering and Designation

The next section of the GFSSP input data file lists each node, designates whether that node is a boundary node (INDEX=2) or an internal node (INDEX=1), and includes a user supplied text description of the node. Nodes are listed in the order that they are created in VTASC, which may not be in numerical order.

```
NODE    INDEX    DESCRIPTION
 1      2        «Node 1»
 2      1        «Node 2»
 3      1        "Node 3"
 4      2        "Node 4"
```

If the user has chosen to activate buoyancy, the reference node will be defined in this section right below the node listing.

```
REFERENCE NODE FOR DENSITY
 2
```

5.5.8 Node Variables

The next section of the GFSSP input data file lists the initial properties at each node based on the user's selections in VTASC.

For a steady state model, the model boundary conditions are listed along with the internal node initial guesses. Concentrations are listed at each node in the same order the fluids are listed (see sec. 5.5.6). If the user has chosen the Constant Property Fluid option, the temperature will not appear in this listing.

114

```
NODE   PRES(PSI)   TEMP(DEGF)   MASS SOURC   HEAT SOURC   THRST AREA   CONCENTRATION
1      500         1500         0            0            0            0.1  0.9
2      500         80           0            0            0            1    0
3      338.2       1500         0            0            0            0.1  0.9
4      14.7        80           0            0            0            0.5  0.5
```

For an unsteady model, the internal node initial solution values are listed first in the same order they were created. These properties include the node volume property, which does not appear in a steady state model. After all of the internal nodes have been listed, each boundary node history file is listed (again, in the order they were created). Each boundary node requires a separate history file.

```
NODE   PRES(PSI)   TEMP(DEGF)   MASS SOURC   HEAT SOURC   THRST AREA   NODE-VOLUME   CONCENTRATION
1      100         80           0            0            0            17280
ex8hs2.dat
```

5.5.9 Transient Heat/Variable Geometry Information

This section of the GFSSP input data file is applicable only for unsteady models where the user has activated either the transient heat or variable geometry options.

If the user has elected to use the variable geometry option, the variable geometry file name will appear right below the last boundary node history file name. There is no description line associated with the variable geometry listing in the input data file.

The transient heat section of the GFSSP input data file first lists the number of nodes identified by the user as having a transient heat load. Each identified node is then listed along with the corresponding heat load file name in the order the user added their information in the Transient Heat dialog (see sec. 5.4.1).

```
Transient Heat Load Information
Number of Nodes with Transient Heat Loads
1
Transient Heat Node Number
2
Corresponding Heat Load History File Name
qdot.dat
```

5.5.10 Node-Branch Connections

The next section of the GFSSP input data file identifies which branches are attached to each internal node. Each internal node (variable INODE) is listed in the same order they were created. The variable NUMBR defines how many branches are attached to that node. The array NAMEBR identifies which branches are attached to that node in the order they were created.

```
INODE    NUMBR    NAMEBR
2        2        12   23
3        2        23   34
```

5.5.11 Branch Flow Designation and Resistance Options

The next section of the GFSSP input data file describes the characteristics of each fluid branch. This section consists of two subsections. The first subsection identifies the upstream node, downstream node, and branch resistance option chosen by the user. This subsection also includes any text description of that branch supplied by the user.

```
BRANCH   UPNODE   DNNODE   OPTION   DESCRIPTION
12       1        2        14       "Pump 12"
23       2        3        13       "Valve 23"
34       3        4        1        "Pipe 34"
```

The second subsection lists the properties of each fluid branch as defined by the user. The branch properties are specific to each branch resistance option. They are discussed in detail in section 3.1.7.

```
BRANCH   OPTION-14   PUMP CONST1   PUMP CONST2   PUMP CONST3   AREA
12                   30888         0             -0.0008067    201.06
BRANCH   OPTION-13   DIA           K1            K2            AREA
23                   6             1000          0.1           28.274
BRANCH   OPTION -1   LENGTH        DIA           EPSD          ANGLE     AREA
34                   18000         6             0.005         95.74     28.274
```

5.5.12 Unsteady Information

The 12th section of the GFSSP input data file is applicable only for unsteady models. This section defines the initial mass flow rates in each fluid branch in the order the branches were created.INITIAL FLOW RATES IN BRANCHES FOR UNSTEADY FLOW

```
1001    0.803
1002    0.803
1003    0.423
1004    0.423
```

5.5.13 Inertia Information

This section of the GFSSP input data file is applicable only for models where the user has activated the Inertia option. This section is divided into four subsections. The first subsection defines a fluid branch's relationship with any upstream branches. Branches are listed in the order they were created. The variable NOUBR defines the number of upstream branches connected to a particular branch. The array NMUBR identifies which branches are attached upstream of that branch in the order they were created. The second subsection defines a fluid branch's relationship with any downstream branches. Each branch is listed in the order they were created. The variable NODBR defines the number of downstream branches connected to a particular branch. The array NMDBR identifies which branches are attached downstream of that branch in the order they were created.

```
BRANCH    NOUBR    NMUBR
12        0
23        1        12
34        1        23
BRANCH    NODBR    NMDBR
12        1        23
23        1        34
```

The third subsection allows the user to define relative angles between branches if desired. As discussed in section 5.2.2, the user must edit this section of the input data file manually. VTASC supplies a template for each branch in the order they were created. The template lists each upstream branch and corresponding angle first, then each downstream branch and corresponding angle. VTASC defines each angle as a placeholder value of zero degrees, which the user must replace with the appropriate angles.

```
BRANCH
12
 UPSTRM BR.    ANGLE
 DNSTRM BR.    ANGLE
 23            0.00000
BRANCH
23
 UPSTRM BR.    ANGLE
 12            0.00000
 DNSTRM BR.    ANGLE
 34            0.00000
```

The fourth subsection identifies the fluid branches where the user has activated Inertia. The first number (16 in the example below) defines the number of branches where the user has activated inertia. The subsequent lines list each branch where the user has activated inertia in the order they were created.

```
NUMBER OF BRANCHES WITH INERTIA
16
12
23
34
45
56
67
78
89
910
1011
1112
1213
1314
1415
1516
1617
```

5.5.14 Fluid Conduction Information (Not Active)

This section of the GFSSP input data file is applicable only for models where the user has activated the Fluid Conduction option. This section is divided into two subsections. The first subsection identifies how many fluid conduction nodes the user has selected. For each of these fluid nodes, the number of upstream and downstream nodes connected to that node (identified here as neighbors) is defined and each neighbor node is listed.

```
NUMBER OF FLUID CONDUCTION NODES
2
NODE   NO. OF NEIGHBORS   NEIGHBOR NODES
2      2                  1  3
3      2                  2  4
```

The second subsection defines the fluid conduction properties (area and internode distance) for the interaction between a particular node and each of its neighbors. For each node, the property of interest between that node and each neighbor is listed in the order the neighbor nodes are listed in the first subsection.

```
NODE    CONDUCTION AREAS
 2      0.04909  0.04909
 3      0.04909  0.04909
NODE    INTERNODE DISTANCES
 2      960  960
 3      960  960
```

5.5.15 Rotation Information

This section of the GFSSP input data file is applicable only for models where the user has activated the Rotation option. This section first defines the number of branches where the user has activated Rotation. Then, each branch is identified along with the rotational information for that branch (upstream and downstream node radii from center of rotation, rotational speed, and the rotational 'slip' factor).

```
NUMBER OF ROTATING BRANCHES
9
BRANCH    UPST RAD    DNST RAD    RPM     K ROT
23        1.25        2.25        5000    0.8671
34        2.25        3.625       5000    0.8158
45        3.625       4.6875      5000    0.763
56        4.6875      5.375       5000    0.7252
67        5.375       5.5         5000    0.7076
89        5.5         5.375       5000    0.7129
910       5.375       4.6875      5000    0.7349
1011      4.6875      3.625       5000    0.7824
1112      3.625       2.65        5000    0.8376
```

5.5.16 Valve Open/Close Information

This section of the GFSSP input data file is applicable only for models where the user has activated the Valve Open/Close option. This section first defines the number of fluid branches where the user will be modeling a valve transient. Next, for each valve, the branch that will represent the valve and the valve history file name are listed.

```
NUMBER OF CLOSING/OPENING VALVES IN THE CIRCUIT
1
BRANCH
67
FILE NAME
ex15vlv.dat
```

5.5.17 Momentum Source Information

This section of the GFSSP input data file is applicable only for models where the user has activated the Momentum Source option. This section first defines the number of fluid branches where the user wishes to add a momentum source. Next, each branch where the user has defined a momentum source is listed along with the momentum source.

```
NUMBER OF BRANCHES WITH MOMENTUM SOURCE
1
BRANCH    MOMENTUM SOURCE
12        100
```

5.5.18 Heat Exchanger Information

This section of the GFSSP input data file is applicable only for models where the user has activated the Heat Exchanger option. First, the number of heat exchangers identified by the user is defined. Then, the characteristics of each heat exchanger are listed as defined by the user, including the 'hot' and 'cold' branches, the type of heat exchanger (Counter Flow=1, Parallel Flow=2), the hot and cold surface areas, UA, and the heat exchanger effectiveness. Note that the hot and cold surface areas are not currently recommended for use in GFSSP and cannot be modified using VTASC.

```
NUMBER OF HEAT EXCHANGERS
1
IBRHOT   IBRCLD   ITYPHX   ARHOT   ARCOLD   UA       HEXEFF
23       67       1        0       0        1.1038   1.5
```

5.5.19 Moving Boundary Information

This section of the GFSSP input data file is applicable only for models where the user has activated the Moving Boundary option. This section defines the number of nodes identified as having moving boundary, and lists each identified node:

```
NUMBER OF NODES WITH MOVING BOUNDARY
2
NODE
1
2
```

5.5.20 Turbopump Information

This section of the GFSSP input data file is applicable only for models where the user has activated the Turbopump option. First, the number of turbopumps in the model is listed. Then, the characteristics for each turbopump (fluid branch representing the pump, fluid branch representing the turbine, speed, turbine efficiency, turbine diameter, design point velocity ratio, and the pump characteristics curve file name) are listed.

```
NUMBER OF TURBOPUMP ASSEMBLY IN THE CIRCUIT
1
IBRPMP   IBRTRB   SPEED(RPM)   EFFTURB   DIATRB   PSITRD
23       1213     80000        0.5       3.435    0.4
PUMP CHARACTERISTICS CURVE DATA FILE
ex11pmp23.dat
```

5.5.21 Tank Pressurization Information

This section of the GFSSP input file is applicable only for models where the user has activated the Tank Pressurization option. First, the number of pressurized propellant tanks in the model is defined. Next, the characteristics of each tank are listed including: the tank type (spherical=0, cylindrical=1); fluid node representing the ullage; ullage-propellant interface pseudo

boundary node; fluid node representing the propellant; fluid branch representing the propellant surface; the initial tank wall surface area exposed to the ullage; the tank wall thickness; the tank wall material density, specific heat and thermal conductivity; the ullage-propellant interface surface area; the heat transfer coefficient adjustment factor; the initial tank wall temperature; the heat transfer correlation ullage-propellant constants; and the heat transfer correlation ullage-tank wall constants.

```
NUMBER OF PRESSURIZATION PROPELLANT TANKS IN CIRCUIT
1
TNKTYPE   NODUL   NODULB   NODPRP    IBRPRP   TNKAR     TNKTH    TNKRHO    TNKCP
1         2       3        4         34       6431.9    0.375    170       0.2
TNKCON    ARHC    FCTHC    TNKTM     CIP      FNIP      CIW      FNIW
0.0362    4015    1        -264      0.27     0.25      0.54     0.25
```

5.5.22 Variable Rotation Information

This section of the GFSSP input file is applicable only for models where the user has activated the Variable Rotation option. The variable rotation history file name is listed in this section.

```
ROTATION DATA FILE
varrot.dat
```

5.5.23 Pressure Regulator Information

This section of the GFSSP input file is applicable only for models where the user has activated the Pressure Regulator option and specified regulator branches.

```
NUMBER OF PRESSURE REGULATOR ASSEMBLY IN THE CIRCUIT
1
PRESS REG BR   HIST FILE  MAX_AREA   PRESSURE  RELAXATION   CONVERGENCE  MAX_ITERATIONS  MIN_AREA
12             1          1.44       40        0.3          0.0001       50              1e-16
PRESSURE REGULATOR HISTORY FILE
preg_hist.dat
```

5.5.24 Flow Regulator Information

This section of the GFSSP input file is applicable only for models where the user has activated the Flow Regulator option and specified regulator branches.

```
NUMBER OF FLOW REGULATOR ASSEMBLY IN THE CIRCUIT
1
FLOW REG BR    HIST FILE   AREA      REGULATOR FLOW   RELAXATION   CONVERGENCE
12             1           0.3       0.012            1            0.001
FLOW REGULATOR HISTORY FILE
freg_hist.dat
```

5.5.25 Phase Separation Model Information

This section of the GFSSP input file is applicable only for models where the user has activated the Phase Separation Model option and applicable nodes PSM checked.

```
NUMBER OF PHASE SEPARATION MODEL NODES
1
PHASE SEPARATION MODEL NODE LIST
4
```

5.5.26 Pressure Relief Valve Information

This section of the GFSSP input file is applicable only for models where the user has activated the Pressure Relief valve option and specified valve branches:

```
NUMBER OF PRESSURE RELIEF ASSEMBLIES IN THE CIRCUIT
 1
RELIEF VALVE BR   CRACKING PRESSURE (psid)
12                12
CORRESPONDING CONTROL FILE
ControlFile.dat
```

5.5.27 Conjugate Heat Transfer Information

This section of the GFSSP input file is applicable only for models where the user has activated Conjugate Heat Transfer. The section is divided into seven subsections.

The first subsection identifies how many solid and ambient nodes are present in the model as well as how many conductors of each type (solid-solid conduction, solid-fluid, solid-ambient, and solid-solid radiation) are present in the model.

```
NSOLID   NAMB   NSSC   NSFC   NSAC   NSSR
8        2      7      8      2      1
```

The second subsection defines the characteristics of each solid node in the order they were created. The properties of each solid node (material, mass, and initial temperature) are listed first. Then, the number of conductors of each type attached to the solid node is listed along with a text description of the solid node provided by the user. Finally, for each type of conductor, every conductor attached to that solid node is listed.

```
NODESL   MATRL   SMASS     TS         NUMSS   NUMSF   NUMSA   NUMSSR   DESCRIPTION
2        41      1.00000   70.00000   1       1       1       1        "S Node 2"
NAMESS
23
NAMESF
122
NAMESA
12
NAMESSR
24
```

The third subsection lists each ambient node in the order they were created and lists the temperature at that node along with a text description of the ambient node provided by the user.

```
NODEAM    TAMB        DESCRIPTION
1         32.00000    "A Node 1"
```

The fourth subsection lists each solid-solid conduction conductor along with its characteristics ('upstream' solid node, 'downstream' solid node, surface area, distance and a user-supplied text description) in the order they were created:

```
ICONSS   ICNSI   ICNSJ   RCSIJ    DISTSIJ   DESCRIPTION
23       2       3       3.1415   3.00000   "Conductor 23"
```

The fifth subsection lists each solid-fluid conductor along with its characteristics including solid node; fluid node; heat transfer coefficient model (User Supplied=0, Dittus-Boelter=1, and Miropolskii=2); surface area, user-supplied heat transfer coefficient (if Model=0); emissivity of the solid; emissivity of the fluid; and user-supplied text description.

```
ICONSF  ICS  ICF  MODEL  ARSF         HCSF         EMSFS         EMSFF        DESCRIPTION
122     2    12   0      1.88500e+01  3.17000e-04  0.00000e+00   0.00000e+00  «Convection 122»
```

The sixth subsection lists each solid-ambient conductor along with it characteristics (solid node, ambient node, surface area, heat transfer coefficient, emissivity of the solid, emissivity of the ambient, and a user-supplied text description).

```
ICONSA   ICSAS   ICSAA   ARSA         HCSA         EMSAS        EMSAA        DESCRIPTION
12       2       1       3.14159e+00  2.00000e-02  0.00000e+00  0.00000e+00  «Convection 12»
```

The seventh subsection lists each solid-solid radiation conductor along with its characteristics ('upstream' solid node, 'downstream' solid node, 'upstream' surface area, 'downstream' surface area, view factor, 'upstream' emissivity, 'downstream' emissivity, and a user-supplied text description).

```
ICONSSR  ICNSRI  ICNSRJ  ARRSI    ARRSJ    VFSIJ    EMSSI    EMSSJ    DESCRIPTION
24       2       3       3.14159  3.14159  1.00000  0.70000  0.70000  «Conductor 24»
```

5.5.28 Restart Information

This section of the GFSSP input file is applicable only for models where the user has elected to Read From and/or Write to Restart Files. The section lists the node and branch restart file names.

```
RESTART NODE INFORMATION FILE
FNDEX15.DAT
RESTART BRANCH INFORMATION FILE
FBREX15.DAT
```

5.5.29 Cyclic Boundary Information

This section of the GFSSP input file is applicable only for models where the user has activated the Cyclic Boundary option. The section lists the boundary node where the Cyclic Boundary option has been activated and the node that is upstream of the cyclic boundary node.

```
CYCLIC    BNDARY NODE    UPSTREAM NODE
1         22
```

5.6 User Executable

Advanced users may wish to take advantage of the User Subroutines to model physics not directly available in VTASC. In order to do this, the user will need access to a FORTRAN compiler so that a specialized GFSSP executable may be created. If the user has the Compaq Visual FOR-TRAN compiler or Intel FORTRAN compiler installed on their computer, they can create a user executable through the Build dialog (activated from the Module menu).

Figure 86 shows the Build dialog. The user supplies the User Subroutine file name and location (if other than the model's working directory). The user may browse to find the appropriate User Subroutine file and can edit the file using the buttons at the right of the User Module File text box. VTASC will automatically identify the default GFSSP, GASPAK, and GASP Object File names and locations based on the user's GFSSP installation directory. If for some reason, the user wishes to use some other object file versions, they may browse to the desired object file locations using the buttons at the right of each text box. When the user clicks the Build button, VTASC uses the Compaq Visual FORTRAN or Intel compiler to create a specialized user executable. The interaction between VTASC and the compiler is shown to the user with the display pane on the Build dialog. Any compilation errors or warnings will appear on the display pane. If the user wishes to stop the build process for any reason, clicking the Stop button will cancel the build session. If the build is successful, a specialized GFSSP executable named after the User Subroutine is created and saved to the model's working directory. Also, the user executable filename is automatically saved as the GFSSP executable filename on the User Information tab of the General Information item of the Options menu (see sec. 5.2.1). The Close button will exit the Build dialog and return the user to the VTASC window.
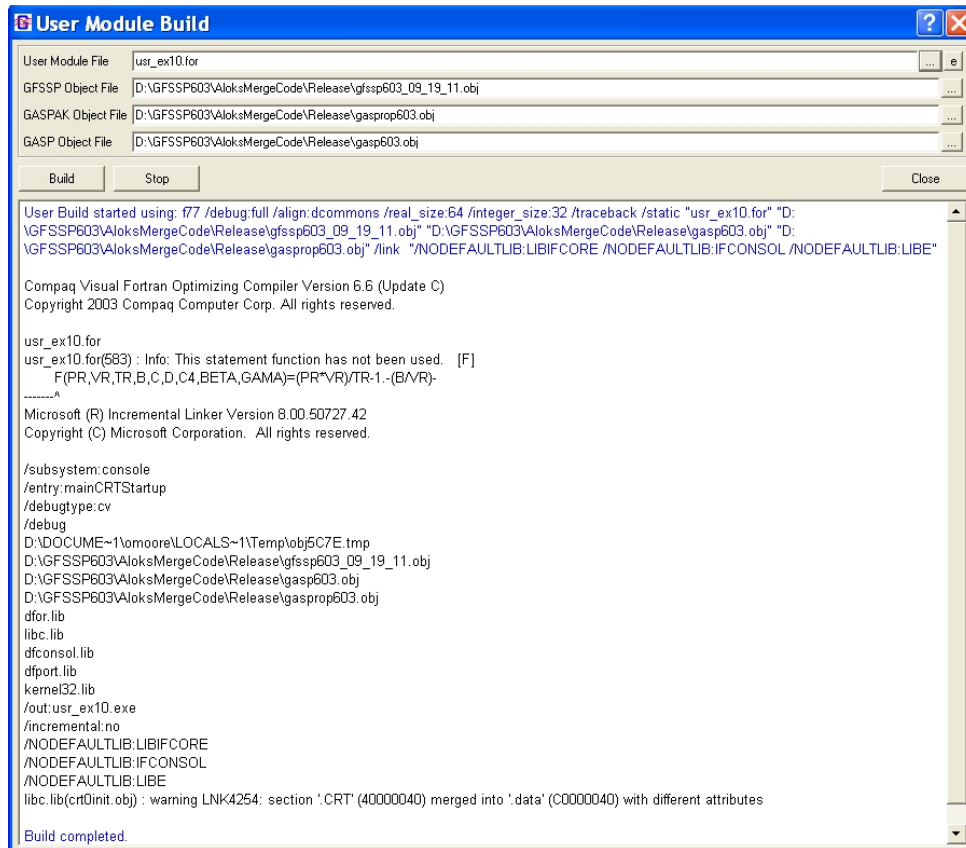
Figure 86. User Executable Build dialog.

5.7 GFSSP Execution

As noted in section 5.1.4, GFSSP can be executed directly from the VTASC environment using either the Run menu or the shortcut on the file Input/Output toolbar. When the user activates the Run GFSSP command, VTASC automatically writes the GFSSP text input file before executing GFSSP. If the text file already exists, VTASC will ask if the user wishes to overwrite the file. VTASC then executes GFSSP and opens the GFSSP Run Manager window. The appearance and function of the Run Manager depends on whether the model is a steady state or transient model.

5.7.1 Steady State Run Manager

Figure 87 shows the GFSSP Run Manager appearance for a steady state simulation. If the user wishes to stop the GFSSP simulation for any reason, the button in the upper left corner of the Run Manager will stop GFSSP execution. During execution, the Run Manager will display GFSSP generated messages in the GFSSP display pane. After execution is complete, the GFSSP messages may be printed by clicking the Print button in the lower right corner of the Run Manager. The GFSSP generated text output file may also be viewed after execution is complete by clicking the Edit Output button in the lower left corner of the Run Manager. Clicking the Close button will exit the Run Manager and return the user to the VTASC window.
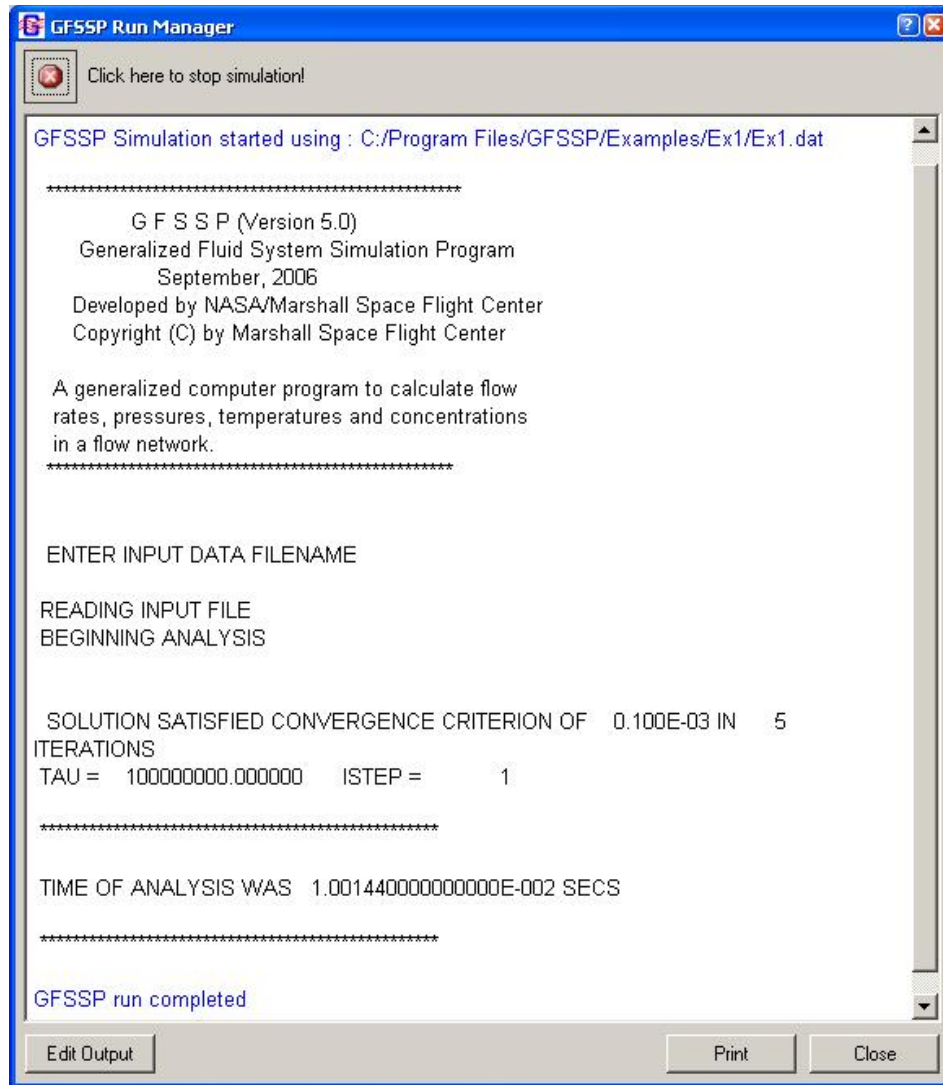
Figure 87. GFSSP steady state run manager.

5.7.2 Unsteady Run Manager

Figure 88 shows the GFSSP Run Manager appearance for an unsteady simulation. The only difference between the unsteady and steady Run Managers is that the GFSSP display pane appears in the top half of the Run Manager and a real-time updated plot of GFSSP's convergence behavior as a function of time appears in the bottom half of the Run Manager. See General Information, Output Control tab to stop receiving this convergence data.
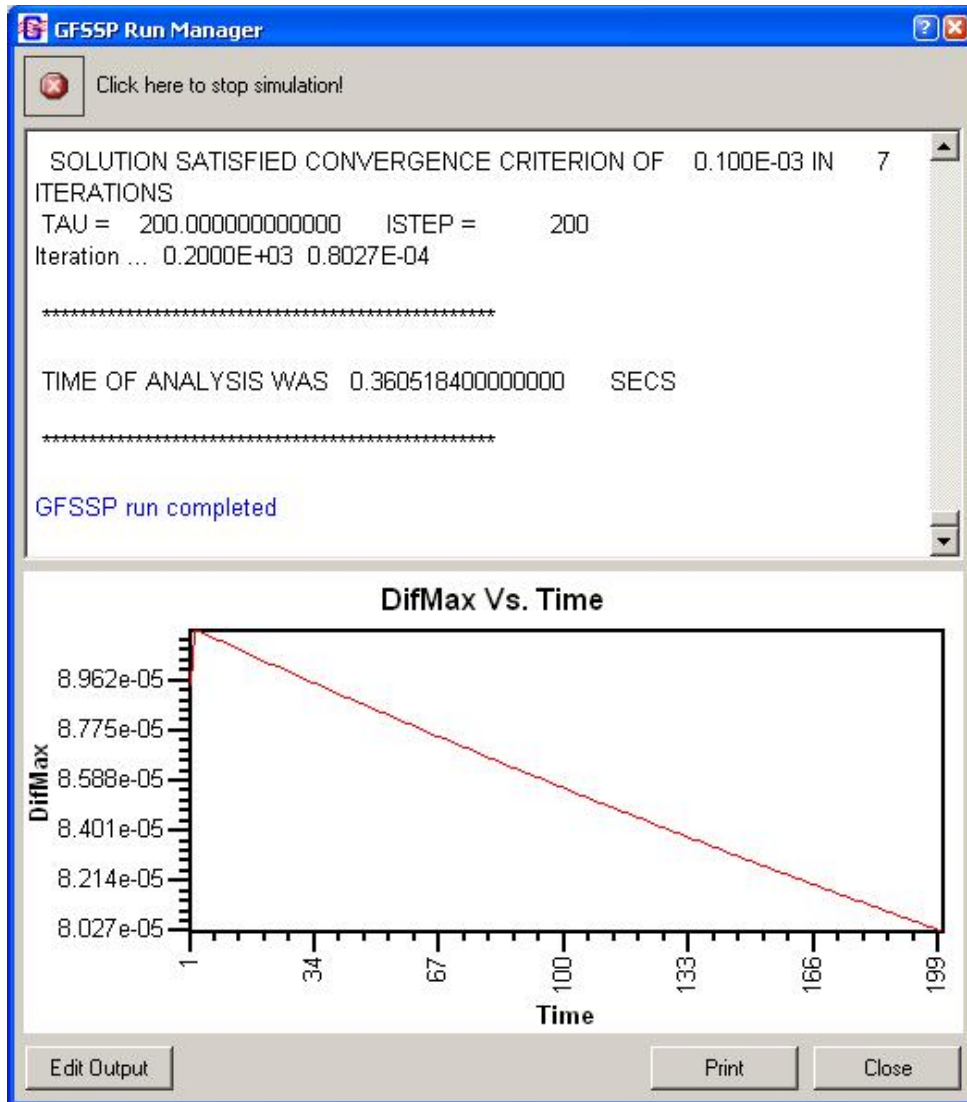
Figure 88. GFSSP unsteady run manager.

5.8 GFSSP Output File

The basic GFSSP output for any simulation is the text output file. As mentioned in the previous section, once a simulation is complete the Run Manager gives the user the option of viewing the output file in a text editor. The content of the output file is dependent on the options selected by the user during VTASC model development. This section is intended to give the user an understanding of the format and general layout of the text output file.

5.8.1 Title and Data Files

Each GFSSP output file begins with the header shown below, which identifies the version of GFSSP that was used for the model simulation.

```
****************************************************
              G F S S P (Version 603)
         Generalized Fluid System Simulation Program
                    October 2011
              Built with Compaq Compiler
          To be used with VTASC 6.3, Build 603.201

        Developed by NASA/Marshall Space Flight Center
        Copyright (C) by Marshall Space Flight Center


      A generalized computer program to calculate flow
      rates, pressures, temperatures, and concentrations
      in a flow network.
****************************************************
```

Directly below the GFSSP header, the run date, the model title, analyst name, model working directory and text input file name, and text output file name as defined by the user are supplied.

```
                    RUN DATE:10/11/2011 16:22


TITLE  :Simulation of a Flow System Consisting of a Pump, Valve and Pipe Line
ANALYST:ALOK MAJUMDAR
FILEIN :C:\Program Files\GFSSP603\Examples\Ex1\Ex1.dat
FILEOUT:Ex1.out
```

5.8.2 Logical Variables

This section of the GFSSP output file lists the logical variable definitions as used in the simulation.

```
                            OPTION VARIABLES

ADDPROP    BUOYANCY  CONDX     CONJUG    CYCLIC    DALTON    DENCON    ENERGY
F          F         F         F         F         F         F         T
FLOWREG    GRAVITY   HCOEF     HEX       HRATE     IFRMIX    INERTIA   INSUC
F          F         F         F         T         1         F         F
INVAL      MIXTURE   MOVBND    MSORCE    NORMAL    NRSOLVT   OPVALVE   PLOTADD
F          F         T         F         F         F         F         F
PRESREG    PRESS     PRINTI    PRNTADD   PRNTIN    RADIATION REACTING  ROTATION
0          F         F         T         F         F         F         F
SAVER      SECONDL   SHEAR     SIMULA    SIUNITS   STEADY    THRUST    TPA
F          T         F         T         F         F         F         F
TRANS_MOM  TRANSQ    TRANSV    TVM       TWOD      USRVAR    VARGEO    VARROT
F          F         T         F         F         F         T         F
```

5.8.3 Node and Branch Information

This section of the output file documents the size and scope of the model. It lists the total number of nodes as well as the number of internal nodes, the number of branches, the number of fluids, the number of variables (or equations) in the model (this is the sum of NBR and NINT), and finally the enthalpy reference node, which is hard-coded in GFSSP as the second node.

128

```
NNODES = 4
NINT   = 2
NBR    = 3
NF     = 1
NVAR   = 5
NHREF  = 2
```

5.8.4 Fluid Information

This section of the output file documents the fluids that were used in the simulation.

For a constant property fluid, the fluid information lists the reference density and viscosity:

```
RHOREF = 62.4000 LBM/FT**3
EMUREF = 0.6600E-03 LBM/FT-SEC
```

For all other fluid options, the fluid information lists each fluid in the order they were entered by the user as shown below.

```
FLUIDS: O2  H2O
```

5.8.5 Boundary Conditions

This section of the output file documents the boundary conditions of the model, which were supplied at each boundary node by the user. For a model with multiple fluids, the pressure, temperature, density, thrust surface area of the node, and concentration of each fluid at that node are listed. A single fluid model provides the same listing with the exception of the fluid concentrations. A constant property fluid model lists only the pressure and surface area.

```
BOUNDARY NODES
NODE    P            T            RHO          AREA         CONCENTRATIONS
        (PSI)        (F)          (LBM/FT^3)   (IN^2)       O2           H2O
1       0.5000E+03   0.1500E+04   0.3931E+00   0.0000E+00   0.1000E+00   0.9000E+00
2       0.5000E+03   0.8000E+02   0.2819E+01   0.0000E+00   0.1000E+01   0.0000E+00
4       0.1470E+02   0.8000E+02   0.4725E+02   0.0000E+00   0.5000E+00   0.5000E+00
```

5.8.6 Fluid Network Information

This section of the output file is only active if the user has selected to print network information from the Global Options dialog (see sec. 5.2.1). For each internal node, it lists the thrust surface area, mass source, and heat source designated by the user. For each branch, it reprints the branch flow designation and resistance option information from the input text file (see sec. 5.5.11).

```
INPUT SPECIFICATIONS FOR INTERNAL NODES
NODE    AREA         MASS         HEAT
        (IN^2)       (LBM/S)      (BTU/S)
2       0.0000E+00   0.0000E+00   0.0000E+00
3       0.0000E+00   0.0000E+00   0.0000E+00
```

```
BRANCH    UPNODE    DNNODE    OPTION
12        1         2         14
23        2         3         13
34        3         4         1
BRANCH OPTION -14   PUMP CONST1    PUMP CONST2    PUMP CONST3    AREA
12                  0.309E+05      0.000E+00      -0.807E-03     0.201E+03
BRANCH OPTION -13   DI             K1             K2             AREA
23                  0.600E+01      0.100E+04      0.100E+00      0.283E+02
BRANCH OPTION -1    LENGTH         DIA            EPSD           ANGLE       AREA
34                  0.180E+05      0.600E+01      0.500E-02      0.957E+02   0.283E+02
```

5.8.7 Initial Field Information

This section of the output file is only active if the user has selected to Print the Initial Field from the Global Options dialog (see sec. 5.2.1).

For each internal node of a single fluid model, it lists the initial guesses for pressure and temperature as well as the resulting compressibility, density, and quality from the thermodynamic property calculations. For a multiple fluid model, the list is the same except that the quality is replaced with the initial guesses for the mass concentration of each fluid. For a constant property fluid model, only the initial guess for pressure is listed.

```
INITIALGUESS FOR INTERNAL NODES
NODE    P(PSI)      TF(F)       Z(COMP)     RHO         QUALITY
                                            (LBM/FT^3)
2       0.1470E+02  0.6000E+02  0.7616E-03  0.6237E+02  0.0000E+00
3       0.1470E+02  0.6000E+02  0.7616E-03  0.6237E+02  0.0000E+00
```

For each branch the trial solution for the pressure drop across the branch and the mass flow rate in the branch is listed.

```
TRIAL SOLUTION
BRANCH    DELP(PSI)    FLOWRATE(LBM/SEC)
12        0.0000       0.0100
23        0.0000       0.0100
34        0.0000       0.0100
```

5.8.8 Conjugate Heat Transfer Network Information

This section of the output file is only active for models where the user has activated Conjugate Heat Transfer and selected to print network information from the Global Options dialog (see sec. 5.2.1). For each solid node and conductor, it reprints the conjugate heat transfer information from the input text file (see sec. 5.5.25)

```
CONJUGATE HEAT TRANSFER
NSOLIDX = 8
NAMB    = 2
NSSC    = 7
NSFC    = 8
NSAC    = 2
```

130

```
NSSR    = 0
NODESL   MATRL   SMASS   TS        NUMSS   NUMSF   NUMSA
2        41      1.0000  70.0000   1       1       1
NAMESS
23
NAMESF
122
NAMESA
12
NODEAM   TAMB
1        32.0000
10       212.0000
ICONSS   ICNSI   ICNSJ   ARCSIJ    DISTSIJ
23       2       3       3.1416    3.0000
34       3       4       3.1416    3.0000
ICONSF   ICS     ICF     ARSF      EMSFS    EMSFF
122      2       12      18.8500   0.0000   0.0000
123      3       12      18.8500   0.0000   0.0000
ICONSA   ICSAS   ICSAA   ARSA      HCSA       EMSAS       EMSAA
12       2       1       0.3142E+01  0.2000E-01  0.0000E+00  0.0000E+00
910      9       10      0.3142E+01  0.2000E-01  0.0000E+00  0.0000E+00
```

5.8.9 Solution Results

This section of the output file documents the solution results of the GFSSP model. If the model is unsteady, a solution will be output at each time step the user has chosen to print (defined by the Print Frequency as discussed in sec. 5.2.3). The first line in the solution will list the current time step and the time at this step.

```
 ISTEP = 25     TAU = 0.25000E+02
```

Next, the unsteady model will print out the boundary conditions at each boundary node for that time step. The format is identical to that discussed in section 5.8.5.

```
BOUNDARY NODES
NODE    P(PSI)       TF(F)        Z(COMP)      RHO          QUALITY
                                               (LBM/FT^3)
2       0.1470E+02   0.8000E+02   0.1000E+01   0.7355E-01   0.0000E+00
```

After this line, the solution will be output in the same format for a steady model or the time step of interest in an unsteady model. For each internal node in a single fluid model, the calculated pressure, temperature, compressibility, density, resident mass, and fluid quality are listed. The listing is identical for a multiple fluid model except that the quality is replaced with the calculated mass concentration of each fluid at that node. For a constant property fluid model, only the calculated pressure and resident mass are listed. Note that for a steady model, the resident mass will always be zero.

```
SOLUTION
 INTERNAL NODES
NODE   P(PSI)       TF(F)       Z           RHO         EM(LBM)     QUALITY
                                            (LBM/FT^3)
2      0.2290E+03   0.6003E+02  0.1186E-01  0.6241E+02  0.0000E+00  0.0000E+00
3      0.2288E+03   0.6003E+02  0.1185E-01  0.6241E+02  0.0000E+00  0.0000E+00
```

If the user elects to print extended information in the Global Options dialog (see sec. 5.2.1), the output file will next list the calculated enthalpy, entropy, viscosity, thermal conductivity, specific heat, and specific heat ratio for each internal node. Note that this information will not be printed for constant property fluid models.

```
NODE   H            ENTROPY     EMU         COND        CP          GAMA
       BTU/LB       BTU/LB-R    LBM/FT-SEC  BTU/FT-S-R  BTU/LB-R
2      0.2869E+02   0.5542E-01  0.7542E-03  0.9523E-04  0.1000E+01  0.1003E+01
3      0.2869E+02   0.5542E-01  0.7542E-03  0.9523E-04  0.1000E+01  0.1003E+01
```

For each branch, the calculated resistance factor, pressure drop, mass flow rate, velocity, Reynolds number, Mach number, entropy generation, and lost work are listed.

```
BRANCHES
BRANCH KFACTOR      DELP        FLOW        VELOCITY    REYN.       MACH        ENTROPY     LOST
       (LBF-S^2/    (PSI)       RATE        (FT/SEC)    NO.         NO.         GEN.        WORK
       (LBM-FT)^2)  (LBM/SEC)                                      BTU/(R-SEC) LBF-FT/SEC
12     0.000E+00    -0.214E+03  0.191E+03   0.219E+01   0.241E+06   0.183E-02   0.000E+00   0.000E+00
23     0.764E-03    0.193E+00   0.191E+03   0.156E+02   0.644E+06   0.130E-01   0.210E-03   0.848E+02
34     0.591E+00    0.214E+03   0.191E+03   0.156E+02   0.644E+06   0.130E-01   0.162E+00   0.657E+05
```

If the second law is used to solve the energy equation, the total entropy generation and work lost will be listed directly below the branch solution information.

```
       ***** TOTAL ENTROPY GENERATION = 0.163E+00 BTU/(R-SEC) ******

       ***** TOTAL WORK LOST = 0.120E+03 HP *****
```

If the user has activated conjugate heat transfer for a model, the conjugate heat transfer results will be listed next in the output file.

For each solid node, the specific heat that was used is listed along with the calculated solid temperature. Note that the specific heat will be zero for steady models.

```
SOLID NODES
NODESL  CPSLD       TS
        BTU/LB F    F
2       0.000E+00   0.423E+02
3       0.000E+00   0.569E+02
```

For each solid to solid conductor, the thermal conductivity that was used is listed along with the calculated heat transfer rate.

```
SOLID TO SOLID CONDUCTOR
ICONSS    CONDKIJ       QDOTSS
          BTU/S FT F    BTU/S
23        0.261E-02     -0.333E-02
34        0.261E-02     -0.279E-02
```

For each solid to fluid conductor, the calculated heat transfer rate is listed along with the convection and radiation heat transfer coefficients that were used.

```
SOLID TO FLUID CONDUCTOR
ICONSF    QDOTSF       HCSF        HCSFR
          BTU/S        BTU/S       FT**2 F
122       -0.115E-02   0.317E-03   0.000E+00
123       -0.544E-03   0.317E-03   0.000E+00
```

For each solid to ambient conductor, the calculated heat transfer rate is listed along with the convection and radiation heat transfer coefficients that were used.

```
SOLID TO AMBIENT CONDUCTOR
ICONSA    QDOTSA       HCSA         HCSAR
          BTU/S        BTU/S FT**2  F BTU/SFT**2 F
12        0.448E-02    0.200E-01    0.000E+00
910       -0.136E-01   0.200E-01    0.000E+00
```

For each solid to solid radiation conductor, the calculated heat transfer rate is listed along with the effective conductivity that was used.

```
SOLID TO SOLID RADIATION CONDUCTOR
ICONSSR   QDOTSSR      EFCSSR
          BTU/S        BTU/S F
79        -0.113E-06   0.421E-08
```

If the user has requested that axial thrust be calculated in the Global Options dialog (see sec. 5.2.2), the calculated axial thrust will be listed next in the output file:

```
AXIAL THRUST = -527.30169 LBF
```

If the user has activated the Turbopump advanced option (see sec. 5.4.4), the turbopump output will be listed next in the output file. First, the number of turbopumps in the model is listed (note that this value is not labeled in the output file). Then, the pump branch, turbine branch, speed, the turbine efficiency at the design point, the turbine velocity ratio at the design point, the required torque and the horsepower are listed for each turbopump in the model.

```
  1
 IBRPMP    IBRTRB    SPEED(RPM)   ETATRB      PSITR       TORQUE(LB-IN)    HPOWER
 23        1213      0.800E+05    0.578E+00   0.269E+00   0.511E+02        0.649E+02
```

If the user has activated the Pressurization advanced option (see sec. 5.4.3), the pressurization output will be listed next in the output file. First, the number of pressurization tanks in the model is listed. Then, the ullage node, propellant node, ullage to propellant heat transfer

rate, ullage to tank wall heat transfer rate, tank wall conduction heat transfer rate, tank wall temperature, propellant volume, and ullage volume are listed for each pressurization tank. Note that the labels for this output do not include the units. The units are Btu/s for the heat transfer rates, degrees Rankine for the tank wall temperature, and cubic feet for the volumes.

```
NUMBER OF PRESSURIZATION SYSTEMS = 1
NODUL   NODPRP  QULPRP  QULWAL  QCOND   TNKTM     VOLPROP   VOLULG
2       4       0.6644  2.1888  0.0000  195.6238  473.0886  26.9114
```

5.8.10 Convergence Information

The final section of the output file contains information on the convergence of the solution. It is important to remember that GASP/WASP allows extrapolation outside the stated limits of its fluid property relationships. While this allows for flexibility during the iterative process, it can occasionally lead to a final solution based on extrapolated properties. For models where the user has selected GASP/WASP, GFSSP checks the pressure and temperature at each node and prints a warning in the output file if they are outside of GASP/WASP's stated limits so that the user can verify that the results are reasonable.

```
WARNING! CHKGASP: T out of fluid property range at node  1
WARNING! CHKGASP: T out of fluid property range at node  3
```

GFSSP also prints a statement indicating whether or not the solution converged. For an unsteady model, this statement is printed at each time step. If the solution converges, the statement lists the convergence criteria and the number of iterations needed to reach convergence.

```
SOLUTION SATISFIED CONVERGENCE CRITERION 0.100E-03 IN 5 ITERATIONS
```

If the solution does not converge, the statement lists the convergence criteria, the number of iterations that were performed and the maximum difference after the last iteration.

```
SOLUTION DID NOT SATISFY CONVERGENCE CRITERION 0.100E-02 IN 541 ITERATIONS
DIFMAX IN SUCCESSIVE ITERATION = 0.175E-02
```

If the model includes the cyclic boundary option (see sec. 5.2.2), the number of adjustment iterations and the final temperature difference are listed next in the output file.

```
 ITERADJC = 3 DIFTEM = 1.634E-16
```

This section of the output file is only active if the user has selected to print the initial field from the Global Options dialog (see sec. 5.2.1). The time and time step will be listed after the convergence information. For a steady model, the time will read 100,000,000 s and the time step will be 1.

```
 TAU = 100000000.000000   ISTEP = 1
```

The final section of the output file lists the CPU time to complete the model simulation.

```
*************************************************

 TIME OF ANALYSIS WAS 1.001440000000000E-002 SECS

*************************************************
```

5.9 Post-Processing Simulation Data

While the GFSSP output file provides a comprehensive summary of model simulation results, it is not always a practical source of information to meet the user's needs. Therefore, VTASC provides alternative methods of viewing GFSSP output for both steady and unsteady simulations.

5.9.1 Steady State Simulation Results

As mentioned in section 5.3, each GFSSP solution model element (fluid internal nodes and branches, conjugate heat transfer solid nodes and conductors) has a Results… dialog option located on their respective popup menus. After running a steady state simulation, if the user selects the Results option for a particular element, a table of results at that location will be displayed.

Figure 89 shows the internal fluid node results table for a steady state simulation with multiple fluids. For this case, the table includes the calculated pressure, temperature, compressibility, density, resident mass, and the mass concentration of each fluid at that node. The internal fluid node table contents will vary just like the fluid node solution results discussed in section 5.8.9 based on the user's selections (multiple fluid, single fluid, constant property fluid, and print extended information). Clicking the OK button will close the Results option and return the user to VTASC.

Figure 89. GFSSP steady state simulation results internal fluid node table.

The results tables for the fluid branches, solid nodes, and conductors are the same in appearance and function as the internal fluid node table discussed above. The parameters that are listed in each table are the same parameters discussed in section 5.8.9 for each respective element.

5.9.2 Unsteady Simulation Results

There are two options available in VTASC for generating plots of unsteady simulations, depending on the user's needs: VTASC Plot and Winplot. There is also a provision of using Tecplot for plotting two-dimensional flow field.

5.9.2.1 VTASC Plot. The VTASC Plot is a built-in plotting capability within VTASC. As with the steady state results tables, it is accessed by selecting the Results… dialog from the desired model element's popup menu. The appearance and function of VTASC Plot is the same for each model element. The only difference will be the parameters available to plot, which are the same parameters discussed for each element in section 5.8.9.

Figure 90 shows the Results dialog for an unsteady simulation. Initially, the plot canvas space will be blank. The user can generate a hard copy or a bitmap of the desired plot by clicking the Print or Print to Bitmap button, respectively. The user creates the desired plot using the Properties dialog, which is activated by clicking the Properties… button. Once the user has finished, clicking the Close button will end the Results dialog and return the user to VTASC.
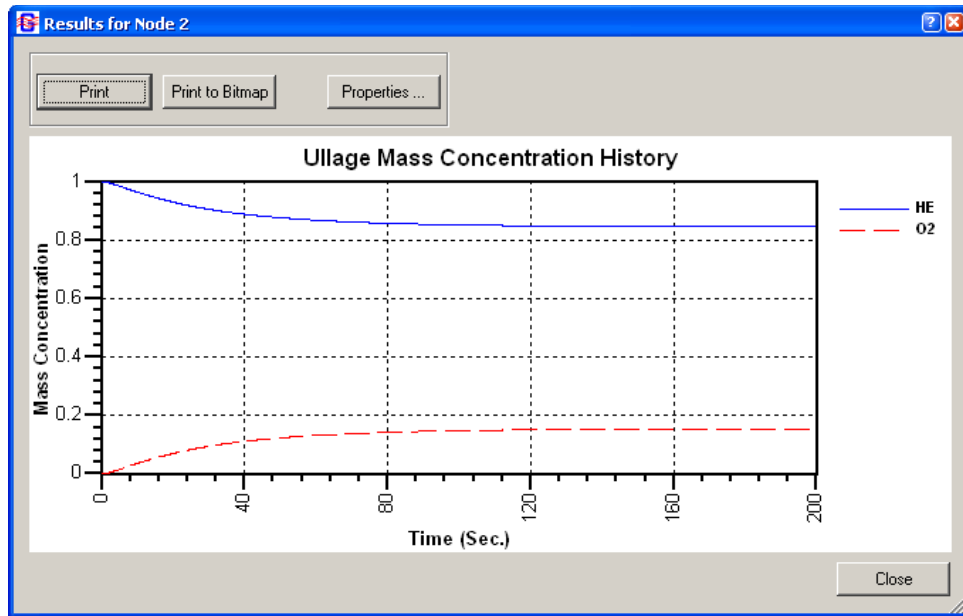
Figure 90. GFSSP Results dialog for unsteady simulation.

Figure 91 shows the Properties dialog used to create a plot of unsteady results. The dialog consists of three tabs, as well as Apply, OK, and Cancel buttons. Clicking the Apply button accepts any changes that have been made. The OK button closes the Properties dialog and returns to the Results dialog. The Cancel button fulfills the same function as the OK button.

(a)

(b)

138

(c)
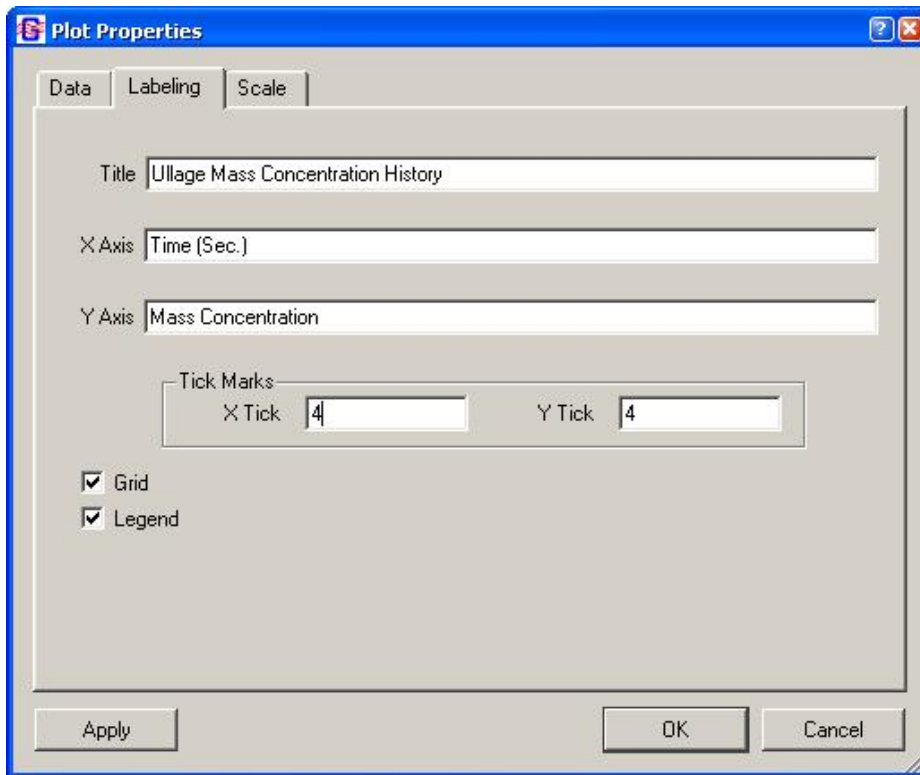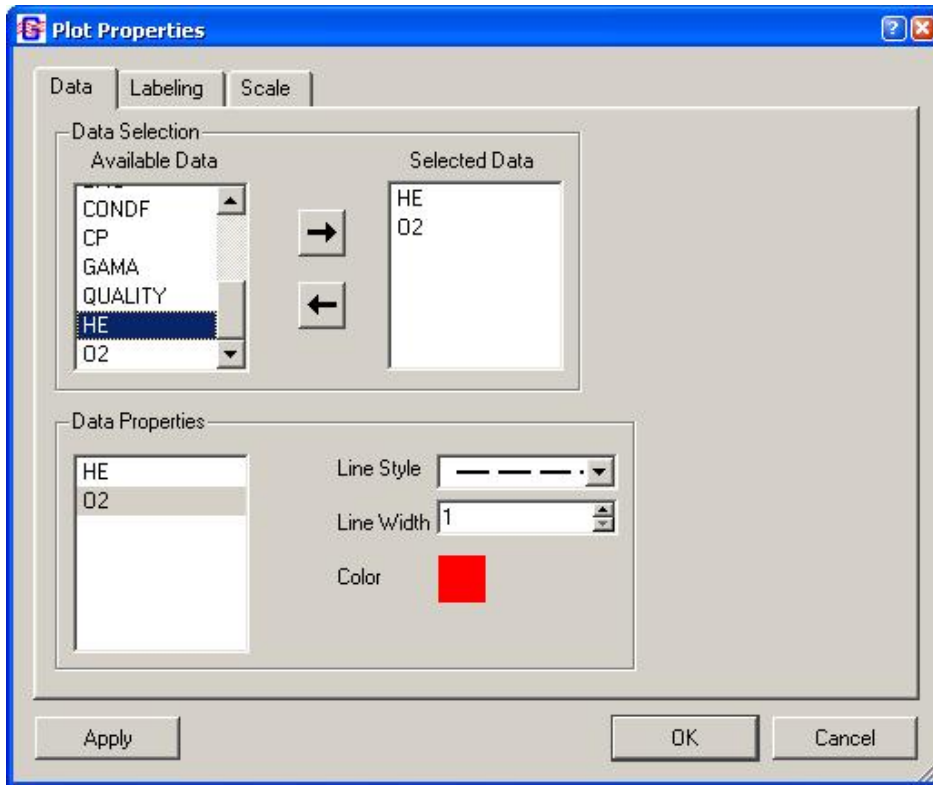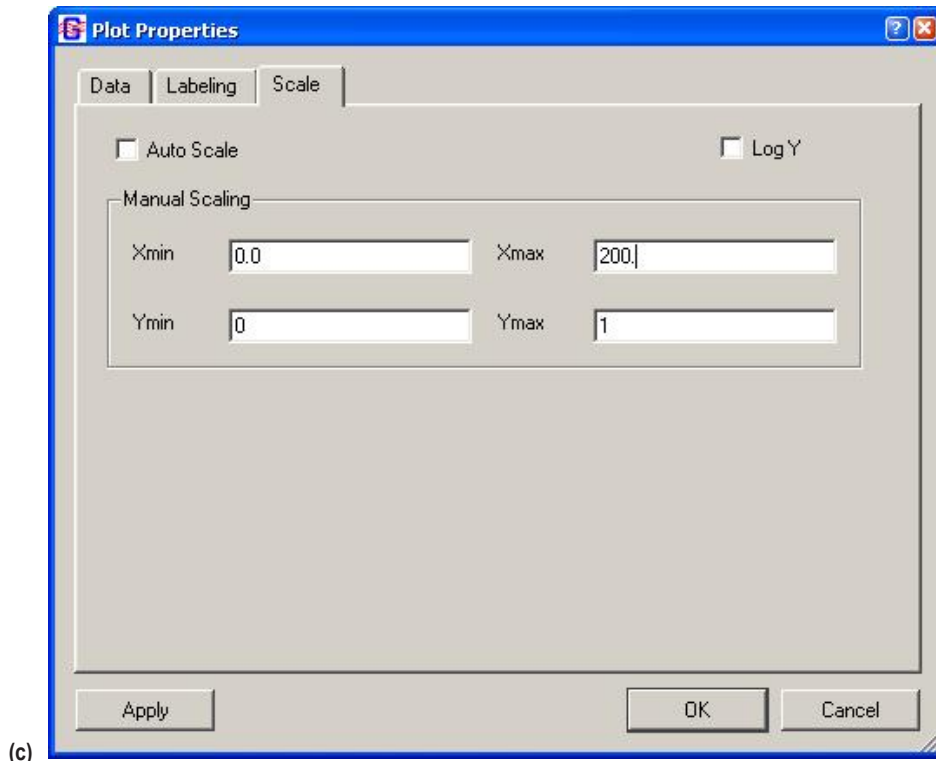
Figure 91. GFSSP VTASC plot Properties dialog: (a) Data tab, (b) labeling tab,
and (c) scale tab.

The first tab (fig. 91(a)) allows the user to define the data they wish to plot. The user may select parameters for plotting using the Data Selection list at the top of the tab. The user selects plot parameters by highlighting the desired parameters in the Available Data list and clicking the ➡ button. These parameters will then be added to the Selected Data list. If the user wishes to remove parameters from the Selected Data list, highlight those parameters and click the ⬅ button. Note that VTASC Plot does not have Multi-Y axis plotting capability so scale should be considered when plotting multiple parameters on a single plot. The Data Properties list at the bottom of the tab can be used to design the line style of each plot parameter. The user highlights the parameter whose line style they wish to design and then selects the line type, width, and color from the available selections on the right. Once all changes have been made to a particular parameter, click the Apply button to accept the changes.

The second tab (fig. 91(b)) allows the user to define the Labeling parameters. Titles may be written or modified for the X and Y axes as well as the overall plot. The user may select whether or not they wish to include a grid or a legend on the plot by clicking the appropriate checkbox. The user may also define the number of minor tick marks they wish to see for each axis. Once all changes have been made, click the Apply button to accept the changes.

The third tab (fig. 91(c)) allows the user to modify the scale of each axis. By default, VTASC will auto scale a plot for the user. Deselecting the Auto Scale checkbox allows the user to define the minimum and maximum values for each axis. The user also has the option of converting the Y axis to a log scale. Once all changes have been made, click the Apply button to accept the changes.

139

5.9.2.2 Winplot. If the user selects the Winplot plotting option from the Global Options dialog (see sec. 5.2.1), unsteady plot files will be generated in either comma delimited or binary formats. If the user selects the comma delimited option, several files are generated. The naming convention, description, and available parameters for each file are listed in table 10. Note that the conjugate heat transfer related output files are only written if a node or conductor of that type is present in the model. If the user selects the binary format, a single file with the name convention 'filename.WPL' will be generated. This file will contain all of the available parameters for each model element as discussed in section 5.8.9 as well as DIFMAX, RSDMAX, and ITER.

Table 10. Winplot comma delimited unsteady output files.

Naming Convention	Description	Parameters
*filename*FN.CSV	Fluid node results	P (psia), T (°F), Z, ρ (lbm/ft^3), x of fluid or c of each fluid, μ (lbm/ft-s), k (Btu/ft-s-R), V (ft^3), DIFMAX, RSDMAX, ITER
*filename*B.CSV	Fluid branch results	v (ft/s), DP (psid), $\dot{m}$ (lbm/s), $\dot{S}_{gen}$ (Btu/R-s)
*filename*SN.CSV	Solid node results	$C_{p,s}$ (Btu/lbm-°F), T_s (°F)
*filename*SF.CSV	Solid-fluid conductor results	h_{csf} (Btu/s-ft^2-°F), $h_{csf,rad}$ (Btu/s-ft^2-°F) if needed, $\dot{Q}_{sf}$ (Btu/s)
*filename*SS.CSV	Solid-solid conductor results	k_{ss} (Btu/ft-s-R), $\dot{Q}_{ss}$ (Btu/s)
*filename*SA.CSV	Solid-ambient conductor results	h_{csa} (Btu/s-ft^2-°F), $h_{csa,rad}$ (Btu/s-ft^2-°F) if needed, $\dot{Q}_{sa}$ (Btu/s)
*filename*SSR.CSV	Solid-solid radiation results	$k_{eff,ssr}$ (Btu/ft-s-R), $\dot{Q}_{ssr}$ (Btu/s)

As mentioned in section 5.1.4, Winplot must be obtained separately by the user. If Winplot is installed on the user's computer, the user may open Winplot using the Run Menu's Winplot selection or the Run Winplot button on the VTASC toolbar. If the unsteady plot files already exist, they will automatically be loaded into Winplot. Otherwise, the user must reselect the Run Menu Winplot selection or toolbar Winplot button to load the plot files into Winplot. For plotting and manipulating data in Winplot, the user is referred to the Winplot user's manual.[13]

5.9.2.3 Tecplot. There is a provision of representing two-dimensional flow field using Tecplot—a commercial flow visualization software. For two-dimensional calculation, the user can activate Tecplot Data in the Output Control window under General Information. This activation allows GFSSP to create a Tecplot data file. After the completion of the run, the user can activate Tecplot in the main window by left clicking the Run Tecplot button in the top line of the canvas.

5.9.3 Display in Flow Circuit

An option exists within VTASC to display results for a particular model element on the VTASC canvas itself. The user can then observe how certain solution parameters change as adjustments are made to the model. This also allows the user to print the model with the results of interest as a hard copy or bitmap for reports or presentations. The user activates this option by first selecting the model element(s) using the selection tool. Then, as discussed in section 5.1.5, the user activates the Display Results/
Properties dialog from the Display menu. Figure 92 shows an example of displaying the results on the VTASC canvas.
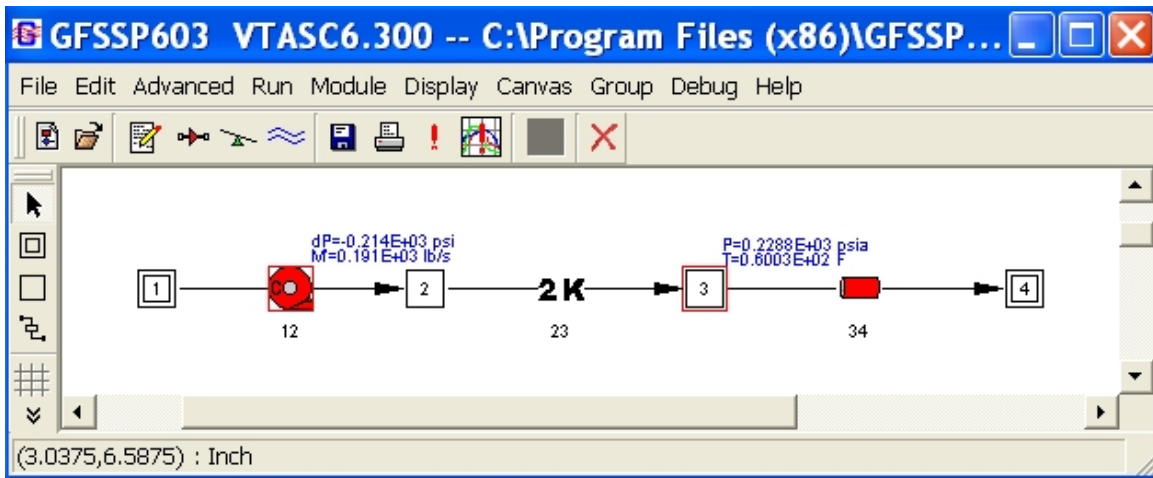
Figure 92. Display results in flow circuit example.

Figure 93 shows the Display Results/Properties dialog. The user can select a maximum of three different parameters to display for each model element. Note that display parameters cannot be varied between like model elements (i.e., the user cannot display the pressure at one fluid node and the temperature at another fluid node). The selected parameters are displayed at all selected elements of that type. Also note that the conjugate heat transfer parameter selections are not active unless the user has activated conjugate heat transfer. Once the user has selected all desired display parameters, the Apply button is used to accept the changes. The OK button closes the Display Results/Properties dialog and returns the user to VTASC. The Cancel button fulfills the same function as the OK button.
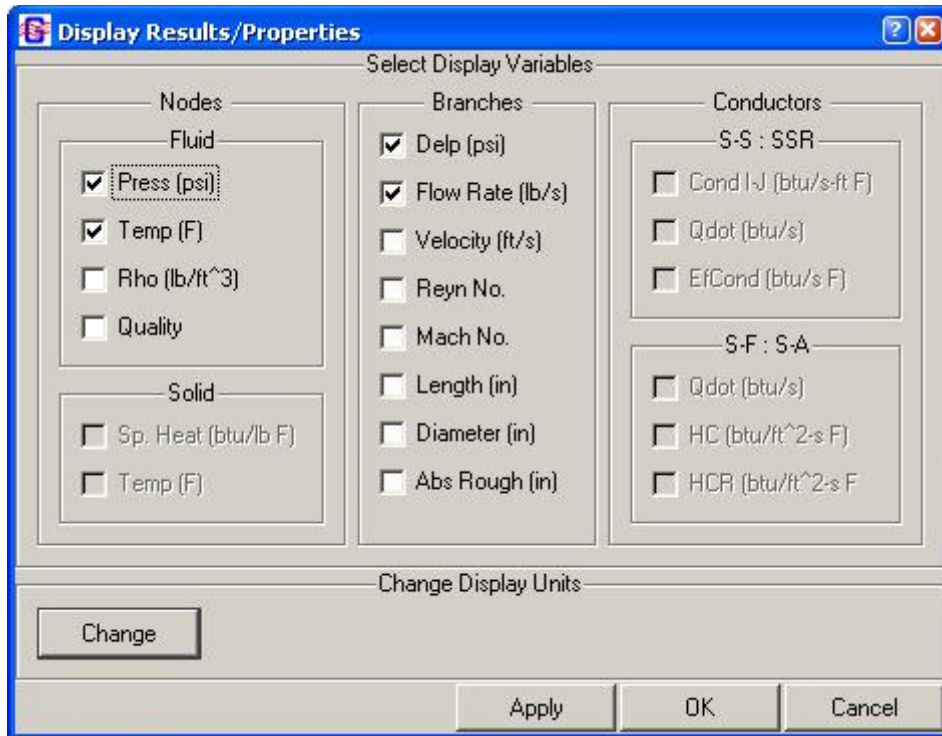
Figure 93. Display Results/Properties dialog.

The display units may be changed through the Display Property Units dialog, which is activated by clicking the Change button. Figure 94 shows the Display Property Units dialog. Alternative display units are available here for certain display parameters. The user toggles between the available units choices for each parameter. The OK button closes the Display Property Units dialog and returns the user to the Display Results/Properties dialog. The Cancel button fulfills the same function as the OK button.
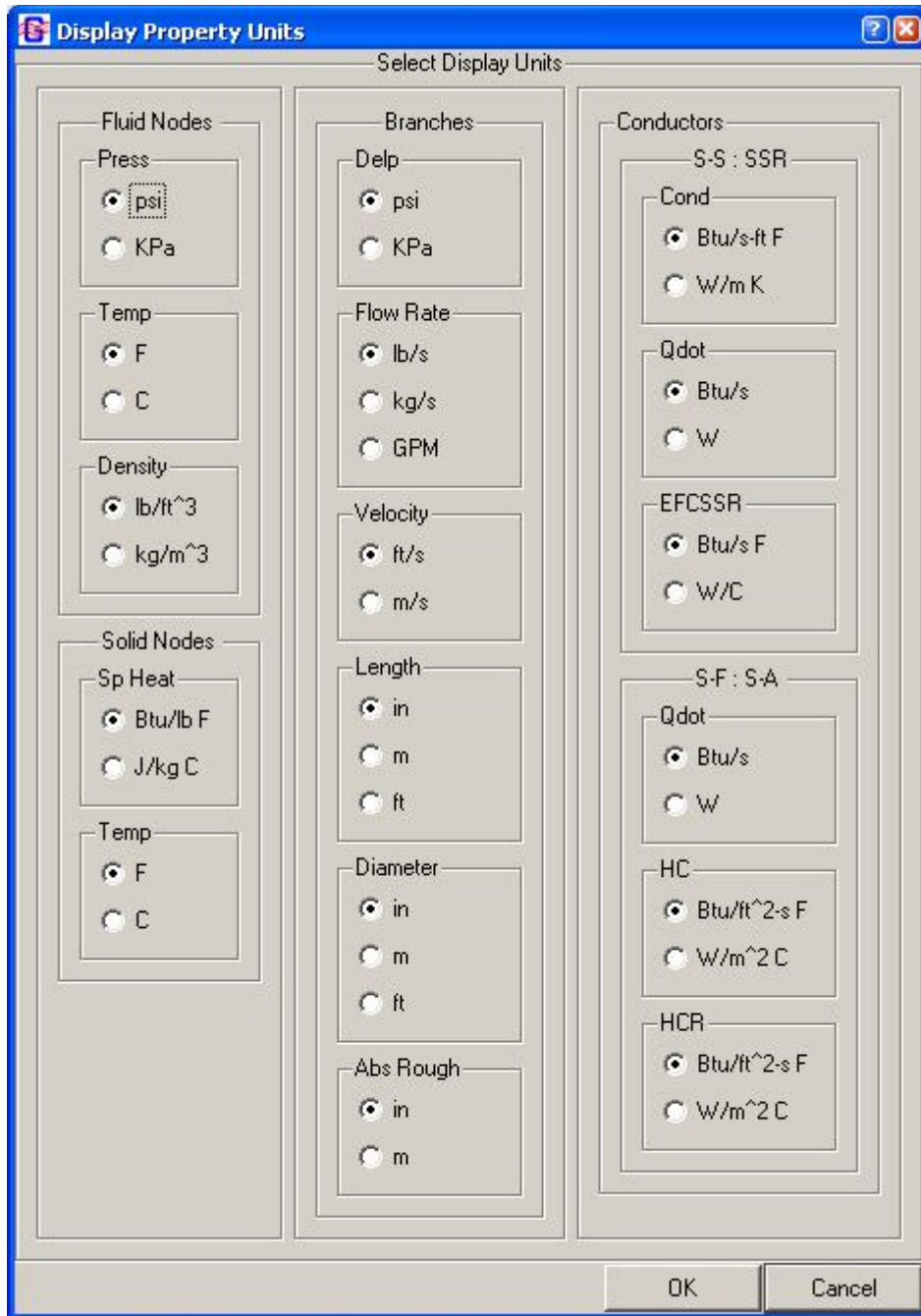
Figure 94. Display Property Units dialog.

6. EXAMPLES

This section demonstrates the major features of the code through 25 example problems, selected to serve two purposes. First, these problems will instruct the user on how to use the various options available in the code to analyze different fluid engineering problems. The other purpose of the examples contained within this section is to verify the code's predictions. This verification was accomplished by comparing the GFSSP solutions with analytical solutions, other numerical solutions, or with test data. The included demonstration problems are as follows:

(1)   Simulation of a flow system consisting of a pump, valve, and pipe line.
(2)   Simulation of a water distribution network.
(3)   Simulation of compressible flow in a converging-diverging nozzle.
(4)   Simulation of the mixing of combustion gases and a cold gas stream.
(5)   Simulation of a flow system involving a heat exchanger.
(6)   Radial flow on a rotating radial disk.
(7)   Flow in a long-bearing squeeze film damper.
(8)   Simulation of the blowdown of a pressurized tank.
(9)   A reciprocating piston-cylinder.
(10) Pressurization of a propellant tank.
(11) Power balancing of a turbopump assembly.
(12) Helium pressurization of LOX and RP-1 propellant tanks.
(13) Steady state and transient conduction through a circular rod, with convection.
(14) Chilldown of a short cryogenic pipeline.
(15) Simulation of fluid transient following sudden valve closure.
(16) Simulation of pressure regulator downstream of a pressurized tank.
(17) Simulation of flow regulator downstream of a pressurized tank.
(18) Subsonic Fanno flow.
(19) Subsonic Rayleigh flow.
(20) Modeling of closed cycle liquid metal (lithium) loop with heat exchanger to heat helium gas.
(21) Internal flow in a turbopump.
(22) Simulation of a fluid network with fixed flow rate option.
(23) Helium-assisted, buoyancy-driven flow in a vertical pipe carrying LOX with ambient heat leak.
(24) Simulation of relief valve in a pressurized tank.
(25) Two-dimensional recirculating flow in a driven cavity.

The selection of the order for these problems is partly determined by their complexities and partly due to the historical developments of new option and capabilities of the code. The first seven problems consider steady state flows and use relatively simple flow networks. Each example demonstrates use of a special option. Example 8 is a two-node model to demonstrate the use of the unsteady option. More complex unsteady flow examples are illustrated in examples 9, 10, and 12.

144

Options for time-dependent geometry and moving boundary are shown in example 9. Pressurization of a cryogenic propellant tank is described in example 10. This example also illustrates the use of User Subroutines to calculate evaporation and mass transfer. The application of the Turbopump and Heat Exchanger options in a typical gas turbine system is illustrated in example 11. The use of a control valve in a pressurization system consisting of both fuel and oxidant tanks is the highlight of example 12. Example 12 is the most complex example, using many options such as mixture and inertia in unsteady flow. Examples 13 and 14 introduce GFSSP's conjugate heat transfer capability. Example 13 is an adaptation of a classical heat transfer problem into GFSSP, while example 14 simulates a common conjugate heat transfer engineering application in cryogenic systems. Example 15 emphasizes GFSSP's ability to predict fluid transient phenomena. Examples 16 and 17 illustrate the use of pressure and flow regulator options which were included in version 6. Two classical problems on compressible flows (Fanno and Rayleigh flows) are described in examples 18 and 19. GFSSP solutions were compared with analytical solutions for both problems. Example 20 illustrates the modeling of a closed loop and the use of a property look-up table. The modeling of internal flow in a turbopump with interpropellant seal for axial thrust calculation is illustrated in example 21. The use of the fixed flow rate option is illustrated in example 22. The use of a new mixture option that allows phase change is demonstrated in example 23. Example 24 demonstrates the use of a pressure relief valve to allow flow out of a pressurized tank that has exceeded a defined pressure. Use of the multidimensional option is demonstrated in example 25 by modeling recirculating flow in a driven cavity. Table 11 describes a matrix of the example problems and their use of various options to model the necessary physical processes. Several technical papers illustrate the application of GFSSP in modeling internal flow in rocket engine turbopump,[28–30] propellant tank pressurization,[31,32] chilldown of cryogenic transfer line,[33,34] propellant tank loading,[35] propellant tank boil-off,[36] fluid transient,[27,37] pressure regulator,[38] and microfluidics application.[39]

Table 11. Use of various options in example problems.

Feature	Example																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Conjugate heat transfer													13	14									23		
Constant property		2					7																		25
Cyclic boundary																				20					
Fixed flow regulator																						22			
Flow regulator																	17								
Gravity	1																						23		
Heat exchanger					5						11									20					
Ideal gas								8								16	17								
Long inertia			3		6							12						18	19						
Fluid mixture				4						10		12											23		
Model import																							23		

145

Table 11. Use of various options in example problems (Continued).

Feature	Example																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Moving boundary							7		9																
Multidi-mensional flow																									25
Noncircular duct							7																		
Phase change														14											
Pressuriza-tion (tank)										10		12													
Pressure regulator																16									
Pressure relief valve																								24	
Pump	1											12													
Turbo pump											11														
Turbo pump-internal flow																					21				
Unsteady								8	9	10		12		14	15	16	17					22	23	24	
User fluid																				20					
User sub-routine										10		12						18	19	20					
Valve O/C															15										
Variable geometry									9																
Water hammer															15										

6.1 Example 1—Simulation of a Flow System Consisting of a Pump, Valve, and Pipe Line

6.1.1 Problem Considered

A problem commonly encountered in fluid engineering is to match a pump's characteristics with the operating system's characteristics. The designer needs to know the flow rate in the system and the power consumed by the pump. The following example problem demonstrates how GFSSP can be used to predict this information.

The system considered for this example is shown in figure 95. It consists of two reservoirs connected by 1,500 ft of 6-in-diameter pipe with a roughness factor (ε/D) of 0.005. The receiving reservoir is located at an elevation that is 150 ft higher than the supply reservoir. The head-flow characteristics of the pump considered in this problem are shown in figure 96. This pump should be used to transport water, at 60 ºF, from the supply reservoir to the receiving reservoir. GFSSP will be used to determine the system flow rate and required pump horsepower.
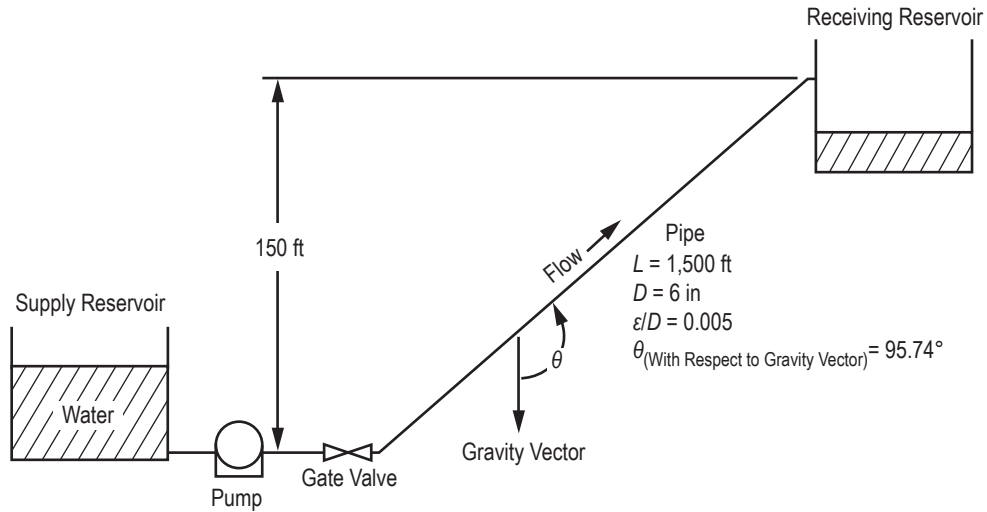
Figure 95. Schematic of pumping system and reservoirs (example 1);
WRT = With Respect To.
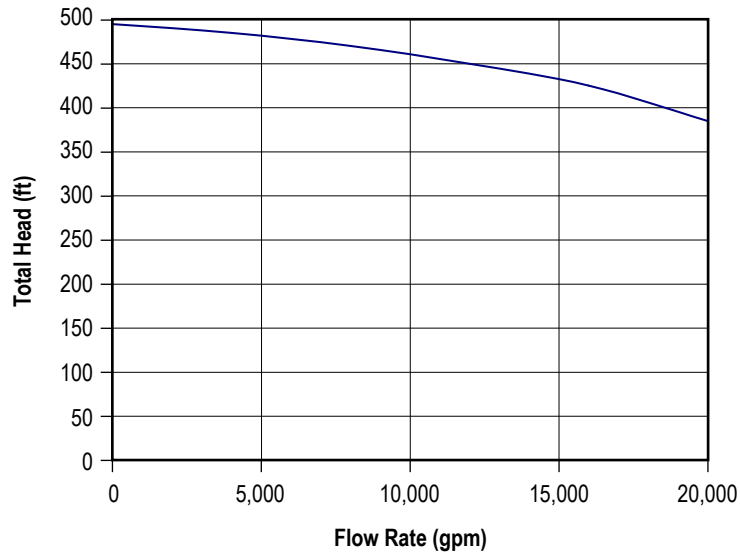
Figure 96. Manufacturer supplied pump head-flow characteristics.

6.1.2 GFSSP Model

The fluid system shown in figure 95 can be simulated with a GFSSP model consisting of four nodes and three branches as shown in figure 97(a). Nodes 1 and 4 are the boundary nodes representing the supply and receiving reservoirs that are both at 14.7 psia and 60 °F. Node 2 is an internal node representing the pump exit and the inlet to the gate valve. Node 3 is an internal node representing the exit from the gate valve and the inlet to the pipeline that connects the valve to the receiving reservoir. Branches 12, 23, and 34 represent the pump, gate valve, and pipeline, respectively. Figure 97(b) shows how this model appears in VTASC.
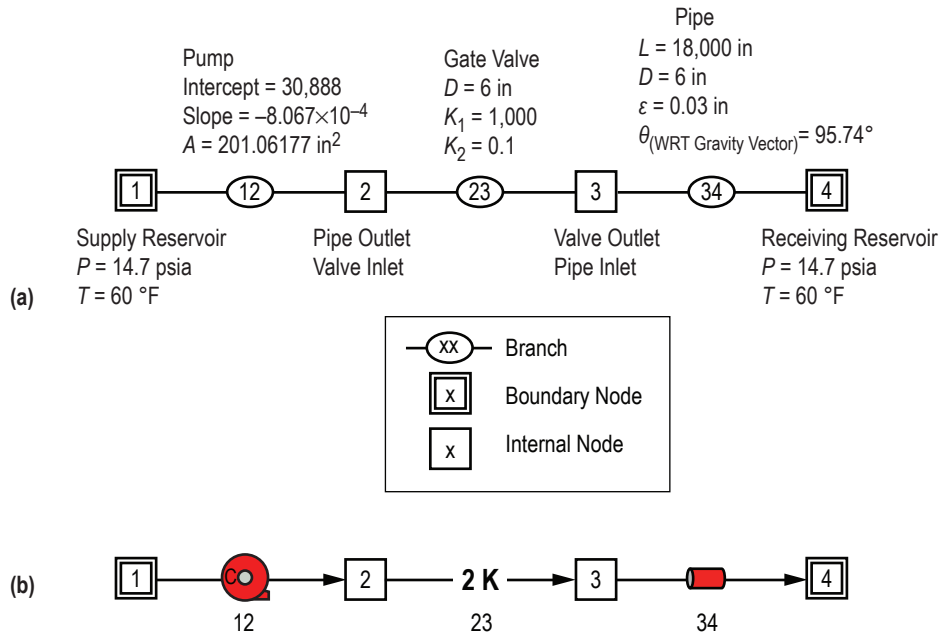
Figure 97. GFSSP model of pumping system and reservoirs: (a) Detailed schematic and (b) VTASC model.

Once the boundary conditions have been established, the next step is to obtain the necessary information required to model the resistances and the momentum source located in the branches.

6.1.2.1 Branch 12 (Pump). Option 14 was selected to represent the pump because this option allows the user to model a pump with a given characteristics curve using either two or three constants. For this problem, two constants, A_0 and B_0, were input as well as the pump area (area is only required for the velocity calculation). These constants represent the slope and the intercept of the $\dot{m}^2$ versus Δp curve. The following procedure is used to obtain these constants:

(1) Construct a table, based on user-selected points, from the pump characteristics curve shown in figure 96 to develop a relationship between $\dot{m}^2$ and Δp. These data are shown in table 12, where $\dot{m} = \rho Q$ and $\Delta p = \rho g/g_c\, H$.

Table 12. Tabulated pump characteristics data.

Q (gpm)	$\dot{m}$ (lb/s)	Head (ft)	Δp (psf)	$\dot{m}^2$ (lb/s)^2
–	–	495	30,888	–
4,000	556.13	485	30,264	3.093×10^5
8,000	1,112.3	470	29,328	1.2372×10^6
12,000	1,668.4	450	28,080	2.784×10^6
16,000	2,224.5	425	26,520	4.9484×10^6
20,000	2,781	385	24,024	7.734×10^6

(2) Plot the Δp and $\dot{m}^2$ data from table 12 as shown in figure 98. Note that the relationship is linear (i.e., $\Delta p = A_0 + B_0\,\dot{m}^2$). Therefore, the pump characteristic curve can be prescribed with two constants, A_0 and B_0, and the optional third constant is not necessary.
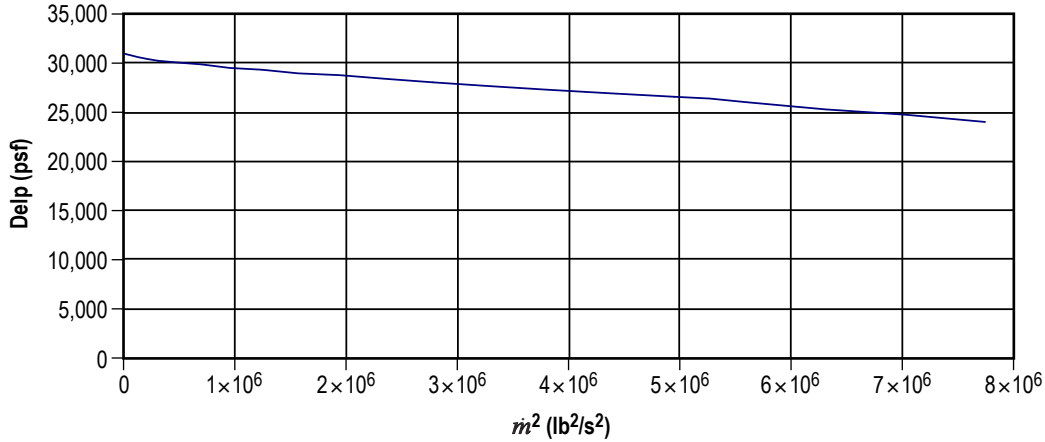
Figure 98. Pump characteristics curve in GFSSP format.

(3) Determine the constants A_0 and B_0 from figure 98:

The intercept (Δp at $\dot{m}^2 = 0$) is $A_0 = 30{,}888$.

The slope (taken about $Q = 10{,}000$ gpm) is $B_0 = \dfrac{28{,}080 - 29{,}328}{\left(2.784\times10^6\right)-\left(1.2372\times10^6\right)} = -\left(8.067\times10^{-4}\right)$.

Since the curve is linear, the slope can be determined at any point without sacrificing accuracy. In VTASC, A_0 is entered in the intercept input box and B_0 is entered in the second order input box.

6.1.2.2 Branch 23 (Gate Valve). Option 13 was used to represent the gate valve. Option 13 requires two constants, K_1 and K_∞ (two-K method), and the internal diameter to model various pipe fittings. The two required constants were obtained from table 7 (see sec. 3) assuming a reduced trim ($\beta = 0.8$) gate valve).

6.1.2.3 Branch 34 (Pipe Line). Option 1 was used to represent the pipe line. This branch resistance option requires the user to supply the length, diameter, roughness factor (ε/D) and the angle between the gravity vector and the pipe line.

6.1.3 Results

The example 1 GFSSP input and output data files (ex1.dat and ex1.out) are included in appendix G (on CD inside back cover).

The example 1 GFSSP model predicts a flow rate of 191 lb/s and the pressure rise across the pump is 214 psi. Interpolation in the table 12 data show that the pump pressure rise for the GFSSP predicted flow rate is 213 psi, which indicates that the model is working as expected. This example demonstrates that GFSSP can accurately predict the operating point of a fluid system consisting of a pump and a pipeline with a valve.

When selecting a pump, the mass flow rate in the attached fluid system is generally unknown. By generating a system characteristic curve and plotting this curve against the pump characteristic curve, the operating point of the system can be determined. Using the example 1 GFSSP model, the system characteristic curve can be generated in the following manner:

(1) Eliminate the pump by setting A_0 and B_0 to zero.
(2) Set boundary pressures P_1 and P_4 to desired values.
(3) Run the model.
(4) Repeat steps (2) and (3) to cover the desired range.

Table 13 shows typical system characteristics generated by performing the parametric study described above. The inlet pressures were arbitrarily selected to cover the expected operating range. The GFSSP predicted mass flow rates are shown in the third column. The flow resistance coefficients for the valve and the pipe are shown in the next two columns. The method of calculating these values is discussed in section 2. The next three columns of table 13 show hand-calculated pressure drops over the gate valve and the pipe and the pressure difference that is associated with the elevation change that exists in the example 1 model. The sum of these three losses is tabulated in the next column. The last column shows the overall pressure drop for the system for the given mass flow rate. A comparison of the values contained in the last two columns shows good agreement.

Table 13. Predicted system characteristics.

P_1 (psia)	P_4 (psia)	$\dot{m}$ (lbm/s)	$K_{f,\,gate}$ lbf-s^2 (lb-ft)^2	$K_{f,\,pipe}$ lbf-s^2 (lb-ft)^2	$\Delta_{p,\,gate}$ (psia)	$\Delta_{p,\,pipe}$ (psia)	$\Delta_{p,\,gravity}$ (psia)	$\Delta_{p,\,cal}$ (psia)	$\Delta_{p,\,pres}$ (psia)
150	14.7	131	0.0019	0.593	0.225	70.67	65	135.89	135.3
200	14.7	171	0.0019	0.592	0.385	120.21	65	185.59	185.3
250	14.7	203	0.0019	0.591	0.544	169.13	65	234.67	235.3
300	14.7	231	0.0019	0.591	0.704	219	65	284.71	285.3

To determine the operating point of the system with a given pump, the system characteristics can be plotted together with the pump characteristics. The system operating point will be determined by the intersection of these two curves. Figure 99 shows the example 1 fluid system characteristics, generated previously, plotted along with the pump characteristics data that are shown in table 12. As seen from this figure, the operating point predicted by this method occurs at a pressure drop of approximately 214 psid and a mass flow rate of 190 lbm/s which compares well with the predicted results from the GFSSP model containing the pump.
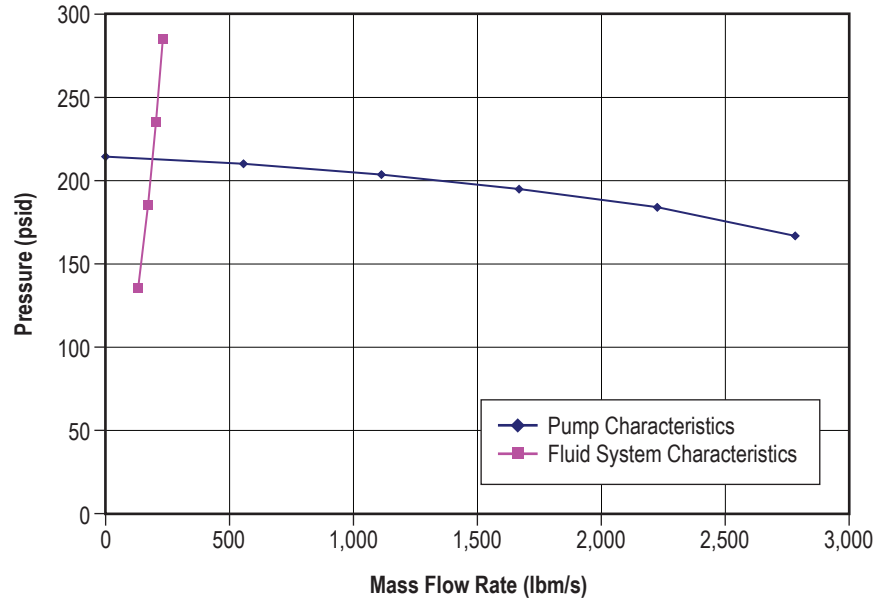
Figure 99. Fluid system operating point.

Finally, the power input to the pump can be calculated from the following relationship:

$$\text{Horse Power} = \frac{\dot{m}}{\rho}\Delta p = \frac{(191\,\text{lbm/s})}{(62.4\,\text{lbm/ft}^3)}\frac{(214\,\text{lbf/in}^2)(144\,\text{in}^2/\text{ft}^2)}{(550\,\text{ft-lbf/hp})} = 171\,\text{hp}\ .$$

6.2 Example 2—Simulation of a Water Distribution Network

6.2.1 Problem Considered

In example 1 a single line pipe flow problem commonly encountered by pipeline designers is analyzed. In this example, an example associated with multipath systems, commonly known as flow networks, is considered. In general, water supply systems are considered as flow networks, since nearly all such systems consist of many interconnecting pipes. A 10-pipe (commercial steel) distribution system is shown in figure 100. Water at 50 psia enters the circuit at boundary node 1. Water is removed from the circuit at boundary nodes 3, 4, and 9 where pressures are also known. Use GFSSP to determine pressures at all of the remaining nodes and the flow rates in all of the pipes. The length, diameter, and roughness factors for each pipe are given in table 14. Note that pipes are designated as branches in the subsequent discussions.
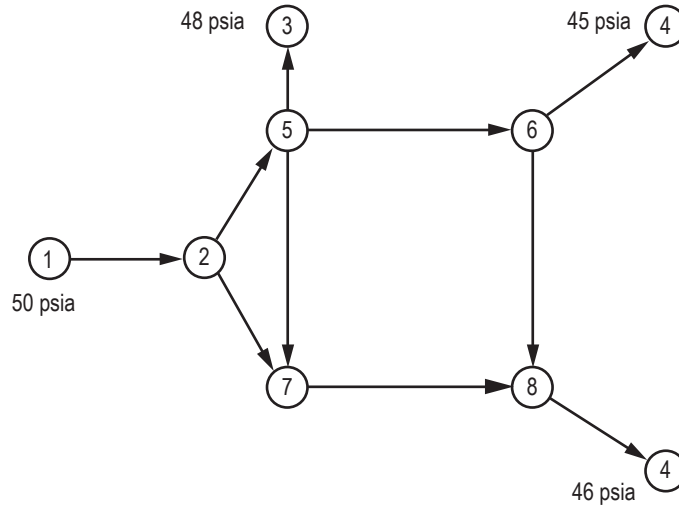
Figure 100. Water distribution network schematic (example 2).

Table 14. Water distribution network branch data.

Branch	Length (in)	Diameter (in)	Roughness Factor
12	120	6	0.0018
25	2,400	6	0.0018
27	2,400	5	0.0018
57	1,440	4	0.0018
53	120	5	0.0018
56	2,400	4	0.0018
64	120	4	0.0018
68	1,440	4	0.0018
78	2,400	4	0.0018
89	120	5	0.0018

6.2.2 GFSSP Model

The system shown in figure 100 is modeled by GFSSP in VTASC using 9 nodes and 10 branches as shown in figure 101. The fluid was assumed incompressible and therefore a constant density (DENCON = .TRUE.) option was used. Nodes 1, 3, 4, and 9 are boundary nodes where the pressures are prescribed. Node 1 represents the inlet boundary node. Nodes 3, 4, and 9 are outlet boundary nodes. All of the remaining nodes (2, 5, 6, 8, and 7) are internal nodes where the pressures are calculated. All of the branches in this circuit simulate pipes. Therefore, each branch uses branch resistance option 1. The length, diameter, and roughness factors of all branches are given in table 14.
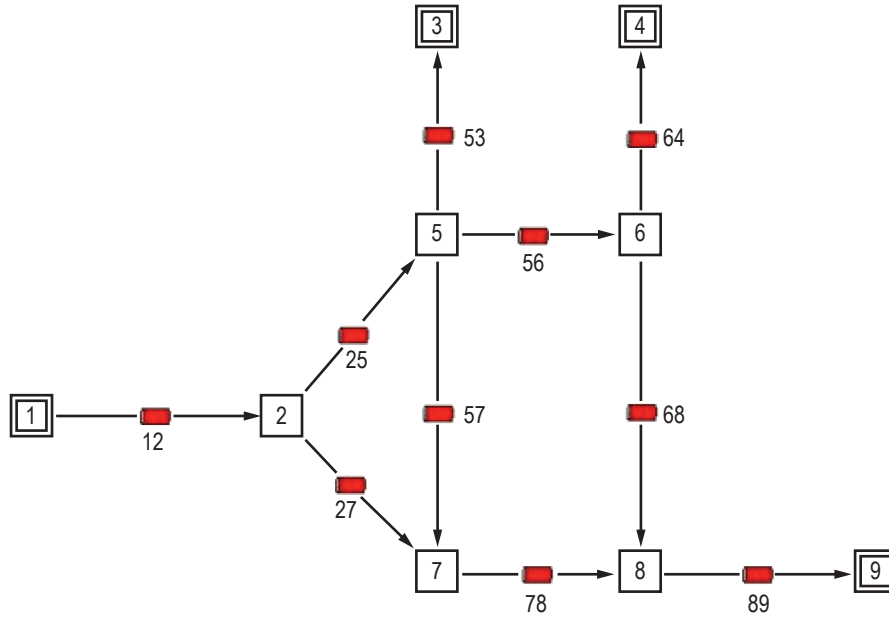
Figure 101. GFSSP model of the water distribution network.

6.2.3 Results

The input and output data files (ex2.dat and ex2.out) are shown in appendix H (on CD inside back cover). The GFSSP predicted results are shown in tables 15 and 16. Table 15 lists the predicted pressures at the internal nodes in psia and feet of water. Table 16 lists the predicted flow rates in lbm/s and ft^3/s. Table 16 also provides a comparison between the GFSSP predicted results and values predicted by the Hardy Cross[2,40] method.

Table 15. GFSSP predicted pressure distribution at the internal nodes.

P_2		P_5		P_6		P_7		P_8	
psia	ft	psia	ft	psia	ft	psia	ft	psia	ft
49.8	114.92	48.11	111.02	45.34	104.63	48.35	111.58	46.01	106.18

Table 16. GFSSP and Hardy Cross method predicted branch flow rates.

Flow Rate	GFSSP		Hardy Cross	
	lb/s	ft³/s	lb/s	ft³/s
$\dot{m}_{12}$	100.16	1.605	100.16*	1.605*
$\dot{m}_{25}$	63.1	1.011	63.59	1.019
$\dot{m}_{27}$	37	0.593	36.58	0.5862
$\dot{m}_{53}$	44.43	0.7115	44.43*	0.7115*
$\dot{m}_{56}$	29.1	0.466	29.11	0.4665
$\dot{m}_{57}$	−10.4	−0.167	−9.93	−0.1592
$\dot{m}_{64}$	47.07	0.7548	47.07*	0.7548*
$\dot{m}_{68}$	−18	−0.288	−17.99	−0.2883
$\dot{m}_{78}$	26.7	0.428	26.64	0.427
$\dot{m}_{89}$	8.66	0.1387	8.66*	0.1387*

* Boundary flow rates are prescribed from GFSSP predictions.

Figure 102 shows a comparison between GFSSP and Hardy Cross predicted flow rates. The comparison appears reasonable considering the fact that the Hardy Cross method assumes a constant friction factor in the branch while the GFSSP computes the friction factor for all branches. Therefore, as the flow rates change, the friction factor also changes.
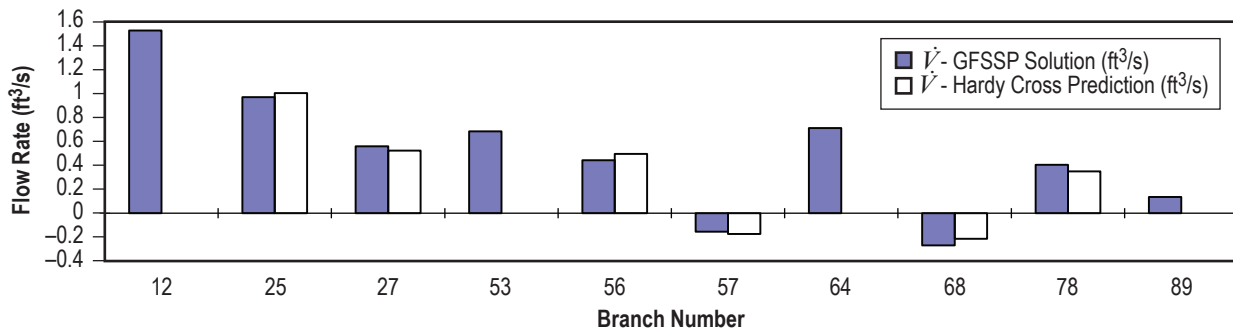
Figure 102. Flow rate comparison between GFSSP and Hardy Cross method predictions.

6.3 Example 3—Simulation of Compressible Flow in a Converging-Diverging Nozzle

6.3.1 Problem Considered

In the previous examples, incompressible flows in fluid systems were considered. In this example, compressible flow in a converging-diverging nozzle, demonstrating GFSSP's capability to handle compressibility, will be considered. One of the characteristics of the compressible flow in a duct is that the flow rate becomes independent of exit pressure after reaching a threshold flow rate. This threshold value is known as the choked flow rate and it is a function of inlet pressure

and temperature. Flow in a confined duct becomes choked when the flow velocity equals the local velocity of sound. The purpose of this example is to investigate how accurately GFSSP can predict the choked flow rate in a converging-diverging nozzle.

The converging-diverging nozzle considered for this example is shown in figure 103. The nozzle is 6.3 in long with a 0.492-in-diameter throat. The inlet diameter of the nozzle is 0.6758 in and throat of the nozzle is located 0.158 in downstream of the inlet. The fluid considered was steam at 150 psia and 1,000 °F. The nozzle backpressure was varied from 134 to 45 psia. GFSSP is desired to predict the flow rate and the pressure distribution for different exit pressures. The predicted flow rate will also be compared with the isentropic solution.
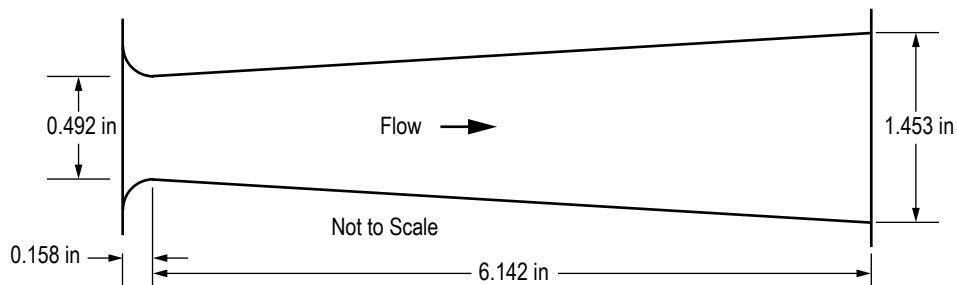
Figure 103. Converging-diverging steam nozzle schematic.

6.3.2 GFSSP Model

The fluid system shown in figure 103 can be simulated with a GFSSP model consisting of 17 nodes and 16 branches as shown in figure 104(a). Nodes 1 and 17 are the boundary nodes representing the inlet and outlet of the nozzle. All of the remaining nodes are internal nodes connected in series. GFSSP can be used to construct an isentropic model by selecting branch resistance option 2 (flow through a restriction), using a flow coefficient, C_L, set equal to zero and by setting the logical flag INERTIA = .TRUE. This option eliminates friction from the momentum equation, which represents a balance between the inertia and pressure force with the inclusion of the inertia term. Each branch assumes a constant flow area that was determined from the nozzle geometry at the mid point of the branch location. The branch information is listed in table 17. Figure 104(b) shows how this model appears in VTASC.

155
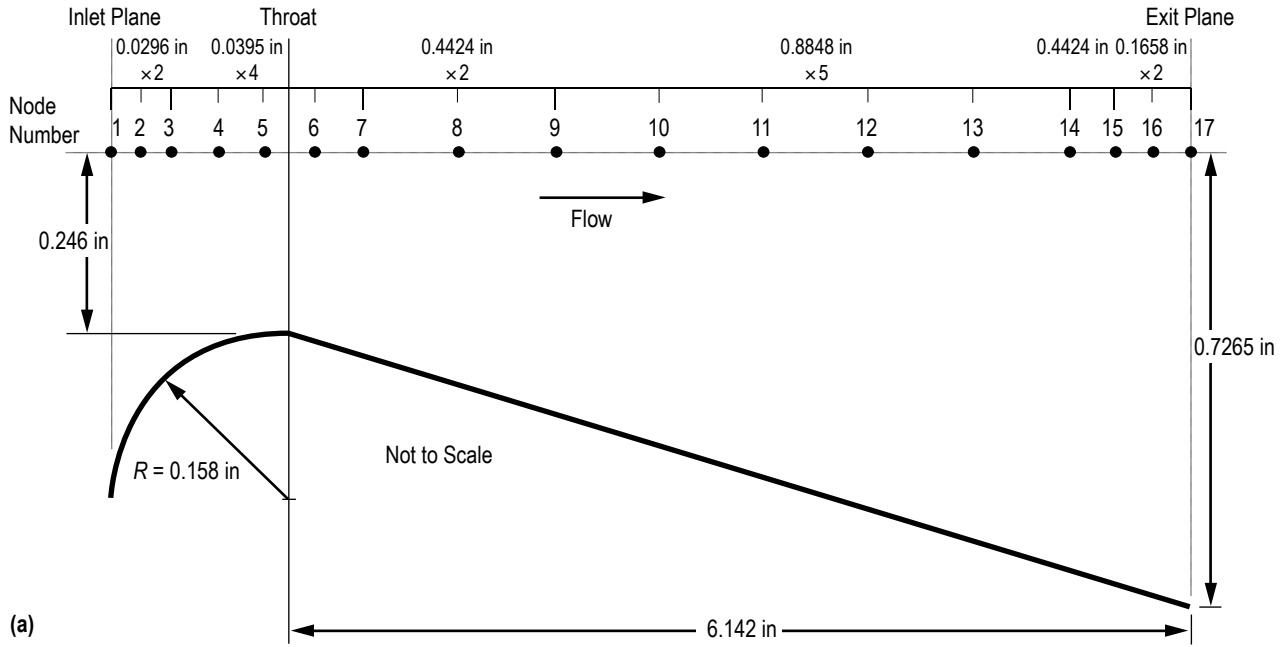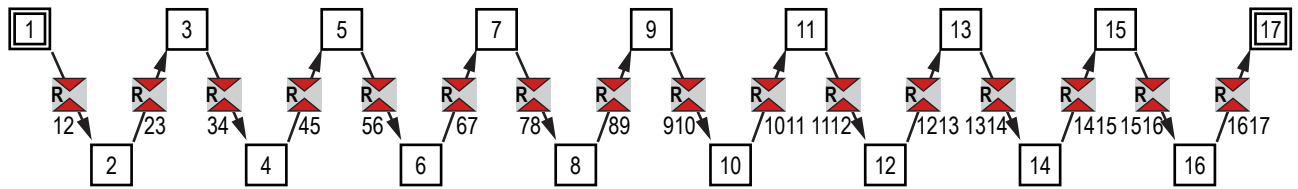
Figure 104. Converging-diverging steam nozzle model: (a) Detailed schematic and (b) VTASC model.

Table 17. Converging-diverging nozzle branch information.

Branches	Option	Area (in^2)
12	2	0.3587
23	2	0.2717
34	2	0.2243
45	2	0.2083
56	2	0.1901
67	2	0.1949
78	2	0.2255
89	2	0.2875
910	2	0.3948
1,011	2	0.564
1,112	2	0.7633
1,213	2	0.9927
1,314	2	1.252
1,415	2	1.4668
1,516	2	1.5703
1,617	2	1.6286

It may be mentioned here that the temperature at the outlet boundary node has no influ-

ence on the solution. However, the code must be supplied with a value to satisfy the input requirements for a boundary node. In this example the temperature was specified arbitrarily to 1,000 °F at node 17. The boundary conditions are specified in table 18.

Table 18. Converging-diverging nozzle boundary conditions.

P_1 (psia)	T_1 (°F)	P_{17} (psia)	T_{17} (°F)
150	1,000	134	1,000
150	1,000	100	1,000
150	1,000	60	1,000
150	1,000	50	1,000
150	1,000	45	1,000

6.3.3 Results

The outlet boundary node pressures were varied to include 134, 100, 60, 50, and 45 psia. The input and output files (ex3.dat and ex3.out) from the example case using an exit pressure of 60 psia are included in appendix I (on CD inside back cover). The predicted pressure distributions for five test cases are shown in figure 105. Figure 106 shows the predicted temperature distributions for the same cases.
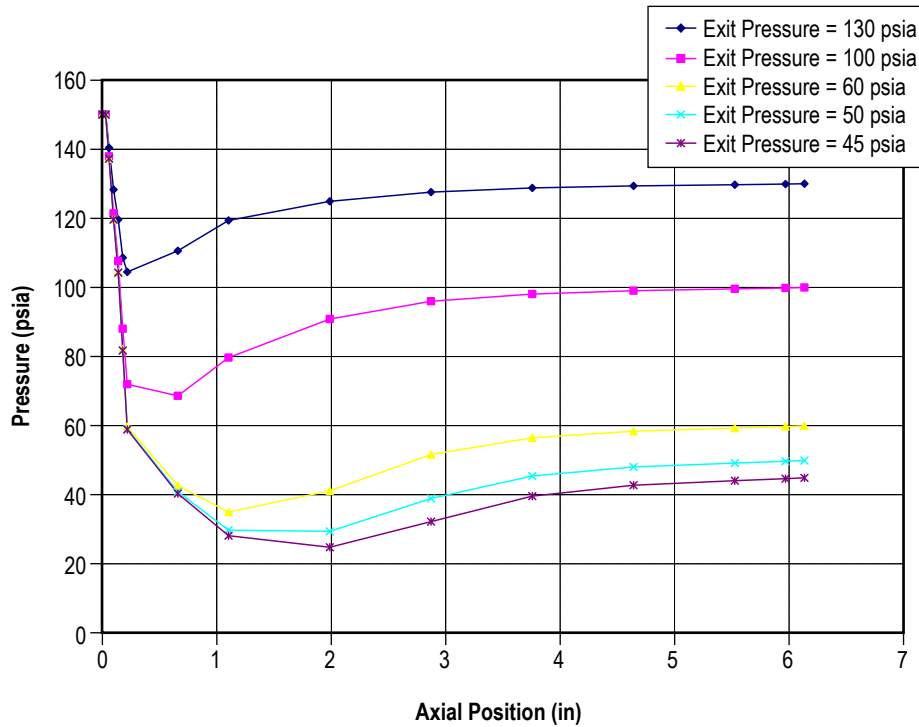
Figure 105. Predicted pressures for the isentropic steam nozzle.
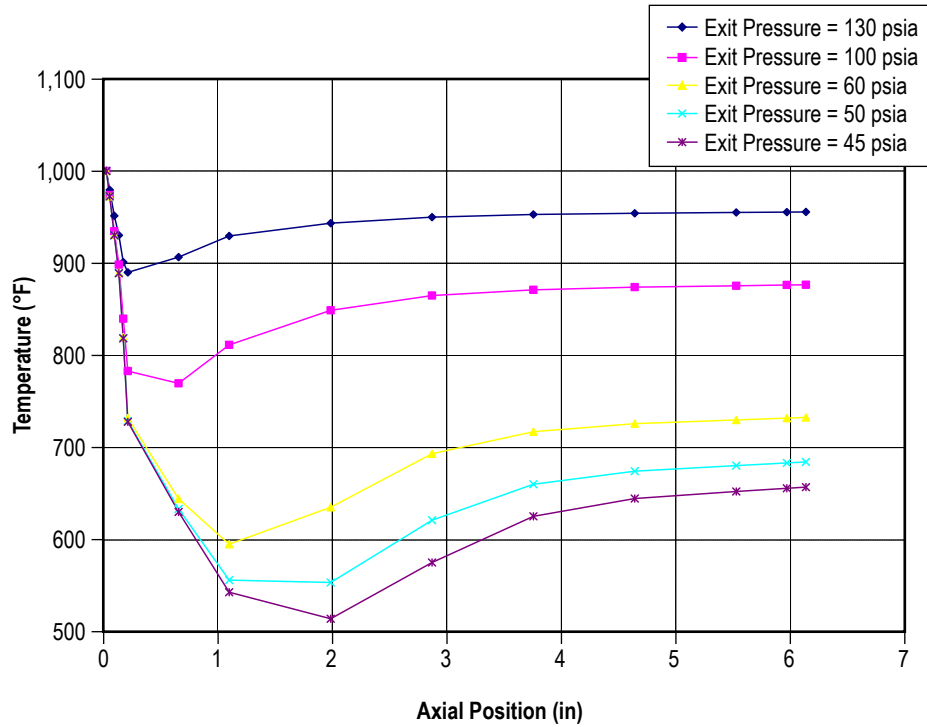
Figure 106. Predicted temperatures for the isentropic steam nozzle.

Table 19 lists the model predicted mass flow rates with varying exit pressures. As expected, the mass flow rate increased as the exit pressure was decreased until the pressure ratio decreased below the critical pressure ratio. At this point and below, the mass flow rate remained constant due to choking of the flow at the nozzle throat.

Table 19. Predicted mass flow rate with varying exit pressure.

P_{exit} (psia)	$\dot{m}$ (lbm/s)
134	0.279
100	0.329
60	0.336
50	0.337
45	0.337

The isentropic flow rate was calculated from equation (72). This equation assumes that the inlet pressure is a stagnation pressure. GFSSP's formulation assumes that the prescribed boundary conditions are taken at a static condition. The nozzle inlet velocity head component must be added to the GFSSP static inlet boundary pressure to obtain the correct nozzle inlet stagnation pressure to use in equation (72) as shown in equation (73):

$$\dot{m} = A_{\text{throat}} P_{\text{inlet}} \sqrt{\frac{g_c \, \gamma}{R \, T_{\text{inlet}}} \left(\frac{2}{\gamma + 1}\right)^{\left(\frac{\gamma + 1}{\gamma - 1}\right)}} \tag{72}$$

and

$$P_{\text{inlet}} = P_{\text{static}} \left(1 + \left(\frac{\gamma - 1}{2}\right) M^2\right)^{\frac{\gamma}{\gamma - 1}} . \tag{73}$$

A value of the specific heat ratio at the nozzle inlet boundary node was obtained from GASP, and the inlet Mach Number from the first branch connected to the inlet node in the model output data file contained in appendix I. Substituting these values into equation (73), gives the following total pressure at the nozzle inlet:

$$P_{\text{inlet}} = (150 \text{ psia}) \left(1 + \left(\frac{1.2809 - 1}{2}\right)(0.342)^2\right)^{\frac{1.2809}{1.2809 - 1}} = 161.6 \text{ psia} . \tag{74}$$

Substituting the calculated total inlet pressure from equation (74) into equation (72) and solving for the choked mass flow rate gives a calculated isentropic choked mass flow rate of 0.327 lbm/s as shown:

$$\dot{m} = \left(0.19012 \text{ in}^2\right)\left(161.6 \frac{\text{lbf}}{\text{in}^2}\right) \sqrt{\frac{32.174 \frac{\text{lbm-ft}}{\text{lbf-s}^2} (1.281)}{85.83 \frac{\text{lbf-ft}}{\text{lbm-}°\text{R}} 1,460 \text{ }°\text{R}} \left(\frac{2}{1.281 + 1}\right)^{\frac{2.281}{0.281}}} = 0.327 \frac{\text{lbm}}{\text{s}} . \tag{75}$$

As the reader can see by comparing the results shown in table 19 and equation (75), there is good agreement (approximately a 3% difference) between the GFSSP predicted choked flow rate and the calculated isentropic choked flow rate. The prediction from first law based formulation has also been compared with the second law formulation. In the first law formulation, enthalpy was used as the dependent variable and equation (3a) was used instead of equation (3b). Unlike the second law formulation where density is computed from pressure and entropy, the first law formulation calculates density from pressure and enthalpy. The comparison of choked mass flow rate for both formulations with the isentropic solution is shown in table 20. It may be noted that the entropy based formulation more accurately predicts the mass flow rate than the enthalpy based formulation.

Table 20. Comparison of choked mass flow rates.

Parameter	GFSSP Version 2.6 (6th Workshop Release)	GFSSP Version 2.0 (5th Workshop Release)
GFSSP predicted isentropic mass flow rate	0.337 lbm/s	0.308 lbm/s
Calculated stagnation inlet pressure	161.6 psia	159.71 psia
Calculated isentropic mass flow rate	0.337 lbm/s	0.323 lbm/s
Percent difference between GFSSP predicted and calculated isentropic mass flow rates	3.06%	−4.64%

To validate GFSSP's ability to predict temperatures, the GFSSP predicted temperature at the nozzle throat was also compared with a hand-calculated temperature using equation (76). This equation assumes an isentropic process. In equation (76), the pressure P_1 was assumed to be the total pressure at the nozzle inlet that was calculated from equation (74) and the throat pressure was taken from the GFSSP output:

$$T_2 = T_1 \left(\frac{P_2}{P_1}\right)^{\frac{\gamma-1}{\gamma}} = (1{,}460 \text{ °R})\left(\frac{82.13 \text{ psia}}{161.6 \text{ psia}}\right)^{\frac{1.2809-1}{1.2809}} = 1{,}258.6 \text{ °R} = 798.6 \text{ °F} . \tag{76}$$

The GFSSP predicted nozzle throat temperature is 819.9 °F. This temperature compares well, within 3%, with the value calculated from equation (76).

The isentropic steam nozzle model represents a reversible process in which no entropy is generated. In an actual steam nozzle there will be frictional losses that result in an increase in entropy. To demonstrate GFSSP's ability to predict the irreversibility (entropy generation) of a process, the isentropic steam nozzle model was modified to allow for frictional losses. The branch resistance option for each model branch was changed from option 2 with a $C_L = 0$, to option 1 using the branch lengths and diameters shown in figure 104 and assuming an absolute roughness of 0.01 in.

Figure 107 shows a temperature/entropy comparison between the GFSSP isentropic steam nozzle model and the GFSSP steam nozzle model with frictional losses. As one can see in the referenced figure, the isentropic model predicts no change in entropy through the nozzle while the irreversible process predicts an increase in entropy of 0.019 Btu/(lbm °R).
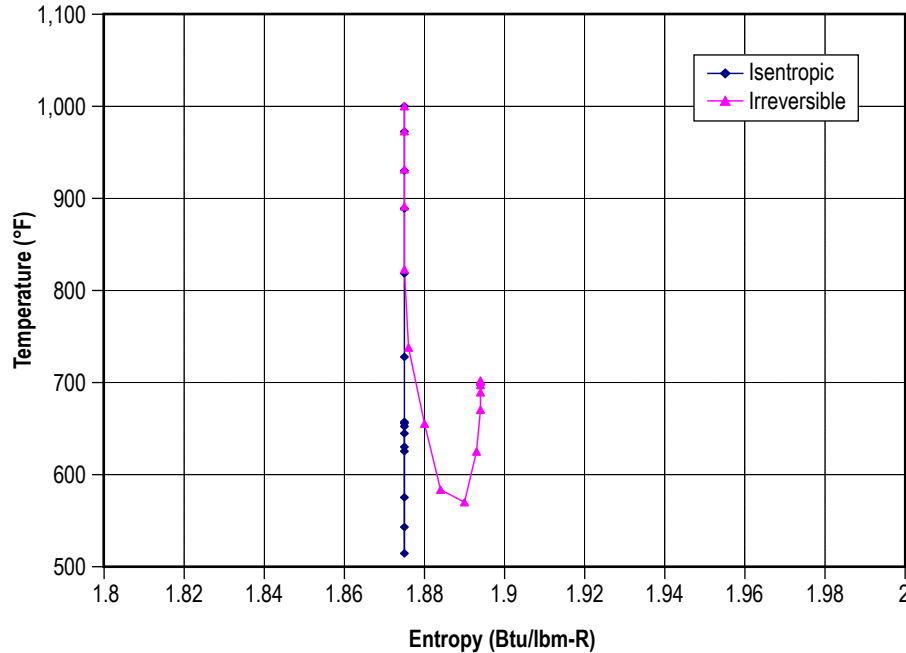
Figure 107. Temperature/entropy plot comparing the isentropic steam nozzle with an irreversible process.

6.4 Example 4—Simulation of the Mixing of Combustion Gases and a Cold Gas Stream

6.4.1 Problem Considered

In the previous examples, the fluid systems that were considered employed a single fluid. In this example, simulation of multiple fluids in a mixing process are considered. The MIXTURE logical option in the code to simulate the mixing of combustion gases and a cold gas stream by utilizing the flow system shown is demonstrated in figure 108. A mixture of hot combustion products, consisting of water vapor and oxygen, is mixed with cooler oxygen gas. The mixture temperature and composition are required to be calculated.

161

500 psia
1,500 °F
$X_{H_2O} = 0.9$
$X_{O_2} = 0.1$
$C_{pH_2O} = 0.5696$ (Btu/(lbm °F))
$C_{pO_2} = 1.2512$ (Btu/(lbm °F))

$\dot{m}_{13}$

$\dot{m}_{34}$

14.7 psia
80 °F

$\dot{m}_{23}$

500 psia
80 °F
$X_{O_2} = 1$
$C_{pO_2} = 0.2324$ (Btu/(lbm °F))
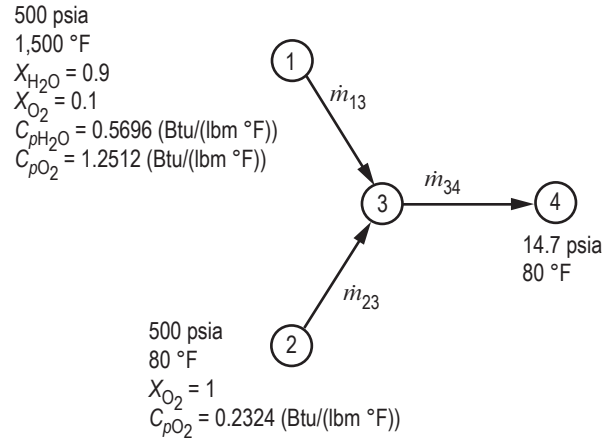
Figure 108. Mixing problem schematic (example 4).

A mixture consisting of 90% water vapor and 10% oxygen (by mass) at 500 psia and 1,500 °F mixes with pure oxygen at 500 psia and 80 °F. GFSSP is desired to predict the flow rate, mixture temperature, and composition of the mixture. A hand calculation of the mixture temperature and the composition of the mixture will also be performed to verify GFSSP's predictions.

6.4.2 GFSSP Model

The mixing chamber shown in figure 108 can be simulated with a GFSSP model consisting of four nodes and three branches as shown in figure 109. Nodes 1, 2, and 4 are the boundary nodes representing the inlet and outlet of the mixing chamber and node 3 is the internal node representing the mixing chamber. Branches 13 and 23 are represented by option 2 using a flow coefficient of 0.6 and area of 1 in². Branch 34 also uses a flow coefficient of 0.6 and area of 1 in², but it is modeled using option 22. The reason option 22 is used for branch 34 is to account for the possibility of choked flow in the branch.
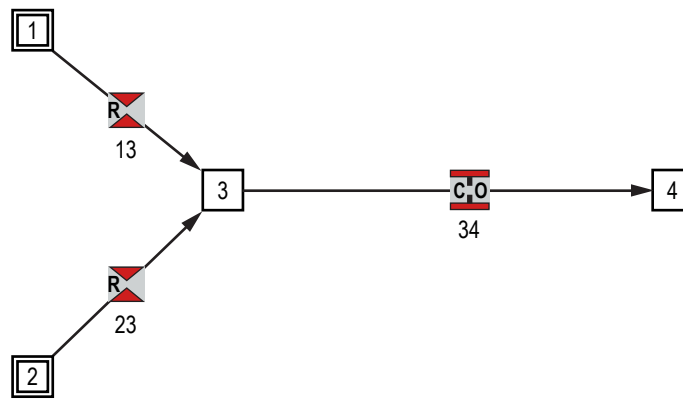
Figure 109. GFSSP model of mixing problem.

162

The input and output files of this example are included in appendix J (on CD inside back cover) as ex4.dat and ex4.out. The predicted flow rates in branches 13, 23, and 34 are 1.16, 3.10, and 4.26 lbm/s, respectively. The predicted temperature at the outlet of the mixing chamber, node 3, is 689 ºF and the composition is 24.47% water vapor and 75.53% oxygen. The mixture will not vary between nodes 3 and 4.

Now, verification of the predicted results will be done by performing hand calculations of the mixing process. The temperature of the mixture can be calculated from the energy conservation equation written for the mixing chamber. The energy conservation equation for node 3 can be written as:

$$x_{H_2O} \dot{m}_{13} C_{p,H_2O} T_1 + x_{O_2} \dot{m}_{13} C_{p,O_2} T_1 + \dot{m}_{23} C_{p,O_2} T_2 = \dot{m}_{34} C_{p,\text{mix}} T_3 \; . \tag{77}$$

The above equation can be rearranged to find T_3:

$$T_3 = \frac{x_{H_2O} \dot{m}_{13} C_{p,H_2O} T_1 + x_{O_2} \dot{m}_{13} C_{p,O_2} T_1 + \dot{m}_{23} C_{p,O_2} T_2}{\dot{m}_{34} C_{p,\text{mix}}}$$

$$= \frac{(0.9)(1.16 \text{ lb/s})(0.5696 \text{ Btu/lb-ºR})(1{,}960 \text{ ºR}) + (0.1)(1.16 \text{ lb/s})(1.2512 \text{ Btu/lb-ºR})(1{,}960 \text{ ºR})}{(4.26 \text{ lb/s})(0.3757 \text{ Btu/lb-ºR})}$$

$$+ \frac{(3.1 \text{ lb/s})(0.2324 \text{ Btu/lb-ºR})(540 \text{ ºR})}{(4.26 \text{ lb/s})(0.3757 \text{ Btu/lb-ºR})} = 1{,}149.1 \text{ ºR or } 689.5 \text{ ºF}. \tag{78}$$

Equation (78) calculates the temperature to be 689.5 ºF, which compares well with the GFSSP prediction of 688.6 ºF.

The mass concentration of the specie can be calculated from the specie conservation equation for node 3. The concentration of water vapor and oxygen, respectively, can be expressed as:

$$X_{3,H_2O} = \frac{X_{1,H_2O} \dot{m}_{13}}{\dot{m}_{34}} = \frac{(0.9)(1.16 \text{ lb/s})}{4.26 \text{ lb/s}} = 0.2451 \tag{79}$$

and

$$X_{3,O_2} = \frac{X_{1,O_2} \dot{m}_{13} + \dot{m}_{23}}{\dot{m}_{34}} = \frac{(0.1)\left(1.16 \text{ lb/s}\right) + 3.1 \text{ lb/s}}{4.26 \text{ lb/s}} = 0.7549 \; . \tag{80}$$

The concentration of water vapor and oxygen from the above equations are 0.2451 and 0.7549, respectively, which also compares well with the GFSSP predictions.

6.5 Example 5—Simulation of a Flow System Involving a Heat Exchanger

6.5.1 Problem Considered

In dealing with fluid system analysis, engineers often encounter systems that contain a heat exchanger. It is important that the thermal behavior of a heat exchanger is correctly accounted for in any system simulation. Otherwise, temperature discrepancies in the fluid property calculations will result in inaccurate system characteristics being predicted. The following example demonstrates GFSSP's ability to accurately predict fluid temperatures in a heat exchanger system using effectiveness calculations.

GFSSP has the ability to calculate temperatures downstream of a heat exchanger for three different cases. For the first case, a known heat exchanger effectiveness is used by GFSSP to calculate the flow temperatures downstream from the heat exchanger. The second case involves requiring GFSSP to calculate the effectiveness of the heat exchanger using the counter flow heat exchanger equations. That calculated effectiveness is then used to calculate the heat exchanger downstream temperatures. For the third case, the heat exchanger effectiveness is calculated using the parallel flow heat exchanger equations and the heat exchanger downstream temperatures are calculated from that effectiveness.

A simple counter flow heat exchanger system configuration, as shown in figure 110, was chosen for this example. As shown in figure 110, counter flow occurs when the hot branch of the heat exchanger has flow that is propagating in a direction opposite to the cold branch. This counter flow heat exchanger configuration consists of hot water, at 50 psi and 100 °F, flowing through 10 in of 0.25-in inner diameter pipe, through a 10-in-long heat exchanger and out through another 10-in-long section of 0.25-in inner diameter pipe at 25 psi. Also, cold water, at 50 psi and 60 °F flows through a 10-in section of 0.5-in inner diameter pipe, through the heat exchanger, and out through another 10-in section of 0.5-in inner diameter pipe at 25 psi. All of the pipes are assumed to have an absolute roughness of zero. At the conditions described above, $C_p = 0.9978$ Btu/lbm-°R is calculated for the hot water and $C_p = 1.0014$ Btu/lbm-°R is calculated for the cold water.
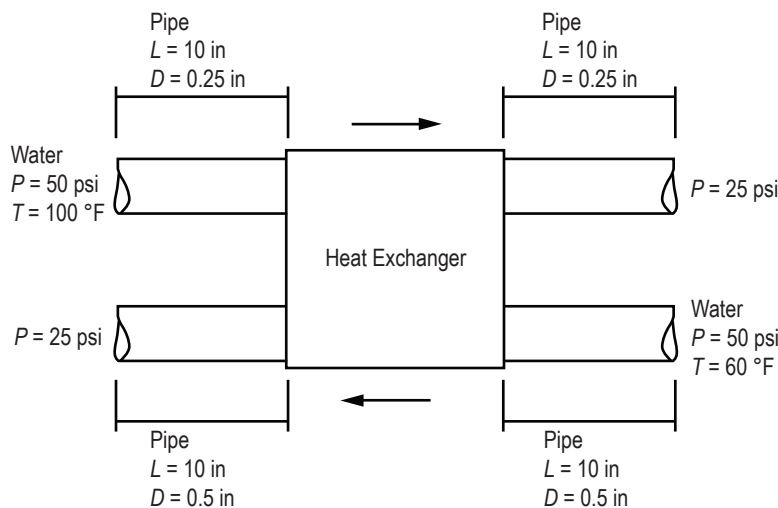
Figure 110. Flow system schematic of a heat exchanger (example 5).

6.5.2 GFSSP Model

A GFSSP model consisting of eight nodes and six branches can represent the counter flow heat exchanger system shown in figure 110. This model is shown in figure 111(a). Nodes 1, 4, 5, and 8 are boundary nodes. Nodes 1 and 5, the inlet boundary nodes for the hot and cold flow, respectively, both have a pressure of 50 psi. In addition, the boundary temperature at node 1 is 100 °F while the boundary temperature at node 5 is 60 °F. Nodes 4 and 8, which are the downstream boundary nodes for the hot and cold flow, both have a boundary pressure of 25 psi. Downstream boundary temperatures are not used in GFSSP calculations so 'dummy' temperature values of 80 and 70 °F are used for the hot and cold flow downstream boundary nodes. Nodes 2 and 6 are internal nodes that represent the entrances to the heat exchanger for the hot and cold flow, respectively. In the same manner, nodes 3 and 7 are internal nodes that represent the hot and cold flow heat exchanger exits. Branches 12, 34, 56, and 78 represent the pipes leading into and out of the heat exchanger for both the hot and cold flows. Finally, branches 23 and 67 represent the hot and cold sides of the heat exchanger. Figure 111(b) shows how this model appears in VTASC.
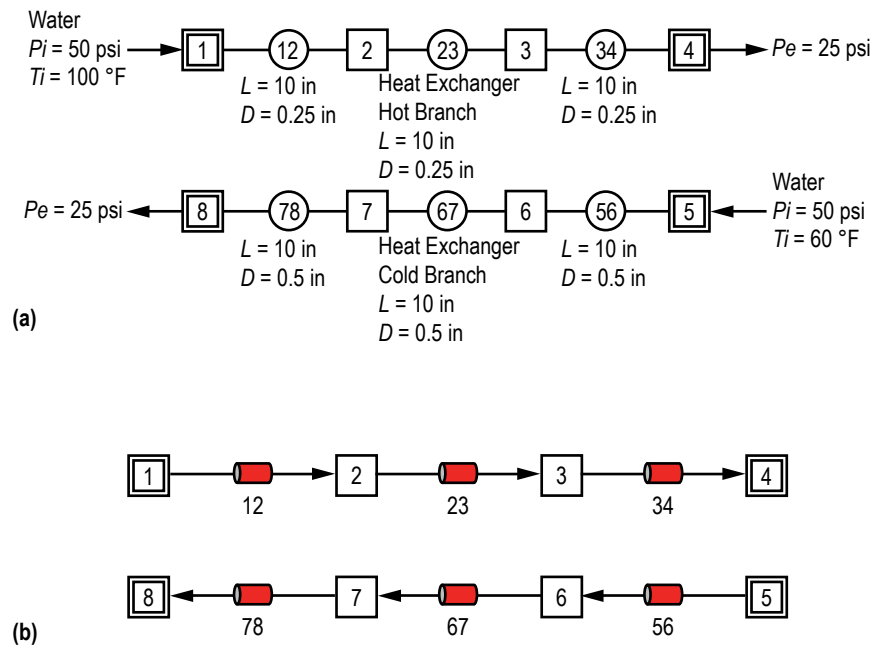
Figure 111. GFSSP model of the heat exchanger: (a) Detailed schematic and (b) VTASC model.

6.5.2.1 Branches 12, 34, 56, and 78 (Pipe Lines). Option 1 was used to represent each of the pipe sections in the heat exchanger model. The user is required to provide the length, inner diameter, and relative roughness factor (ε/D) for this branch resistance option.

6.5.2.2 Branches 23 and 67 (Heat Exchanger). Option 1 was also used to simulate the two heat exchanger branches. In addition to providing the length, inner diameter, and ε/D for the two

branches, the user must designate additional information in the Heat Exchanger dialog window shown in figure 112. First, the user must add a heat exchanger using the Add button. Next, the user must input which branch represents the hot flow side of the exchanger and which branch represents the cold flow side of the exchanger. Also, the user may either designate an effectiveness between zero and 1 or enter an effectiveness >1 and designate the type of heat exchanger in the system and a value for UA. UA is the product of the overall conductance for heat transfer and the surface area on which that conductance is based.
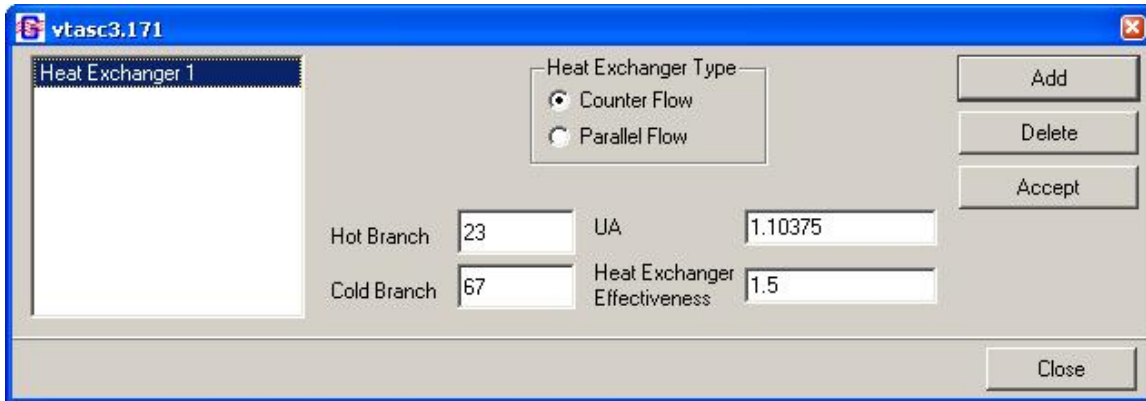
Figure 112. VTASC Heat Exchanger dialog.

The example 5 GFSSP input and output data files (ex5.dat and ex5.out) are included in appendix K (on CD inside back cover). The output file includes all of the input data, the trial solution for the internal nodes, and the final model solution.

The first area of interest is the counter flow heat exchanger effectiveness calculations. During the GFSSP preprocessor input, a value for UA and an effectiveness are input to define the heat exchanger's characteristics. For these verifications a value for UA was assumed with the following process. It was assumed that the heat exchanger had an effectiveness of $\varepsilon = 0.7$. A GFSSP model was run with that effectiveness to obtain the mass flow rates. The mass flow rate for the cold branch was calculated to be 5.41 lbm/s and the flow rate for the hot branch was calculated to be 0.885 lbm/s. Then, equations (81) and (82) were used to calculate the hot and cold fluid capacity rates:[24]

$$C_h = \left(\dot{m}\, C_p \right)_{\text{hot branch}} = (0.885 \text{ lbm/s})(0.9978 \text{ Btu/lbm} - {}^{\circ}\text{R}) = 0.883 \text{ Btu/s-}{}^{\circ}\text{R} \tag{81}$$

and

$$C_c = \left(\dot{m}\, C_p \right)_{\text{cold branch}} = (5.41 \text{ lbm/s})(1.0014 \text{ Btu/lbm} - {}^{\circ}\text{R}) = 5.418 \text{ Btu/s-}{}^{\circ}\text{R} \ . \tag{82}$$

Based on the previously calculated values, $C_{max} = C_c$ and $C_{min} = C_h$. For the counter flow heat exchanger case, it was assumed that C_{max}, C_{min}, and ε would remain the same as the values above. Next, a counter flow exchanger performance table was used, along with the previously calculated values, to estimate the number of heat transfer units, $N_{tu} = 1.25$, for the counter flow exchanger.[24] Then eq. (83) was used to calculate UA in eq. (84):[24]

$$N_{tu} = \frac{UA}{C_{min}} \tag{83}$$

and

$$UA = N_{tu}C_{min} = (1.25)(0.883 \text{ Btu/s -°R}) = 1.10375 \text{ Btu/s -°R}) \ . \tag{84}$$

A value >1 was used for the effectiveness, which instructed GFSSP to calculate the effectiveness instead of employing a user input value. For this case, the counter flow heat exchanger option was chosen. Figure 113 shows the predicted temperatures and mass flow rates from this model.
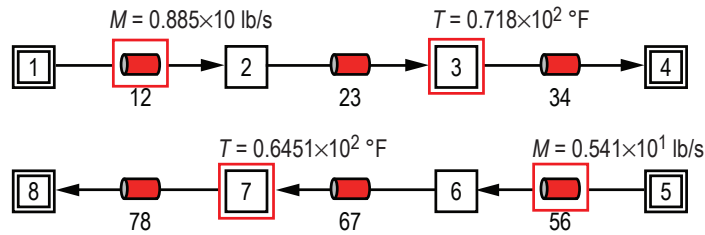
Figure 113. Temperature and flow rate predictions in heat exchanger.

The mass flow rates predicted by this model (0.885 lbm/s at branch 12 and 5.41 lbm/s at branch 56) were the same as predicted for the counter flow exchanger that was used for the assumed case. Therefore, C_c and C_h remained the same. The heat exchanger effectiveness is not included in the data written to the GFSSP output file so it was necessary to calculate the effectiveness that GFSSP used. GFSSP's effectiveness was calculated using eq. (85):[24]

$$\varepsilon = \frac{C_h\left(T_{h,in} - T_{h,out}\right)}{C_{min}\left(T_{h,in} - T_{c,in}\right)} = \frac{C_c\left(T_{c,out} - T_{c,in}\right)}{C_{min}\left(T_{h,in} - T_{c,in}\right)}$$

$$\varepsilon = \frac{(0.883 \text{ Btu/s–°R})(560 \text{ °R} - 531.8 \text{ °R})}{(0.883 \text{ Btu/s–°R})(560 \text{ °R} - 520 \text{ °R})} = 0.705 \ . \tag{85}$$

For comparison, a hand-calculated counter flow effectiveness was determined using eq. (86).[24]

$$\varepsilon = \frac{1 - e^{-N_{tu}\left(1 - C_{min}/C_{max}\right)}}{1 - \left(C_{min}/C_{max}\right)e^{-N_{tu}\left(1 - C_{min}/C_{max}\right)}}$$

$$\varepsilon = \frac{1 - e^{-1.25\left(1 - \left(\frac{0.883 \text{ Btu/s-}^{\circ}\text{R}}{5.418 \text{ Btu/s-}^{\circ}\text{R}}\right)\right)}}{1 - \left(\frac{0.883 \text{ Btu/s-}^{\circ}\text{R}}{5.418 \text{ Btu/s-}^{\circ}\text{R}}\right)e^{-1.25\left(1 - \frac{0.883 \text{ Btu/s-}^{\circ}\text{R}}{5.418 \text{ Btu/s-}^{\circ}\text{R}}\right)}} = 0.688 . \tag{86}$$

Good agreement can be seen in a comparison between the hand-calculated value and the GFSSP value.

The second area of interest is the accuracy of GFSSP's temperature predictions at the nodes downstream of the hot and cold heat exchanger branches. Equation (85) was manipulated to come up with equations (87) and (88) which were used to hand calculate $T_{h,out}$ and $T_{c,out}$:[24]

$$T_{h,\text{out}} = T_{h,\text{in}} - \frac{C_{min}\left(T_{h,\text{in}} - T_{c,\text{in}}\right)\varepsilon}{C_h}$$

$$T_{h,\text{out}} = 560 \text{ }^{\circ}\text{R} - \frac{(0.883 \text{ Btu/s-}^{\circ}\text{R})(560 \text{ }^{\circ}\text{R} - 520 \text{ }^{\circ}\text{R})(0.688)}{(0.883 \text{ Btu/s-}^{\circ}\text{R})} = 532.48 \text{ }^{\circ}\text{R} = 72.48 \text{ }^{\circ}\text{F} . \tag{87}$$

$$T_{c,\text{out}} = T_{c,\text{in}} + \frac{C_{min}\left(T_{h,\text{in}} - T_{c,\text{in}}\right)\varepsilon}{C_c}$$

$$T_{c,\text{out}} = 520 \text{ }^{\circ}\text{R} - \frac{(0.883 \text{ Btu/s-}^{\circ}\text{R})(560 \text{ }^{\circ}\text{R}{-}520 \text{ }^{\circ}\text{R})(0.688)}{(5.418 \text{ Btu/s-}^{\circ}\text{R})} = 524.49 \text{ }^{\circ}\text{R} = 64.49 \text{ }^{\circ}\text{F} . \tag{88}$$

Comparing the hand-calculated values with GFSSP's temperature results of 71.8 °F at node 3 and 64.51 °F at node 7 in figure 113 shows very good agreement, verifying the heat exchanger temperature calculation process used by the GFSSP code.

6.6 Example 6—Radial Flow on a Rotating Radial Disk

6.6.1 Problem Considered

This example illustrates the rotational effect (centrifugal force contribution) capability of GFSSP by modeling the flow of water through a closed impeller.[41] The impeller has an 11-in

diameter, uses water as the operating fluid, and is running at 5,000 rpm. The 'slip' of the fluid is described by the rotational K-factor (K_{rotation}). K_{rotation} is defined as the ratio of the mean circumferential fluid speed divided by the impeller speed: ($K_{\text{rotation}} = u_\theta/r\omega$). (Higher K_{rotation} factors translate to a higher pressure rise for radially outward flow.) A K_{rotation} for each side of the impeller has been proposed and it is the purpose of this example to validate these K_{rotation} correlations. The proposed correlations are ($K_{\text{front face}} = 0.8455 - 0.1403(r-r_i/r_o-r_i)$) for the front face; ($K_{\text{back face}} = 0.8857 - 0.1762(r-r_i/r_o-r_i)$) for the back face. It is desired to compare the results with experimental data. The impeller is schematically shown in figure 114. For this example, the effects of friction will be neglected for the rotating branches.
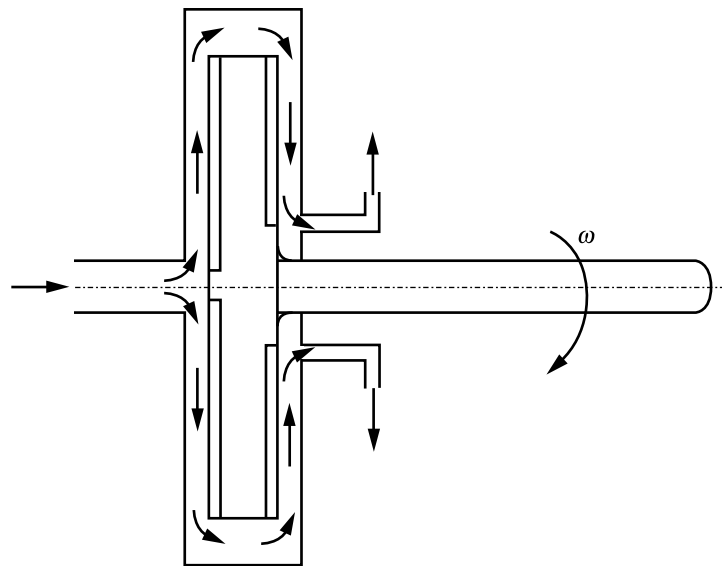
Figure 114. Flow schematic of a rotating radial disk (example 6).

6.6.2 GFSSP Model

The GFSSP model circuit is shown in figure 115. All branches are modeled using option 2. In the model, branches 23, 34, 45, 56, 67, 89, 910, 1011, and 1112 are rotating at 5,000 rpm. The inlet and outlet radii are defined in the preprocessor for each of the rotating branches. The area of each of the radial branches is calculated as the average cross-sectional area for each branch, $\left(A_{\text{branch } ab} = 1/r_b - r_a \int_a^b 2\pi r dr \right)$. Figure 115(b) shows how this model appears in VTASC.
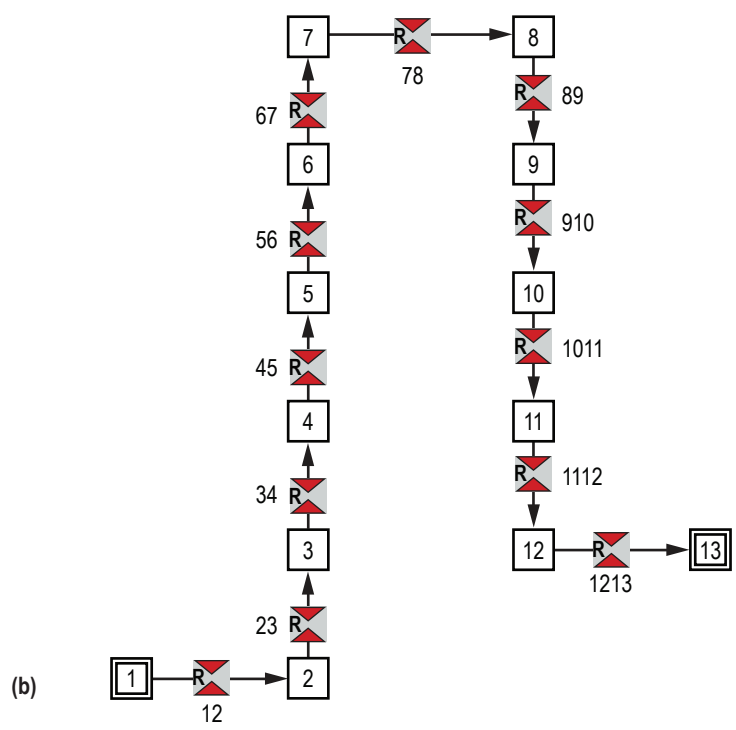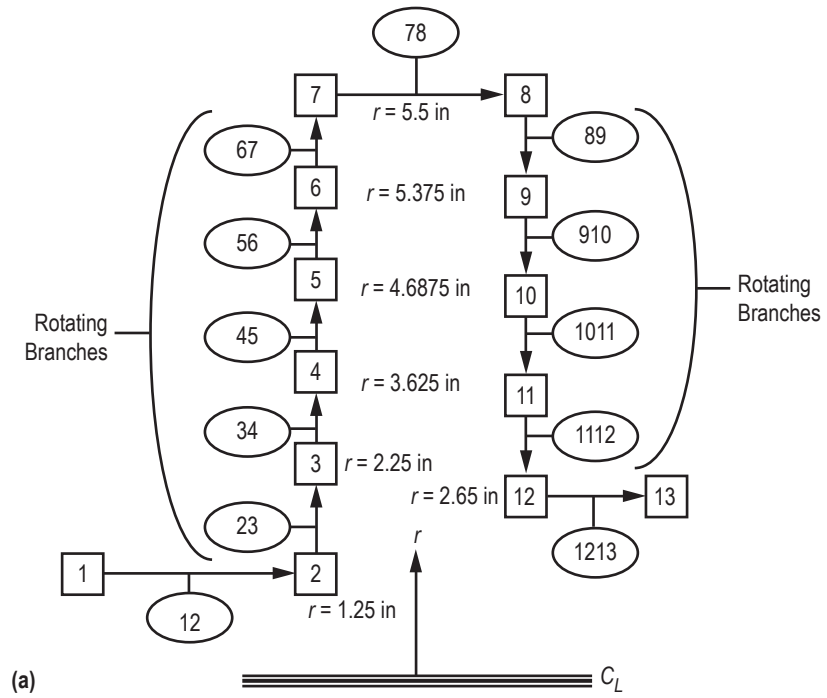
Figure 115. GFSSP model of the rotating radial disk: (a) Detailed schematic and (b) VTASC model.

6.6.3 Results

The example 6 GFSSP input and output data files (ex6.dat and ex6.out) are included in appendix L (on CD inside back cover).

The pressure distribution predicted by GFSSP for the front and back faces of the impeller is shown in figure 116. As is seen in figure 116, the model results show excellent agreement with the experimental data.
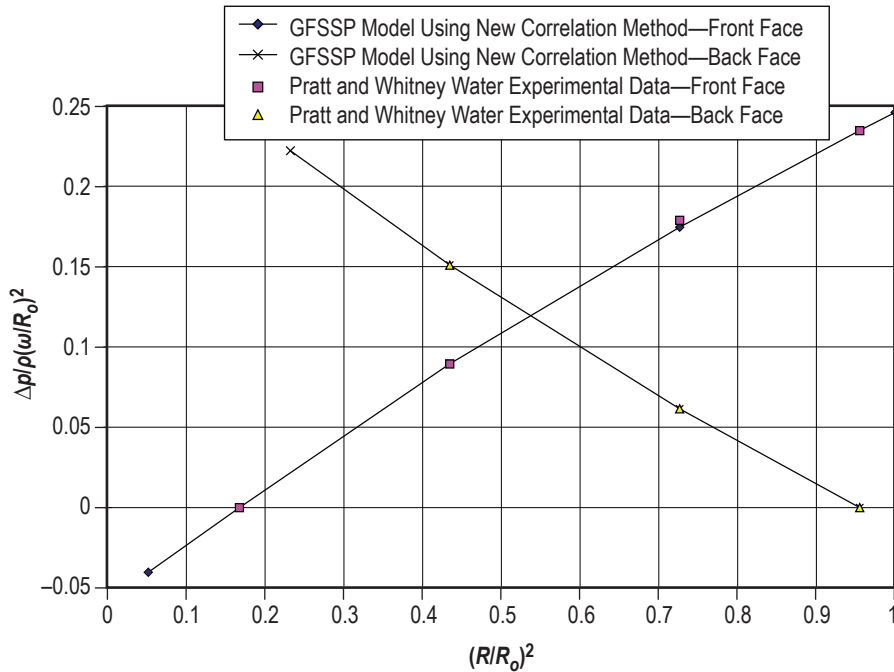
Figure 116. Comparison of GFSSP model results with experimental data.

6.7 Example 7—Flow in a Long-Bearing Squeeze Film Damper

6.7.1 Problem Considered

Squeeze film dampers are used in turbomachinery to dampen out unstable behavior. The damper is installed at the bearing supports of a rotor-stator system on the outer race of a rolling-element bearing. The squeeze film damper consists of inner and outer elements separated by fluid (usually an oil). The inner element is mounted to the outer race of the rolling-element bearing, and the outer element is mounted to the bearing support. The arrangement is similar to a journal bearing except that the inner damper element does not rotate; it only translates. In order to calculate the effect of the squeeze film damper on the system, the forces generated by the squeeze film damper in the radial and tangential directions must be estimated. The forces are estimated by integrating the pressure distribution of the fluid in the damper. The difficulty for the designer/analyst is the estimation of the pressure distribution. The following example problem demonstrates how GFSSP can be used to predict this pressure distribution.

The squeeze film damper considered for this example is shown schematically in figures 117 and 118. Since the damper has sealed ends, the axial flow is neglected. The diameter (d) of the bearing is 5 in, the width (w) of the bearing is 0.94 in, the clearance (c) is 0.0625 in, and the ratio of the dynamic eccentricity (ε, radius of orbit to the clearance) is 0.82. The fluid density (ρ) is 57.806 lbm/ft^3. The fluid viscosity (μ) is 5.932×10^{-3} lbm/(ft•s). The running speed (ω) is set at 1,770 rpm ($\omega = 185.354$ radians/s). GFSSP will be used to determine the pressure distribution around the damper and the results will be compared with experimental data.[42]
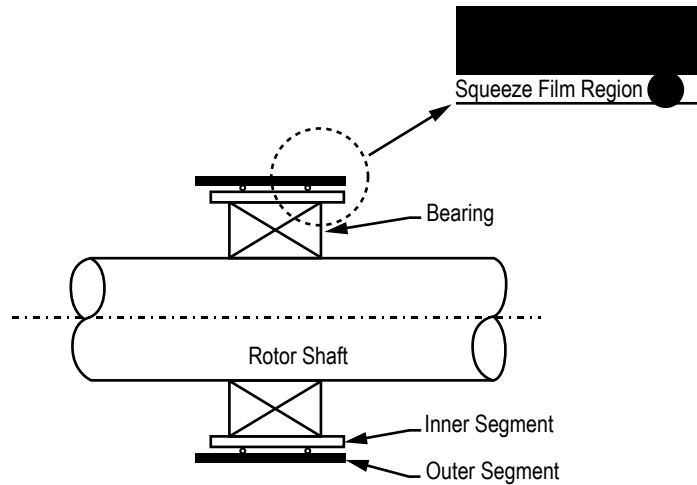
Figure 117. Squeeze film damper schematic (example 7, view 1).
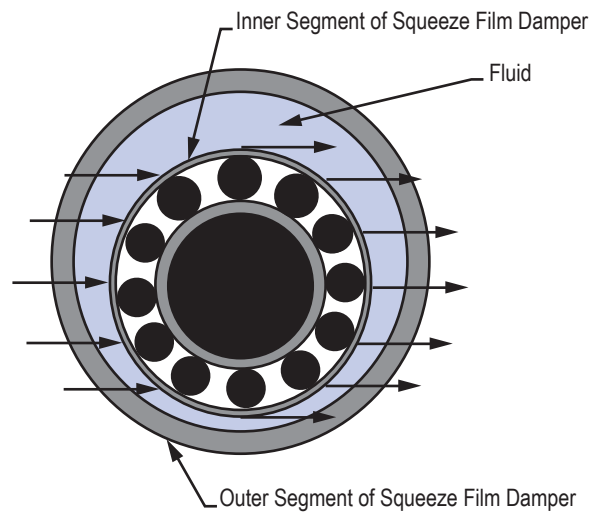
Figure 118. Squeeze film damper schematic (example 7, view 2).

6.7.2 GFSSP Model

A GFSSP model consisting of 20 nodes and 19 branches can approximate the fluid contained within the squeeze film damper system shown in figures 117 and 118. The fluid to be used is not contained in the standard library of fluids and is assumed to be incompressible; therefore, the constant density feature of GFSSP must be used. In order to model the squeeze film damper, the damper will be 'unwrapped.' Figure 119 shows the unwrapping of the damper and the discretization of the flow region. The GFSSP model is shown in figure 120. As is shown in figure 120, nodes 1 and 20 are the boundary nodes. The branches will use branch resistance option 3—Non Circular Duct, Sub-option 1 - Rectangular Duct. The heights of the branches are given in table 21. The motion of the inner element will be simulated using the moving boundary option in GFSSP. (Only the motion of the damper normal to the inner element is modeled in this technique.) The velocity of the moving boundary is given in table 22. Setting the boundary nodes at the same pressure will simulate the periodic behavior of the damper. The boundary pressure is set at 0.0 psi.
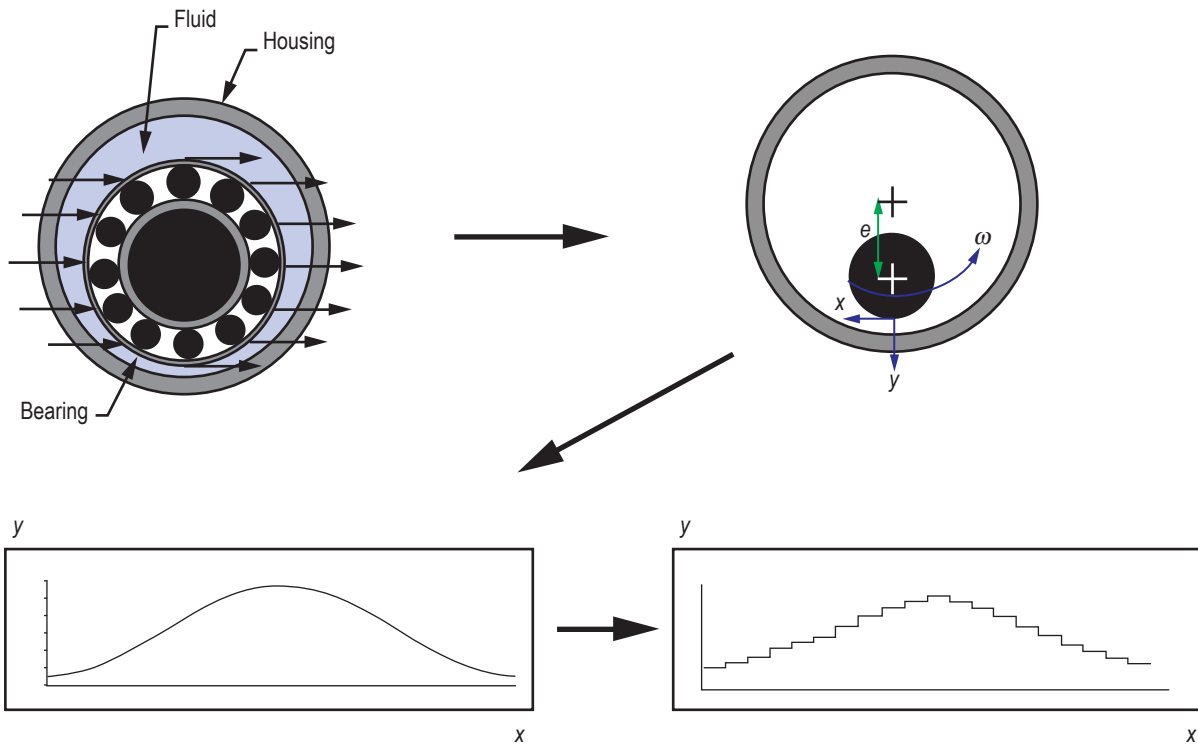
Figure 119. Unwrapping and discretization of squeeze film damper.
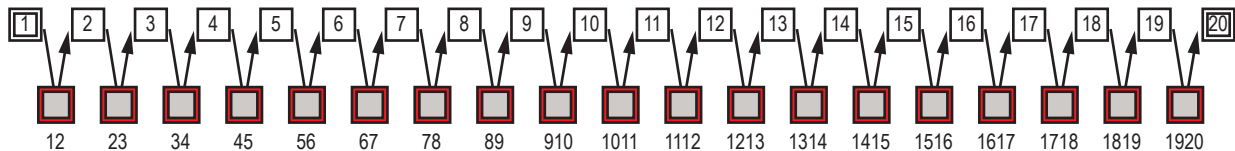
Figure 120. GFSSP model of squeeze film damper.

Table 21. Branch dimensions of squeeze film damper.

Branch Number	Width (in)	Length (in)	Height (in)
12	0.94	0.82673	0.012578
23	0.94	0.82673	0.017987
34	0.94	0.82673	0.028221
45	0.94	0.82673	0.042169
56	0.94	0.82673	0.05832
67	0.94	0.82673	0.074925
78	0.94	0.82673	0.090183
89	0.94	0.82673	0.102441
910	0.94	0.82673	0.11037
1011	0.94	0.82673	0.113113
1112	0.94	0.82673	0.11037
1213	0.94	0.82673	0.102441
1314	0.94	0.82673	0.090183
1415	0.94	0.82673	0.074925
1516	0.94	0.82673	0.05832
1617	0.94	0.82673	0.042169
1718	0.94	0.82673	0.028221
1819	0.94	0.82673	0.017987
1920	0.94	0.82673	0.012578

Table 22. Moving boundary information of squeeze film damper.

Node	Normal Area (in^2)	Velocity (ft/s)
2	0.777126	0.25618
3	0.777126	0.484598
4	0.777126	0.660503
5	0.777126	0.764832
6	0.777126	0.78628
7	0.777126	0.722522
8	0.777126	0.580468
9	0.777126	0.37551
10	0.777126	0.129861
11	0.777126	–0.129861
12	0.777126	–0.37551
13	0.777126	–0.580468
14	0.777126	–0.722522
15	0.777126	–0.78628
16	0.777126	–0.764832
17	0.777126	–0.660503
18	0.777126	–0.484598
19	0.777126	–0.25618

6.7.3 Results

The example 7 GFSSP input and output data files (ex7.dat and ex7.out) are included in appendix M (on CD inside back cover).

The pressure distribution predicted by GFSSP is shown in figure 121. The plot of pressure versus angle (i.e., node position) in figure 121 shows that the pressure is symmetric about

the boundary pressure of zero psi. The model results are compared to experimental results in figure 122. In figure 122, the pressure is normalized with a characteristic pressure (CpRe) and the angle has been converted to compare to a dimensionless time (*wt*) for comparison with the experimental data. As seen in figure 122, the pressure profile of the GFSSP model compares favorably with the experimental results in shape and magnitude, the only major difference between the two results is a phase shift.
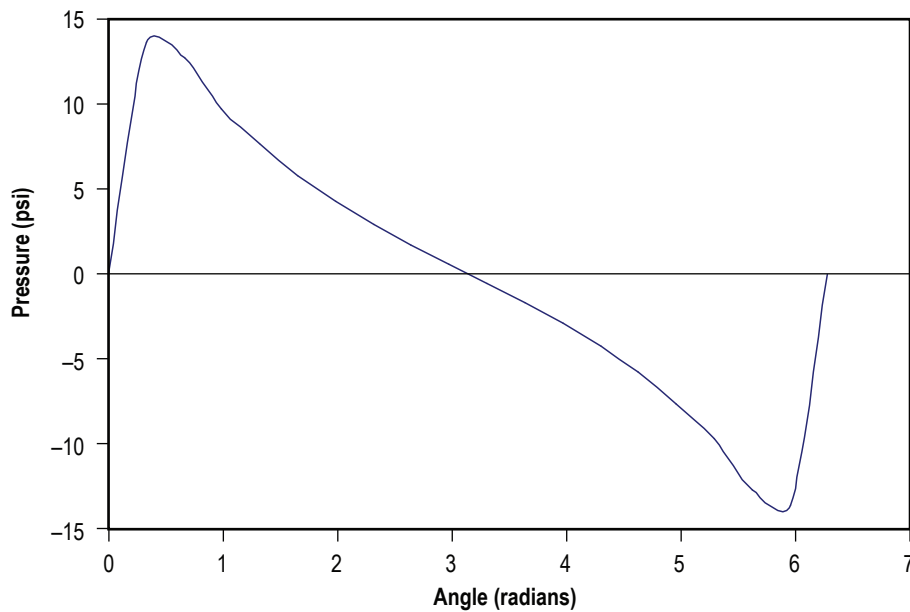
Figure 121. Predicted circumferential pressure distributions in the squeeze film damper.
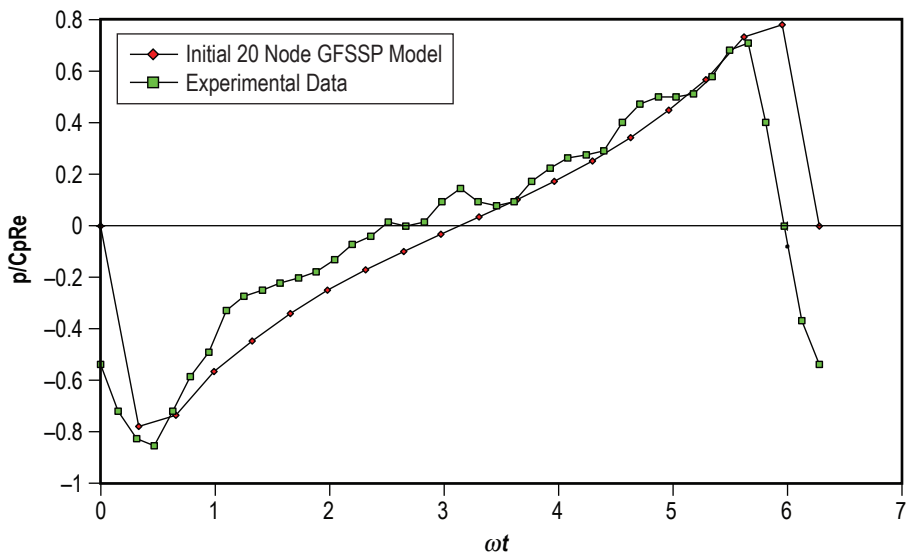
Figure 122. Comparison of GFSSP model results with experimental data
or squeeze film damper.

6.8 Example 8 —Simulation of the Blowdown of a Pressurized Tank

6.8.1 Problem Considered

In the previous examples, the simulation of steady state flow in a given flow circuit was considered. In this example, the capabilities of the unsteady flow formulation of GFSSP to simulate the process of blowing down a pressurized tank will be deployed.

Consider a tank with an internal volume of 10 ft^3, containing air at a pressure and temperature of 100 psia and 80 °F, respectively. Air is discharged into the atmosphere through an orifice of 0.1-in diameter for a period of 200 s. GFSSP will be used to determine the pressure, mass flow rate, and temperature history of the isentropic blowdown process. These predicted values are then compared with the analytical solution.

6.8.2 GFSSP Model

The physical schematic for example 8 is shown in figure 123(a) and a schematic of the corresponding GFSSP model is shown in figure 123(b). Figure 123(c) shows how the model looks in VTASC. The venting process can be modeled with two nodes and one branch. Node 1 is an internal node that represents the tank. For the unsteady formulation, the node volume and the initial conditions must be supplied for each internal node and a history file must be supplied for each boundary node. The history file contains the pressures, temperatures, and concentrations at discrete times. At a minimum, this file should include values at the process start time and at some time corresponding to the expected process stop time. Additional times can be included to account for nonlinear variation in the values if required. The code interpolates in the history file data to determine the values for a particular instant. Shown below is a listing of EX8HS2.DAT, which is the history file of example 8, used to provide the boundary conditions for node 2. The file listing has been annotated to explain the meaning of the entries.

```
EX8HS2.DAT


    2 - Number of data points
    tau(sec)   p(psia)   T (˚F)    Concentration
    0          14.700    80.00     1.00
    1000       14.700    80.00     1.00
```

In addition to supplying the internal node volumes and history data files, the time step (DTAU), start time (TAUF), stop time (TAUL), and print interval (NPSTEP) must also be included within the model input data file when creating an unsteady flow (STEADY = .False.) model.

The initial pressure within the tank (node 1) was 100 psia. Resistance option 22 was used for branch 12 with a flow coefficient of 1. Air is modeled using the ideal gas option that is available in VTASC.

176
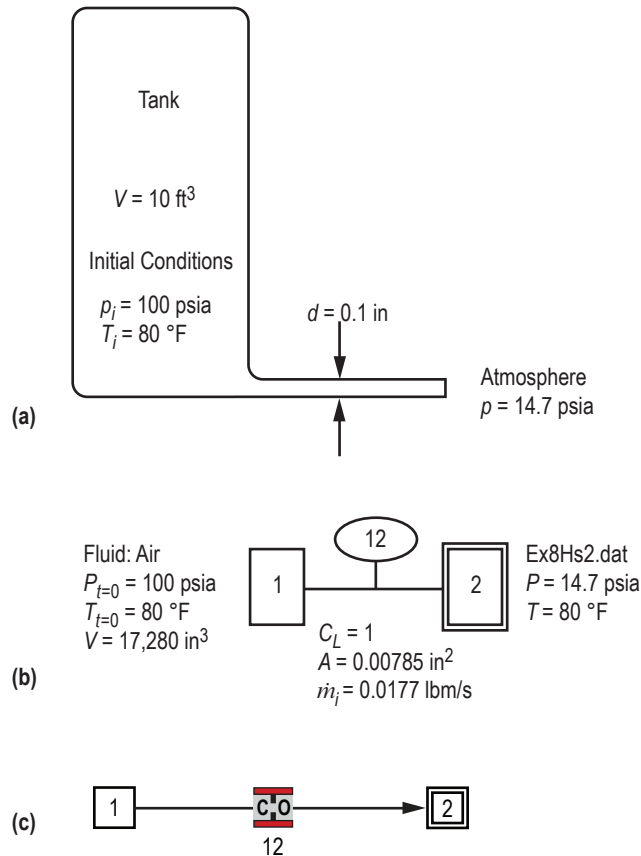
Figure 123. Venting air tank schematics: (a) Physical schematic, (b) detailed model
schematic, and (c) VTASC model.

As an interesting note, this example could be used to model an isothermal process by using the SORCEQ User Subroutine. The temperature of the fluid remains constant in an isothermal process. In this example it is presumed that initially the air and tank wall are at the same temperature. During blowdown, the air temperature tends to drop. With heat transfer from the wall, temperature drop would be less compared to an isentropic process. For an isothermal process, there will be no change in temperature. This particular situation (isothermal) can be modeled by setting an infinite heat transfer coefficient between the wall and fluid in SORCEQ.

6.8.3 Results

The input and output files of this example are included in appendix N as ex8.dat and ex8.out (on CD inside back cover). It may be noted that for each time step, solutions for each node and branch are printed in the output file.

6.8.3.1 Analytical Solution. The differential equation governing an isentropic blowdown process can be written as:

$$\left(\frac{p}{p_i}\right)^{(1-3\gamma)/2\gamma} \frac{d(p/p_i)}{d\tau} = \frac{\gamma A}{\rho_i V}\sqrt{\gamma g_c p_i \rho_i}\left(\frac{2}{\gamma+1}\right)^{(\gamma+1)/2(\gamma-1)} . \tag{89}$$

This is an initial value problem and the initial conditions are:

$$\tau = 0, \ \frac{p}{p_i} = 1 . \tag{90}$$

The analytical solution for p/p_i is given by Moody[43] as:

$$\frac{p}{p_i} = \left[1+\left(\frac{\gamma-1}{2}\right)\left(\frac{2}{\gamma+1}\right)^{(\gamma+1)/2(\gamma-1)}\sqrt{\frac{\gamma g_c p_i}{\rho_i}}\frac{A\tau}{V}\right]^{-2\gamma/(\gamma-1)} . \tag{91}$$

The analytical and GFSSP solutions are compared in figure 124. The figure shows a comparison between the GFSSP solution and the analytical solution of pressures. The difference in pressures is also shown plotted for three different time steps (1, 0.1, and 0.01 s). The discrepancies between analytical and numerical solutions are found to diminish with reduction in time step. This observation is in conformity with expectations.
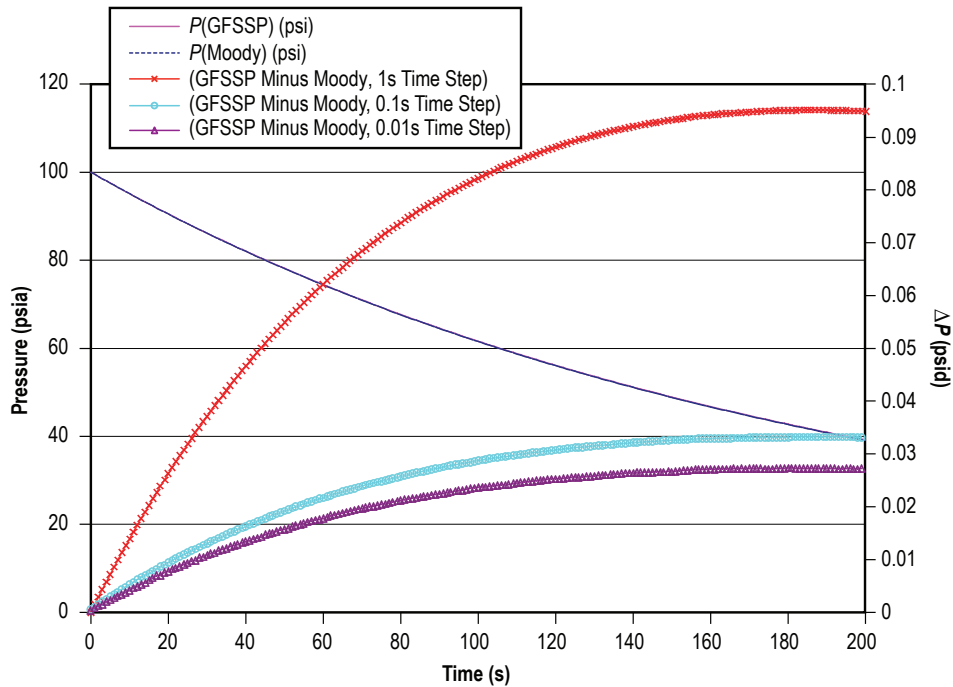
Figure 124. Comparison of the predicted pressure history by GFSSP and the analytical solution.

6.9 Example 9—A Reciprocating Piston-Cylinder

6.9.1 Problem Considered

This example further illustrates GFSSP's capability to model complex unsteady flow. Figure 125 shows the piston-cylinder configuration considered by this example problem. The cylinder has a diameter of 3 in. Within the cylinder is nitrogen gas, sealed in by a piston moving at a rotational speed of 1,200 rpm and a stroke of 3 in. GFSSP will be used to predict the pressure and temperature within the system and the results will be compared with the isentropic solution.
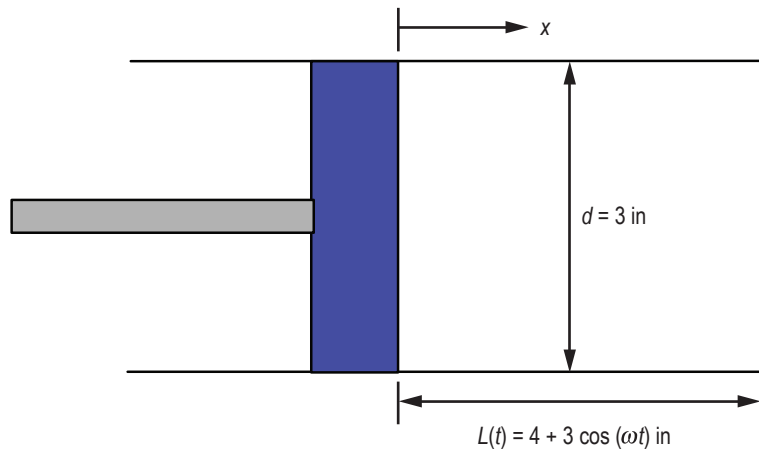
Figure 125. Piston-cylinder configuration.

6.9.2 GFSSP Model

In order to model this configuration, a coordinate transformation is utilized. In this new coordinate system, the endplate of the cylinder is modeled as another piston and the origin of the coordinate system is at the midpoint between the two 'pistons.' Figure 126 demonstrates the modified piston-cylinder arrangement.

The GFSSP model of the piston-cylinder arrangement consists of two internal nodes and one branch (note: the model does not have any boundary nodes). Figure 127(a) shows the GFSSP piston-cylinder model. In order to model the motion of the piston, two special options are utilized: the moving boundary option and the variable geometry option. The moving boundary option is required to adequately model the work input by the motion of the pistons. The variable geometry option is required to model the variation of the geometry of branch 12. The initial condition of the nitrogen in the cylinder is 14.7 psia and 75 °F. Figure 127(b) shows how the model looks in VTASC.
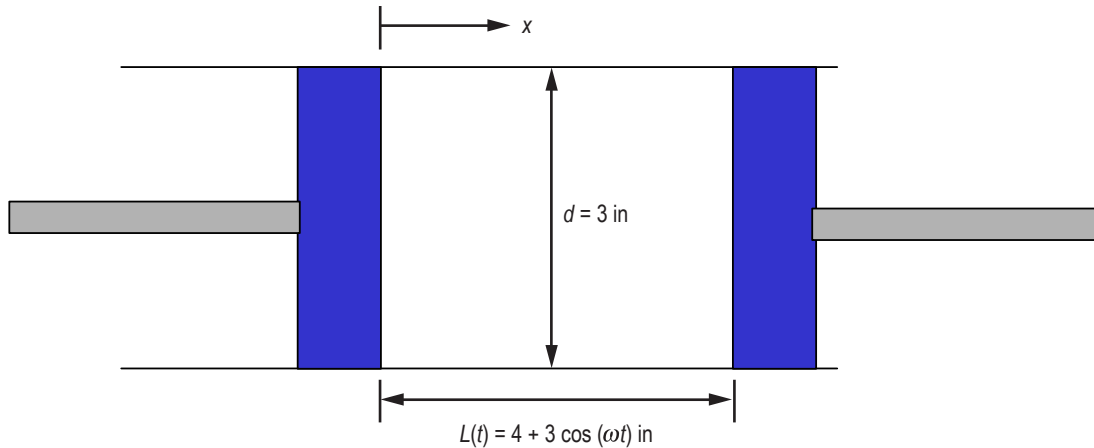
Figure 126. Coordinate transformed piston-cylinder configuration.
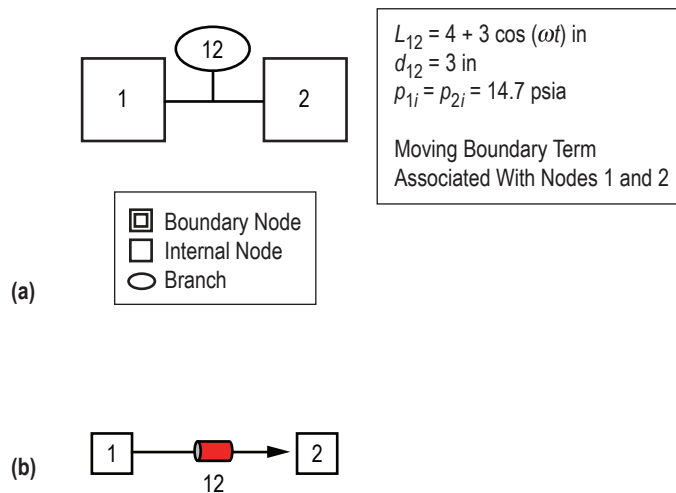
Figure 127. GFSSP model of the piston-cylinder: (a) Detailed model schematic and (b) VTASC model.

6.9.3 Results

Appendix O contains the input, variable geometry history, and output files for this example (on CD inside back cover). The results of the study are compared to an analytical solution (for constant ratio of specific heat, γ). Equations (92) and (93) are used to obtain the analytical solution assuming an isentropic process:

$$T(t) = \left\{ (T_0 + 459.6) \left[\frac{\rho(t)}{\rho_0} \right]^{\gamma-1} \right\} - 459.6 \tag{92}$$

and

$$p(t) = p_0 \left[\frac{\rho(t)}{\rho_0} \right]^{\gamma} \quad , \tag{93}$$

where T_0, p_0, and ρ_0 are temperature, pressure, and density, respectively, at time equal zero.

Figures 128 and 129 compare the results of the GFSSP piston-cylinder model with the analytical solution from equations (92) and (93). As these figures illustrate, the GFSSP model compares favorably to the analytical solution. It should be noted that the isentropic solution uses a constant ratio of specific heats (γ), whereas GFSSP accounts for the variation of specific heat ratios with changes in temperature and pressure.
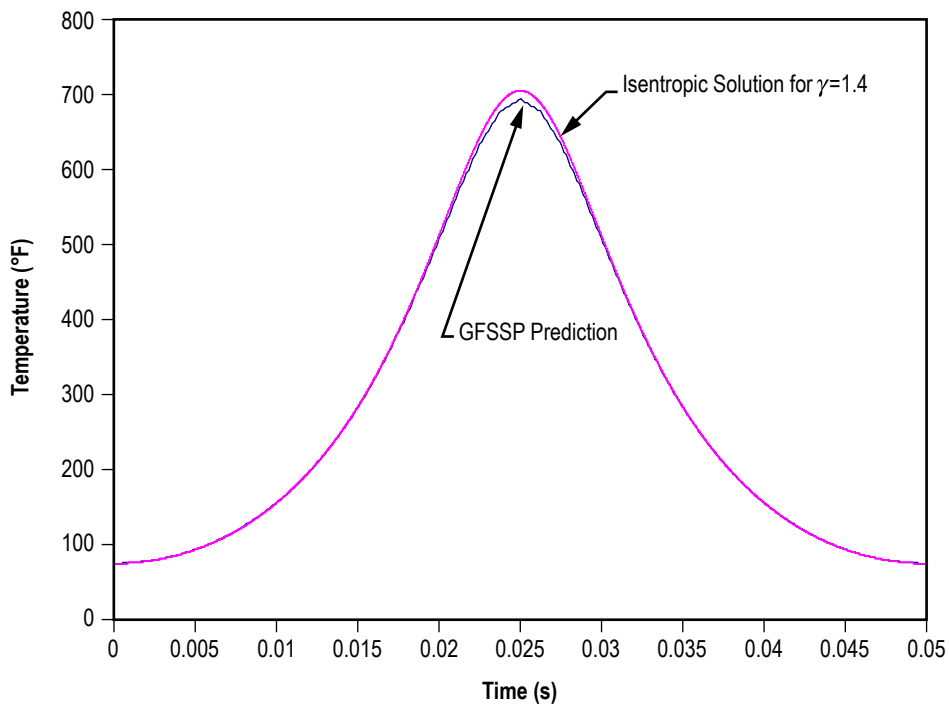
Figure 128. Predicted temperature history of piston cylinder mode.
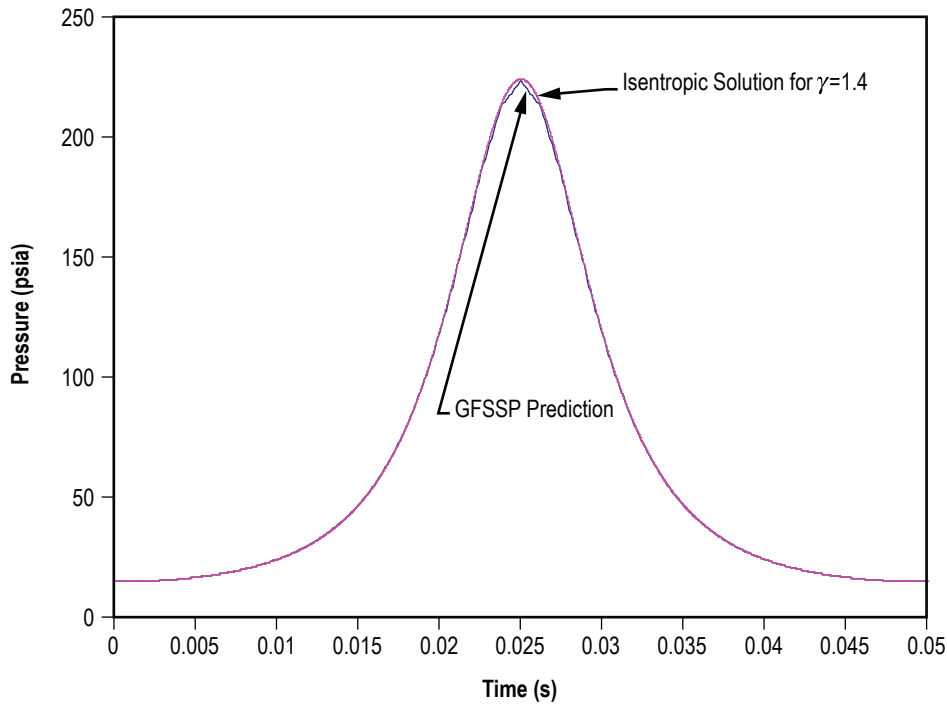
Figure 129. Predicted pressure history of piston cylinder model.

6.10 Example 10—Pressurization of a Propellant Tank

6.10.1 Problem Considered

Example 8 demonstrates the use of GFSSP's unsteady formulation by predicting the pressure and temperature history during the blowdown of a pressurized tank. A more complex unsteady process will be considered in this example, the pressurization of a propellant tank.[32] This example will also illustrate the use of User Subroutines to construct a model of mass transfer due to evaporation of propellant to the ullage space.

The tank pressurization option incorporated in GFSSP models the following physical processes:

(1) Change in ullage and propellant volume.
(2) Change in gravitational head in the tank.
(3) Heat transfer from pressurant to propellant.
(4) Heat transfer from pressurant to the tank wall.
(5) Heat conduction between the pressurant exposed tank surface and the propellant exposed tank surface.
(6) Mass transfer from propellant to ullage.

A schematic of a propellant pressurization system is shown in figure 130. It is assumed that initially the ullage space is filled with pressurant at propellant temperature. As the warm pressurant enters the ullage space, it mixes with the cold ullage gas and the temperature of the ullage gas

182

starts to increase due to mixing and compression. Initially, the walls of the tank are also at propellant temperature. Heat transfer from the ullage gas to the propellant and the tank wall and mass transfer from the propellant to the ullage start immediately after the pressurant begins flowing into the tank. Propellant flows from the tank to the engine under the influence of ullage pressure and gravitational head in the tank. In the current model, condensation of propellant vapor has been neglected.
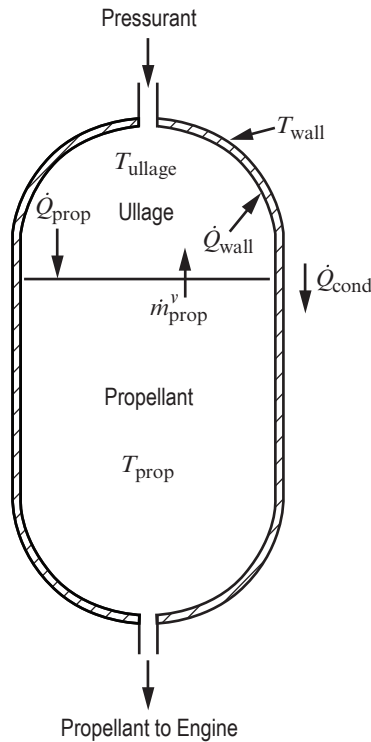
Figure 130. Schematic of propellant tank pressurization system.

6.10.2 GFSSP Model

A five-node pressurization system GFSSP test model, as shown in figure 131(a), was developed to test the implementation of the pressurization option. Helium at 95 psia and 120 °F enters the ullage space, which is initially filled with helium at 67 psia and –264 °F. Node 2 represents the ullage space, which has an initial volume of 25 ft^3. A pseudo boundary node (node 3) has been introduced to exert ullage pressure on the initial propellant volume of 475 ft^3, which is represented by node 4. The pressure at the pseudo boundary node is calculated from the ullage pressure and gravitational head and is the driving force to supply the propellant to the engine. This pressure is calculated at the beginning of each time step. Branch 12 models the tank inlet, branch 34 represents the propellant surface, and branch 45 represents the line to the engine. All three branches were modeled using a Flow Through a Restriction (option 2). The flow coefficient of branch 45 is adjusted to restrict the propellant flow such that all propellant is expelled from the tank over the course of the run. In this test model, the engine inlet pressure was set at 50 psia. Figure 131(b) shows how the model looks in VTASC. Figure 132 shows the VTASC tank pressurization dialog and inputs for example 10.
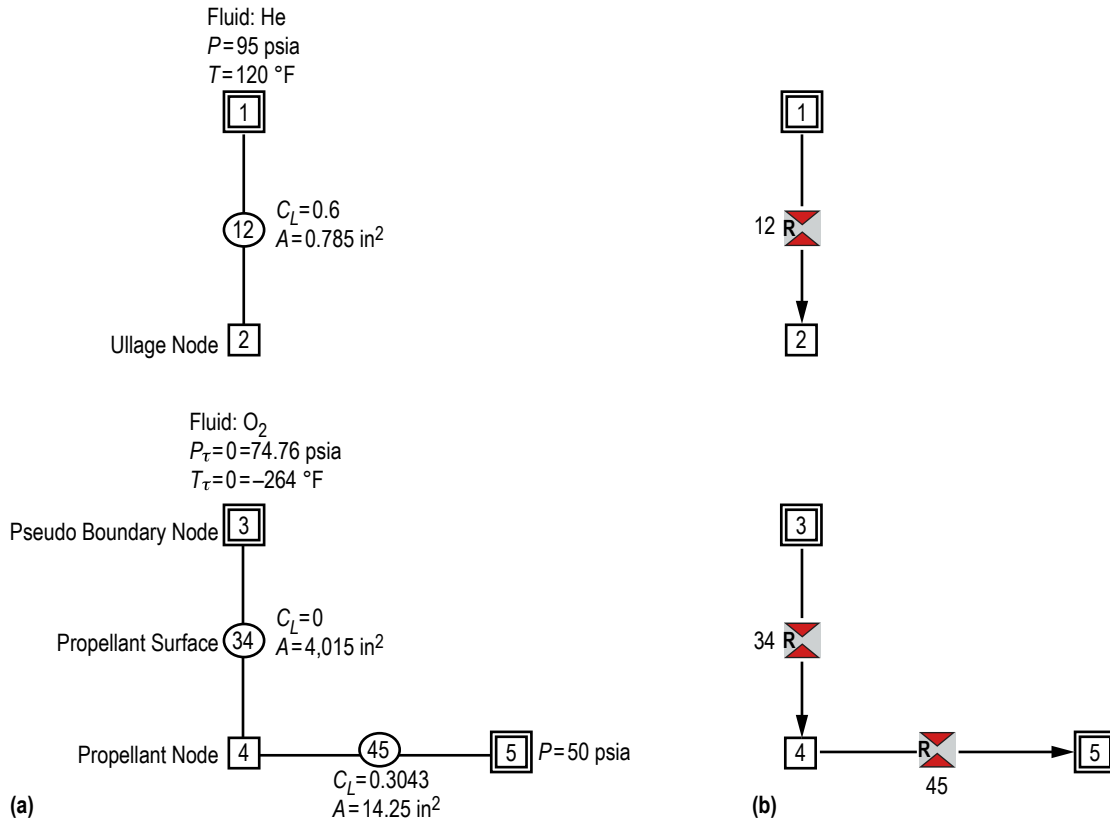
Figure 131. Simple pressurization system test model: (a) Detailed model schematic and (b) VTASC model.
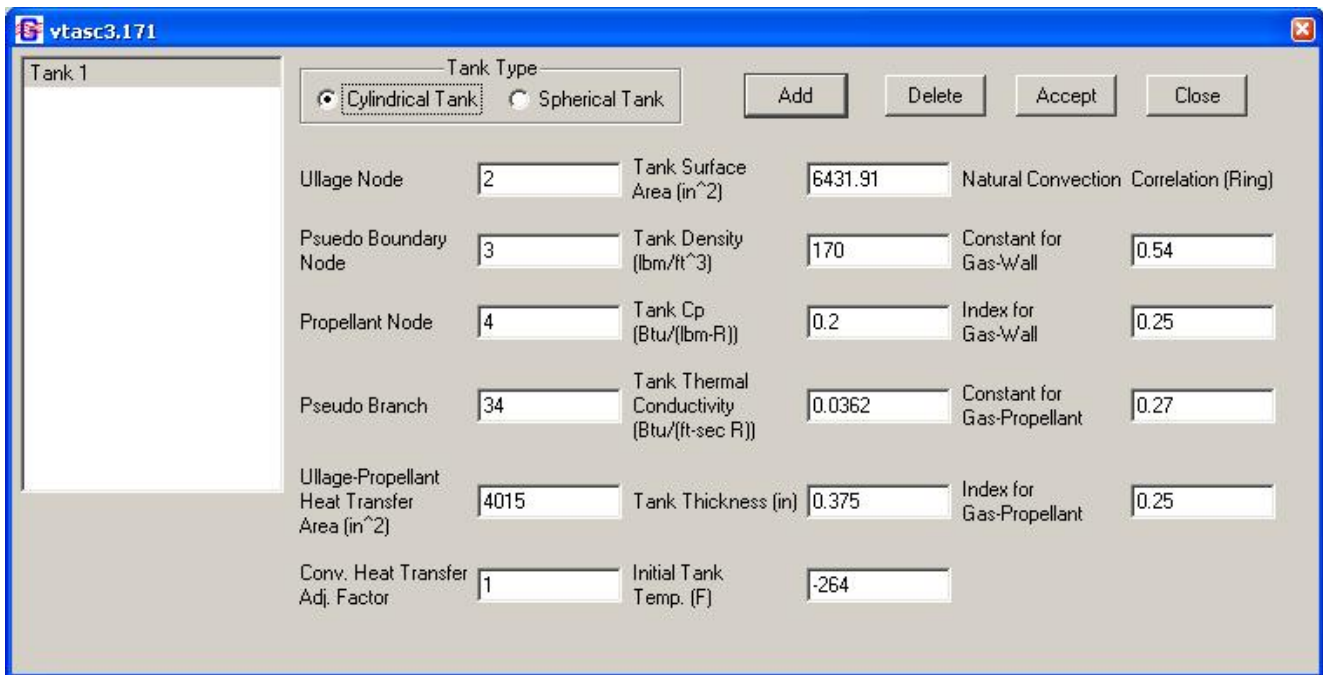
Figure 132. Example 10 tank pressurization dialog.

6.10.2.1 User Subroutine. The calculation of mass transfer from propellant to ullage is not a capability that is available with the pressurization option, so in the course of developing this test model, separate coding was written to account for the mass transfer. This coding was included in the user routine SORCEC. This routine uses the heat transfer rate from the ullage to the propellant to calculate the mass transfer rate of vaporized propellant to the ullage. It is assumed that the propellant is vaporized from the surface and the heat transfer from the ullage only contributes to the vaporization of propellant. The mass transfer due to vaporization is expressed as:

$$\dot{m}^v_{prop} = \frac{\dot{Q}_{prop}}{h_{fg} + c_{pf}\left(T_{sat} - T_{prop}\right)} \ . \tag{94}$$

The saturation temperature in equation (94) is calculated using the vapor pressure relation:

$$\ln p_{sat} = A + \frac{B}{T_{sat}} + C \ln T_{sat} + D T_{sat} \ , \tag{95}$$

where A, B, C, and D are fluid specific vapor pressure relation constants. Table 23 lists the values of the vapor pressure relation constants for the propellants considered in these routines.

Table 23. Vapor pressure relation constants.

Fluid	A	B	C	D
Oxygen	81.66	−2,857	−13.05	0.031
Nitrogen	67.79	−2,156	−10.97	0.0327
Hydrogen	11.4	−211.9	−1.228	0.0405
RP-1	−3,552	888,438	68.05	2.732

The enthalpy of vaporization in equation (94) is calculated using the Clapeyron equation:

$$h_{fg} = T_{sat}\left(v_g - v_f\right)\frac{dP}{dT}\bigg|_{sat} \ , \tag{96}$$

where v_g is found using the Lee and Kesler modified BWR equation as described by Reid et al.[44] with the exception of RP-1, where v_g is calculated using the ideal gas equation. v_f is determined from the following correlation:

$$v_f = C_0 + C_1 T + C_2 T^2 + C_3 T^3 + \dots \ , \tag{97}$$

where C_0, C_1, C_2, etc. are curve fit constants. It should be noted that in the case of RP-1, v_f is assumed to be constant at a value of 0.01923 ft³/lbm. Table 24 lists the values of the correlation constants for the other propellants considered in these routines.

Table 24. Liquid specific volume correlation constants.

	Oxygen	Nitrogen	Hydrogen
C_0	−0.34614	−0.01204	−13.132
C_1	0.011286	0.00061	1.7962
C_2	−0.00013837	$−4.23216\times10^{-6}$	−0.094964
C_3	8.2613×10^{-7}	1.06765×10^{-8}	0.002464
C_4	$−2.4007\times10^{-9}$	–	$−3.1377\times10^{-5}$
C_5	2.7247×10^{-12}	–	1.5712×10^{-7}

6.10.2.2 Subroutine SORCEC. This subroutine is called from subroutine MASSC. The purpose of this subroutine is to calculate the rate of mass transfer of propellant, $\dot{m}^v_{prop}$, in the ullage space due to evaporation. This subroutine can handle four liquid propellants, namely nitrogen, oxygen, hydrogen, and RP-1. For each fluid, the saturation temperature and enthalpy of evaporation were computed in subroutine SATPRP. SORCECON(IPUL,KFLU) is the source term of propellant specie in the ullage node and SORCEMAS is the mass source in the ullage node. The subroutine SATPRP calculates saturation temperature of the propellant at the prevailing ullage pressure. It employs a Newton-Raphson method to compute temperature from the vapor pressure relation shown in equation (95) It also calls subroutine BWR to calculate specific volumes of liquid and vapor at a given pressure and temperature. Finally, it calculates enthalpy of evaporation as given in equation (96).

6.10.2.3 Subroutine PRNUSER. This subroutine is called from subroutine PRINT, with the purpose of writing specific variables in a file for plotting. The variables written in this subroutine include various heat and mass transfer rates, temperature, and volumes. For description of each variable, the reader is referred to appendix D.

6.10.3 Results

The User Subroutine, input and output files including history files of example 10 have been attached in appendix P (on CD inside back cover). The pressurization system transient test model was run for 200 s with 0.1-s time step.

Figure 133 shows both the ullage pressure and tank bottom pressure histories for the test model. After an initial pressure rise due to a 'ramping up' transient effect, both pressures begin a slow but steady decline for the remainder of the run. It should be noted that tank bottom pressure was calculated by adding ullage pressure with pressure due to the gravitational head. Figure 133 shows that as the gravitational head decreases, the ullage and tank bottom pressures slowly converge until all propellant is drained from the tank. The slow decline in ullage pressure is mainly due to the expanding ullage volume.
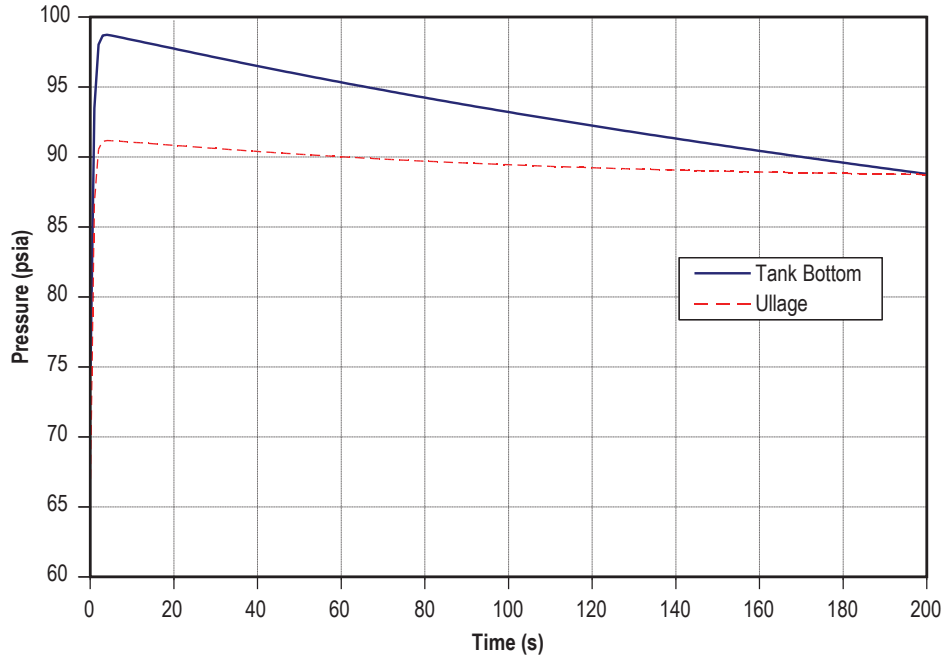
Figure 133. Ullage and tank bottom pressure history.

Figure 134 shows the histories for the ullage temperature and the tank wall temperature. This figure shows that the tank wall temperature rises 32 deg over the course of the model run. It reveals that the 120 °F helium gas entering the tank has an increasing effect on the tank wall as propellant is drained from the tank and the wall surface area exposed to the warmer ullage gas grows. This effect is somewhat dampened, however, because the heat gained by the wall is conducted to the portion of the tank that is submerged in LOX, which acts as a heat sink. The ullage temperature rises 192 deg during the first 60 s of tank pressurization before beginning a slow decline for the remainder of the simulation. This large initial temperature rise is primarily due to the mixing of hot helium gas with the relatively cold gas present in the ullage. The decline in temperature is a result of expansion due to a continuous increase of the ullage volume.
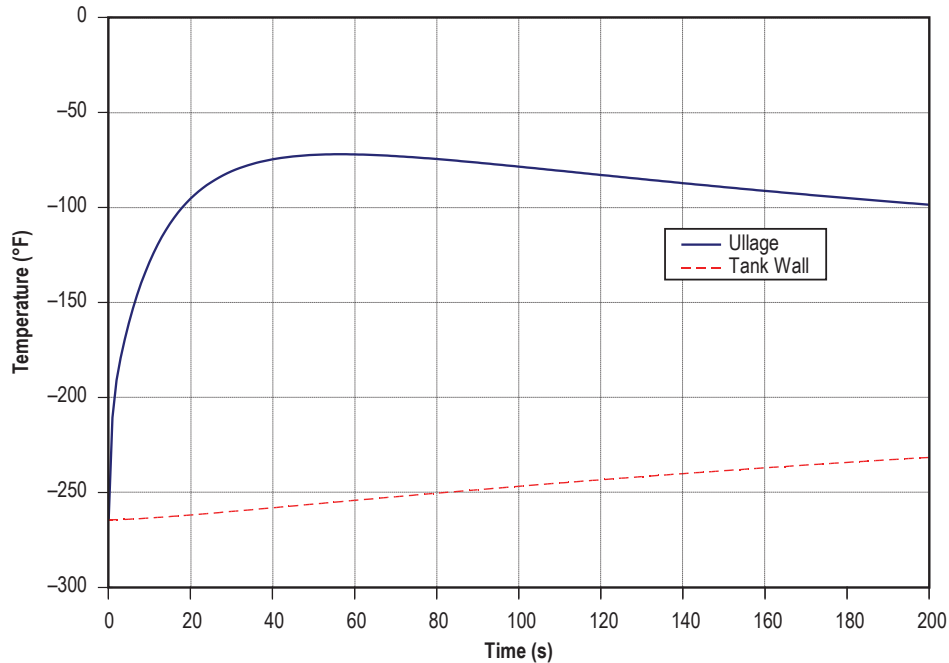
Figure 134. Ullage and tank wall temperature history.

Helium flow rate into the tank is shown in figure 135. The helium flow rate was found to drop initially as the start transient takes place, which is consistent with the 'ramp up' effect noted in figure 133. Then the flow rate begins to gradually increase as ullage pressure drops due to the expanding ullage volume. LOX flow rate into the engine is shown in figure 136. The LOX flow rate curve mirrors the ullage and tank bottom pressure curves, rising through an initial start transient to a peak value and then declining for the remainder of the run as tank pressure drops.
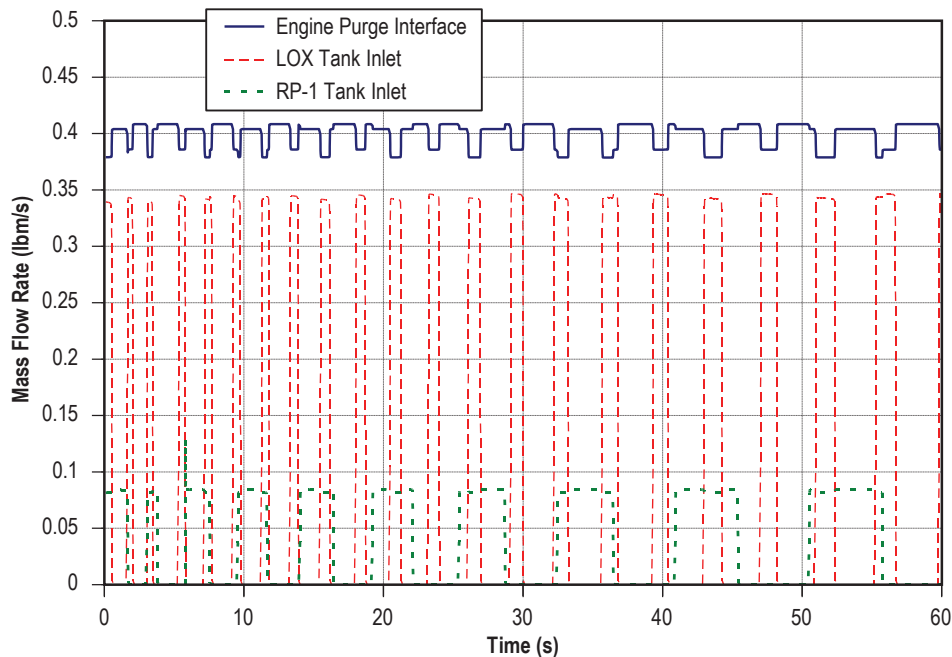
Figure 135. Helium mass flow rate history.
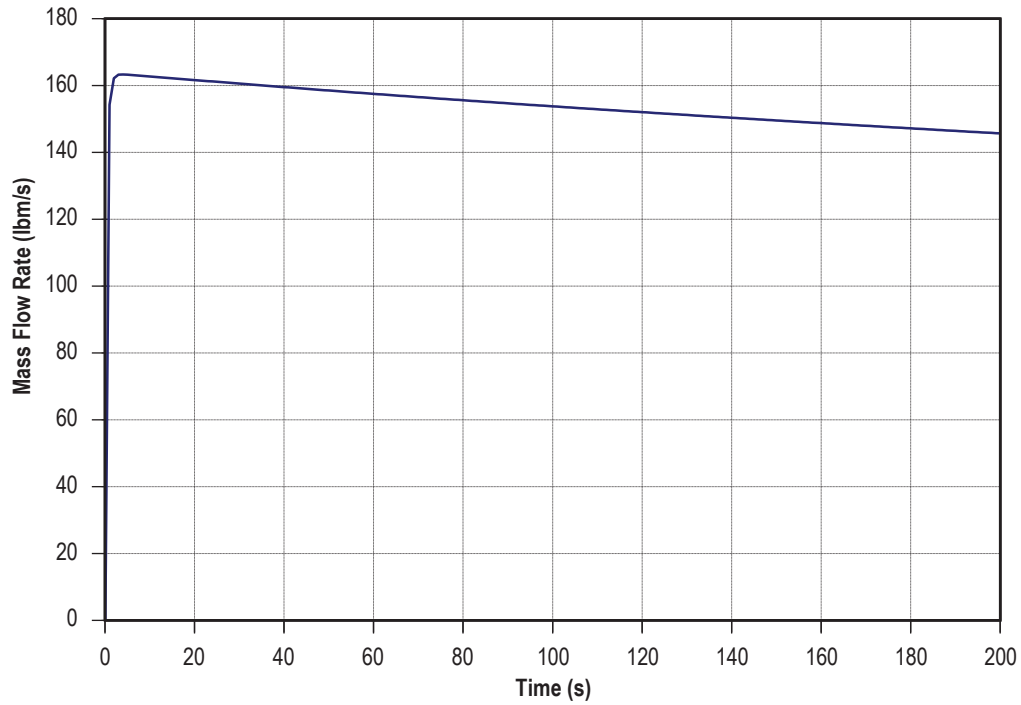
Figure 136. LOX mass flow rate history.

Figure 137 shows the mass transfer rate of gaseous oxygen (GOX) into the ullage space over the duration of the run. The mass transfer rate curve mirrors the ullage temperature curve, which is what one expects since the mass transfer is based on the ullage to propellant heat transfer, which is based on ullage temperature. GFSSP predicts a final GOX mass concentration of 0.15 in the ullage.
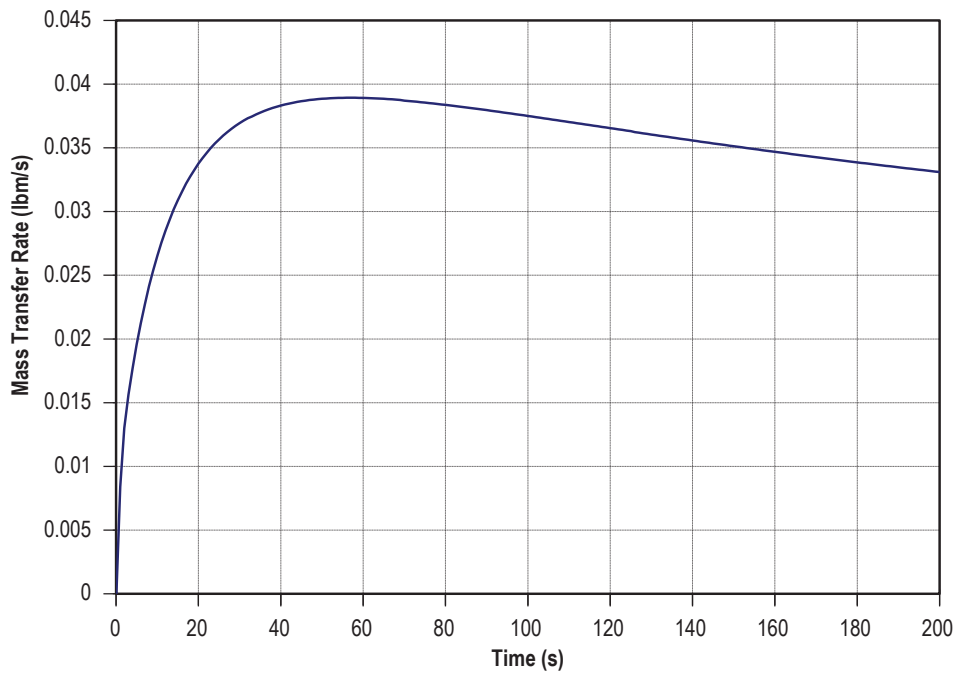
Figure 137. GOX mass transfer rate history.

As a validation, the model results were compared with a published correlation of pressurant requirements for a given displacement of propellant as published by Epstein and Anderson.[45] The correlation calculates the collapse factor, which is defined by Van Dresar[46] as a ratio of the actual pressurant consumption to an ideal pressurant consumption where no heat or mass transfer from the pressurant occurs. This correlation takes the form shown in equations (98) through (102).

$$\frac{w_p}{w_p^0} = \left\{ \left(\frac{T_0}{T_s} - 1 \right) \left[1 - \exp\left(-p_1 C^{p_2}\right) \right] \times \left[1 - \exp\left(-p_3 S^{p_4}\right) \right] + 1 \right\}$$

$$\times \exp\left[-p_5 \left(\frac{1}{1+C} \right)^{p_6} \left(\frac{S}{1+S} \right)^{p_7} Q^{p_8} \right] , \tag{98}$$

where

$$w_p^0 = \rho_G^0 \Delta V , \tag{99}$$

$$C = \frac{\left(\rho c_p^0 \delta \right)_{wall}}{\left(\rho c_p \right)_G^0 D_{eq}} \frac{T_s}{T_0} , \tag{100}$$

$$S = \frac{h \theta_T}{\left(\rho c_p \right)_G^0 D_{eq}} \frac{T_s}{T_0} , \tag{101}$$

and

$$Q = \frac{\dot{q} \theta_T}{\left(\rho c_p \right)_G^0 D_{eq} T_0} . \tag{102}$$

Van Dresar[46] later modified this correlation by redefining D_{eq} as shown in equation (103):

$$D_{eq} = 4 \frac{\Delta V}{A_{sw}} . \tag{103}$$

For this validation exercise, the tank is assumed to be cylindrical and therefore the tank diameter is used in place of Van Dresar's equivalent diameter definition. The tank characteristics used are those values utilized in the GFSSP test model. Also, the ideal pressurant properties of $c_{p_G}^0 = 1.24$ Btu/lbm-R and $\rho_G^0 = 0.06087$ lbm/ft^3 are found using the helium inlet conditions of $P_0 = 95$ psia and $T_0 = 120$ °F. The saturation temperature (T_s) of LOX is taken to be –264 °F. The heat transfer coefficient is calculated to be $h = 8.36 \times 10^{-4}$ Btu/ft^2-s-°R by taking the average value

190

of the heat transfer coefficients calculated by GFSSP at each time step. The change in propellant volume is the value predicted by the GFSSP test model, and the ambient heat flux is neglected in this model. The constants p_1 through p_8 are provided by Epstein and Anderson[45] and are shown in table 25.

Table 25. Constants for LOX propellant.

p_1	0.775
p_2	0.209
p_3	3.57
p_4	0.79
p_5	0.755
p_6	0.271
p_7	0.236
p_8	0.895

Solving equations (100) through (102) and substituting into equation (98) gives $w_p / w_p^0 = 1.51$. Solving equation (99) gives $w_p^0 = 28.9$ lbm. The GFSSP output file predicts a required pressurant mass of approximately 46.29 lbm. Dividing this number by the ideal pressurant mass gives a GFSSP predicted collapse factor of 1.6. Therefore, the predicted discrepancy of GFSSP with respect to Epstein's method is 5.96%. It is believed that this discrepancy is due mainly to the sensitivity of the pressurization process to the heat transfer coefficient, which is difficult to calculate accurately.

6.11 Example 11—Power Balancing of a Turbopump Assembly

6.11.1 Problem Considered

Example 11 illustrates the modeling of the mechanical coupling between two flow components. In the turbopump assembly shown in figure 138, a co-axial shaft mechanically connects the pump and turbine. The power required by the pump must be transmitted from the turbine in order for the system to be in balance. The purpose of this example is to demonstrate this power balancing for a turbopump when used in a gas turbine cycle. The physical plausibility of the predicted results was demonstrated by performing parametric studies on shaft speed.
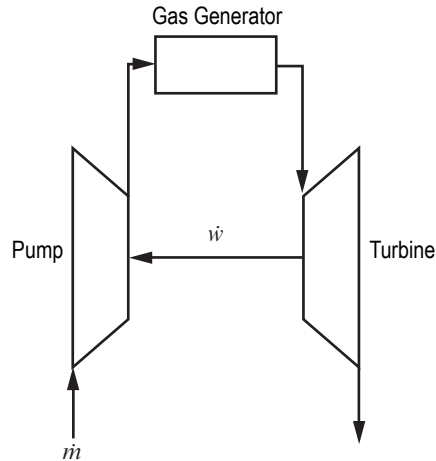
Figure 138. Simplified turbopump assembly.

6.11.2 GFSSP Model

A model of the turbopump portion of a flow circuit is shown in figure 139(a). This model consists of an inlet from a hydrogen tank, a turbopump assembly (pump, turbine and a connection between them (shaft)), two heat exchangers, a bypass dump outlet, and an outlet to the power turbine. The first of the heat exchangers (denoted on fig. 139(a) as the Regenerator) is used to heat a small portion of the main LH_2 flow by using the 'hotter' hydrogen exiting the turbine, while the remainder of the LH_2 flow bypasses this heat exchanger. The second heat exchanger is used to boil and superheat the hydrogen by means of external heat addition. The shaft speed for this model is set in the input file to 80,000 rpm. Figure 139(b) shows how this model looks in VTASC.

This model uses the following options:

• Branch Resistance Options:

   (1) Pipe Flow (option 1).
   (2) Pump with Pump Efficiency (option 15).
   (3) Valve with a Given C_v (option 16).

• Special Options:

   (1) Heat Exchanger (2), (Logical Variable HEX).
   (2) Turbopump Assembly (1) , (Logical Variable TPA).
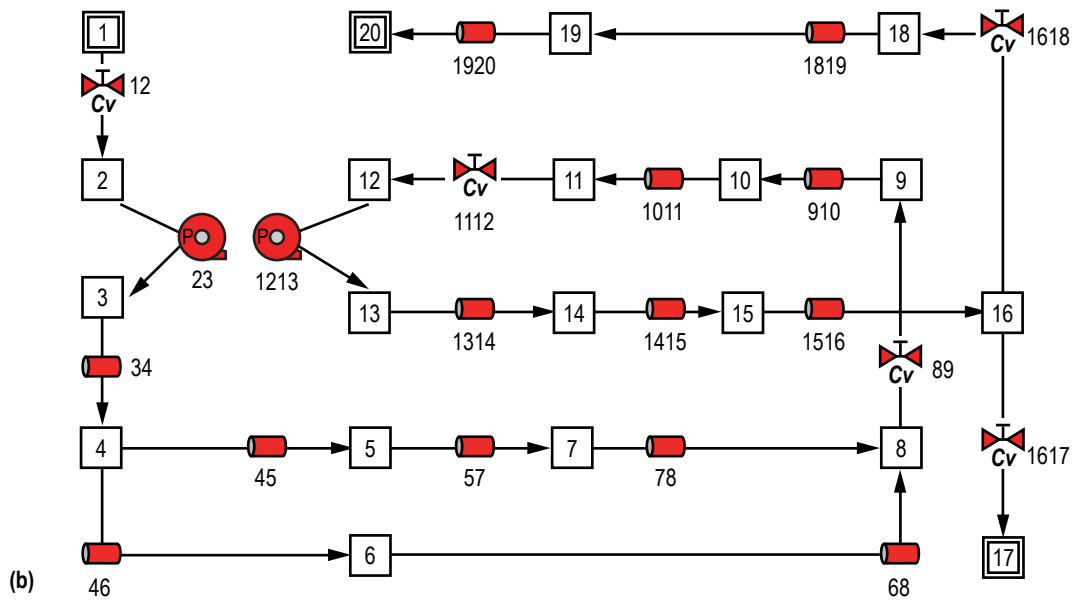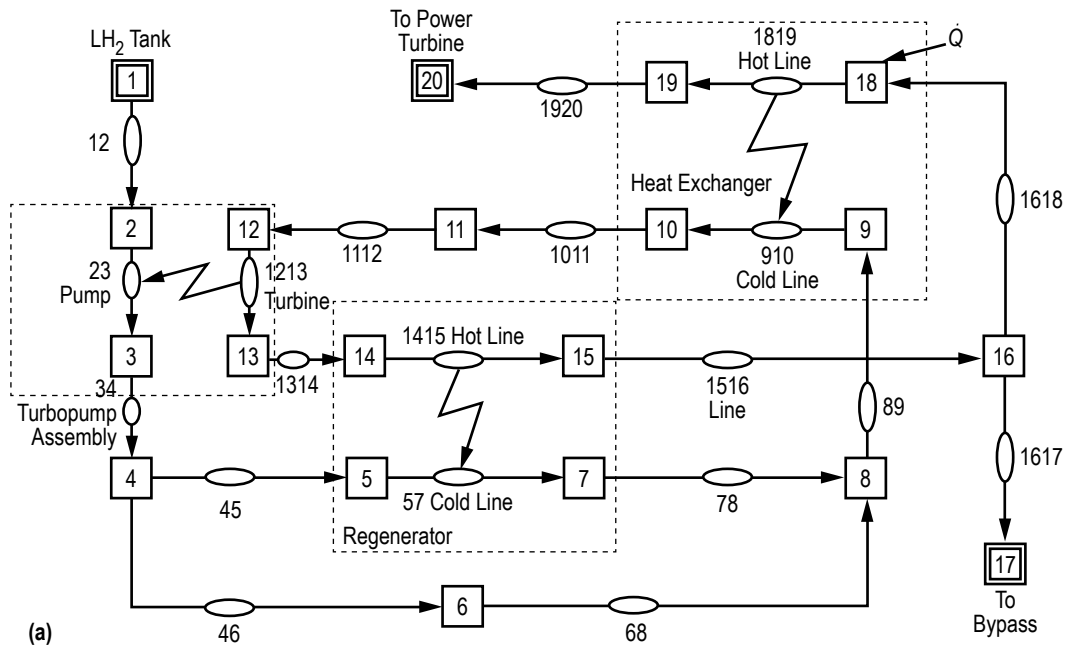
Figure 139. A flow circuit containing turbopump assembly: (a) Detailed model schematic and (b) VTASC model.

Each pipe flow branch has a length of 100 in, an inside diameter of 0.3927 in, and an absolute roughness of 0.00098175 in. Branches 89, 1112, and 1618 each have $C_v = 3.554$ and $A = 0.19635$ in^2. Branch 12 has $C_v = 2.877$ with $A = 0.19635$ in^2 while branch 1617 has $C_v = 0.00354$ with $A = 0.01$ in^2. The branch options chosen to represent the turbine (branch 1213) and the pump (branch 23) have no bearing on the model calculations except for the flow areas that are provided. For this case, option 15 was used for both branches with arbitrary inputs for pump horsepower and efficiency. The flow areas are 0.12112 in^2 for branch 23 and 0.019635 in^2 for branch 1213. The turbopump characteristics are defined in the Turbopump dialog shown in figure 140. The pump characteristic file is also shown below with annotations to explain the meanings of each value. The two Heat Exchanger dialogs are shown in figure 141(a) and (b).
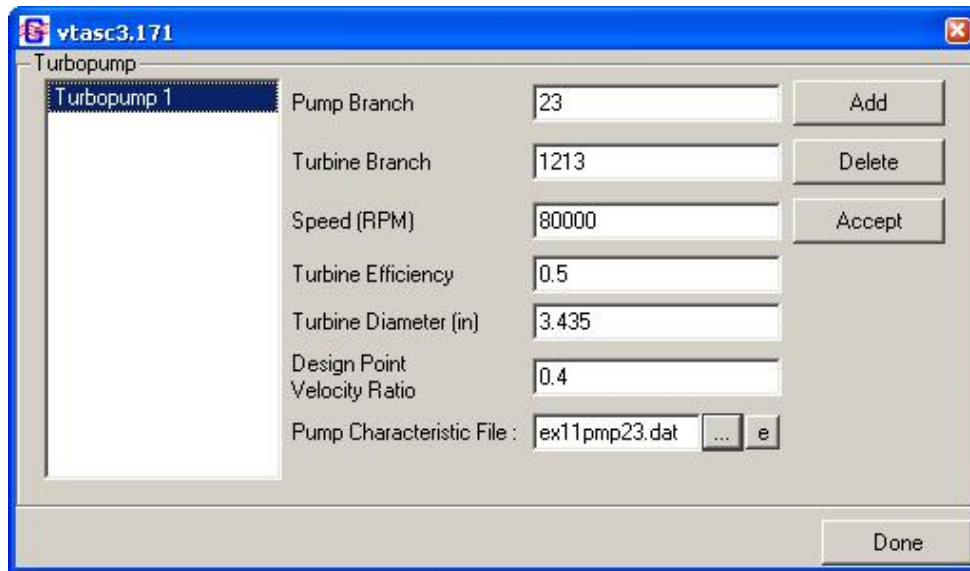
Figure 140. Example 11 Turbopump dialog.

EX11PMP23.DAT

```
18 – Number of lines of data
Flow rate/Speed    Head/Speed²     Torque/(Density*Speed²)
0.000 8.680E-06    0.000
3.035E-05          8.971E-06       8.8724E-10
6.071E-05          9.190E-06       9.7065E-10
9.106E-05          9.341E-06       1.0804E-09
1.214E-04          9.436E-06       1.2166E-09
1.518E-04          9.486E-06       1.3393E-09
1.821E-04          9.486E-06       1.4570E-09
2.125E-04          9.445E-06       1.5644E-09
2.428E-04          9.372E-06       1.6733E-09
2.732E-04          9.263E-06       1.7872E-09
3.035E-04          9.117E-06       1.9105E-09
3.339E-04          8.935E-06       2.0558E-09
3.643E-04          8.753E-06       2.2161E-09
3.718E-04          8.689E-06       2.2698E-09
3.749E-04          8.625E-06       2.2869E-09
```

194

```
3.794E-04          8.479E-06      2.3215E-09
3.807E-04          8.388E-06      2.3281E-09
3.810E-04          0.000E+00      0.000
```
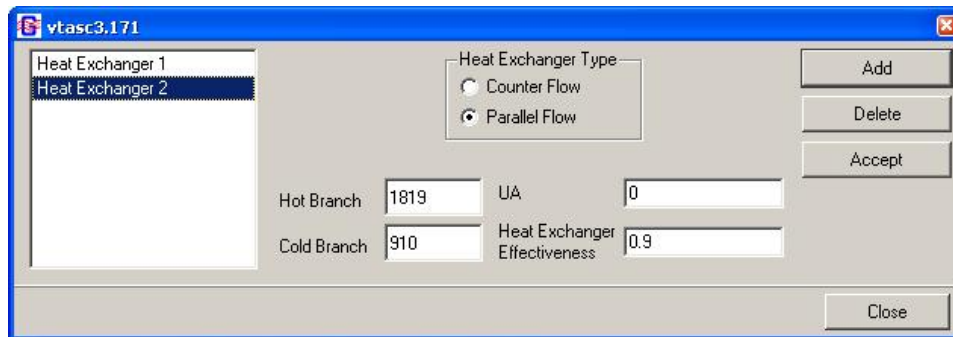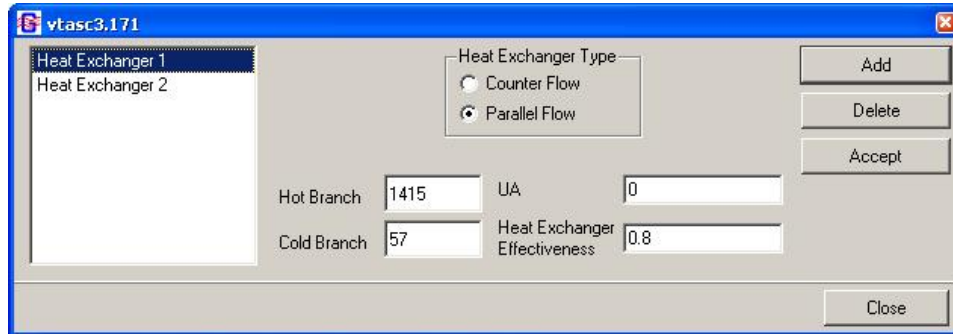

Figure 141. Example 11 Heat Exchanger dialog.

6.11.3 Results

Appendix Q contains the input, pump characteristics, and output files for example 11 (ex11.dat, ex11pmp23.dat and ex11.out) (on CD inside back cover). The results of the study are illustrated in figure 142.
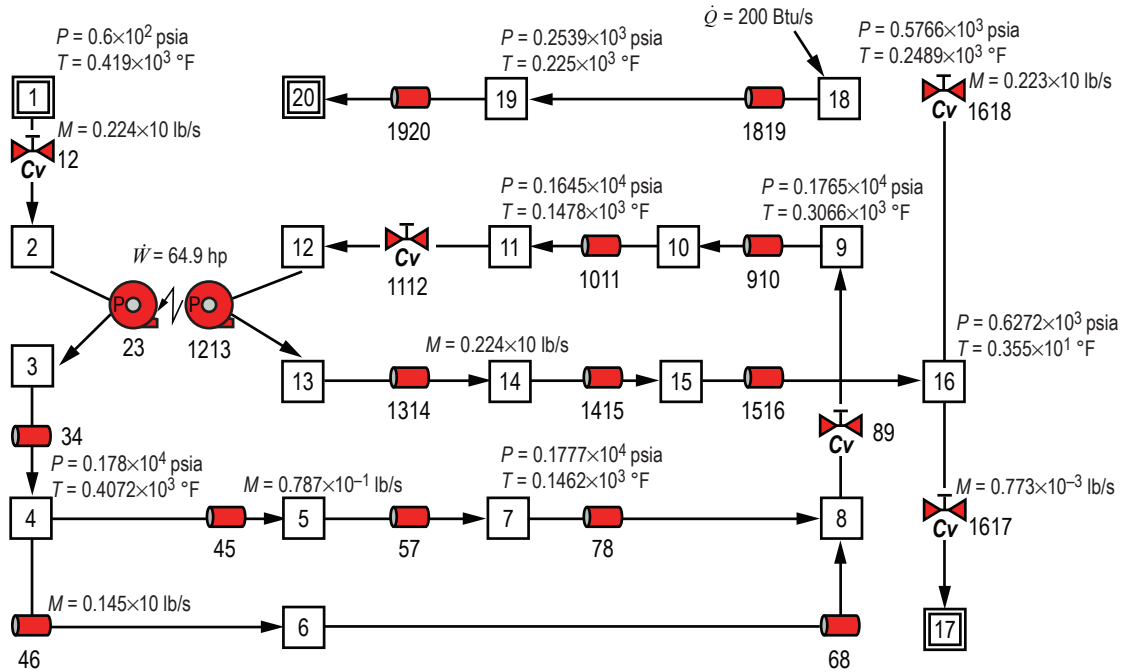
Figure 142. GFSSP RCS model results.

6.11.4 Parametric Study

In order to verify this complicated model, a parametric study on the shaft speed of the turbopump was conducted. Figures 143–145 illustrate the results of this model. Figure 143 illustrates the pressure differential across the turbopump for both the pump and the turbine as a function of the shaft speed. Figure 144 illustrates the hydrogen mass flow rate through the turbopump as a function of the shaft speed. Figure 145 illustrates the torque and horsepower transmitted in the turbopump as a function of the shaft speed. As each of these figures illustrates, a functional relationship is identifiable for each predicted variable as a function of shaft speed.
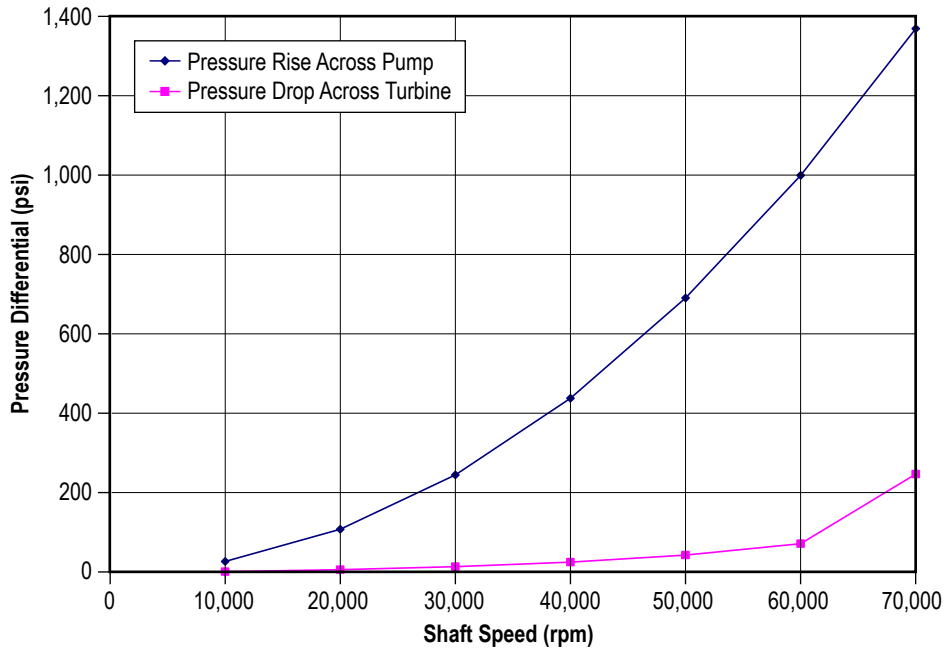
Figure 143. Parametric study results: turbopump pressure differential.
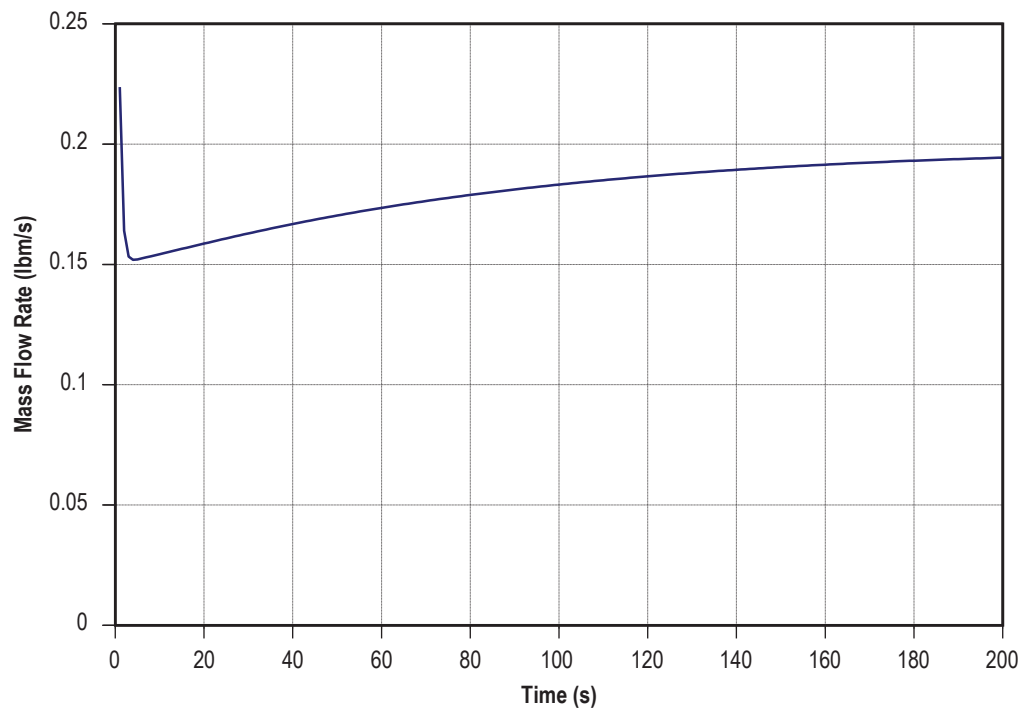
Figure 144. Parametric study results: turbopump hydrogen mass flow rate.
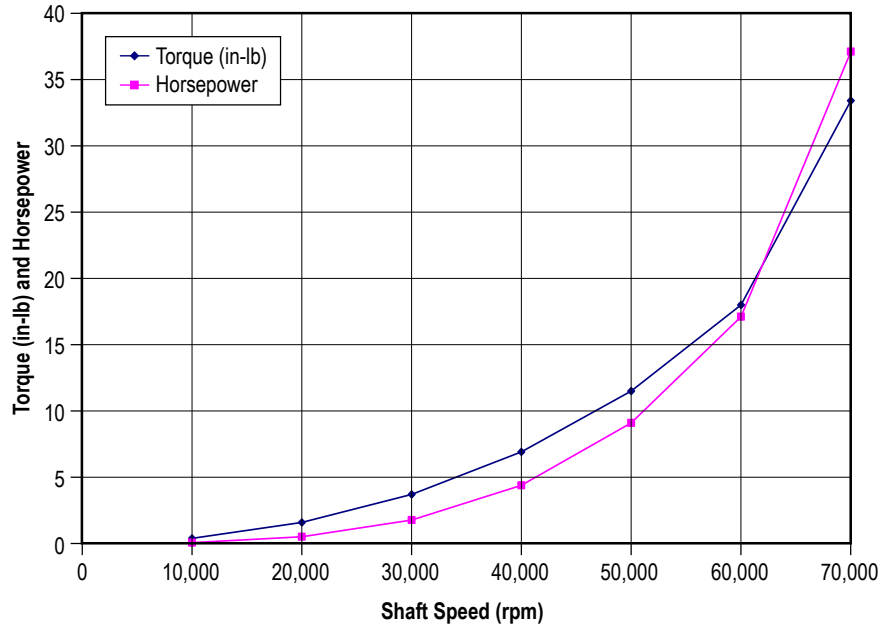
Figure 145. Parametric study results: turbopump torque and horsepower.

6.12 Example 12 —Helium Pressurization of LOX and RP-1 Propellant Tanks

6.12.1 Problem Considered

Example 10 illustrates the use of the pressurization option in modeling ullage and propellant conditions in a tank. In this example, an integrated model consisting of two propellant tanks, a flow network for the ullage pressurant supply from a Facility interface, and engine propellant feed lines will be constructed. The pressurization system of Propulsion Test Article 1 (PTA1) consists of a LOX tank and an RP-1 tank that are both pressurized by helium. This configuration is represented in the schematic shown in figure 146. The objective of the present example is to develop an integrated mathematical model from the helium supply line to the engine inlet to model 60 s of engine operations. The model has three primary functions:

(1) To predict the flow rate and pressure distribution of the helium supply line feeding both the LOX and RP-1 tanks.

(2) To predict the ullage conditions considering heat transfer between the ullage, propellant and the tank wall, as well as mass transfer from propellant to ullage.

(3) To predict the propellant conditions leaving the tank.
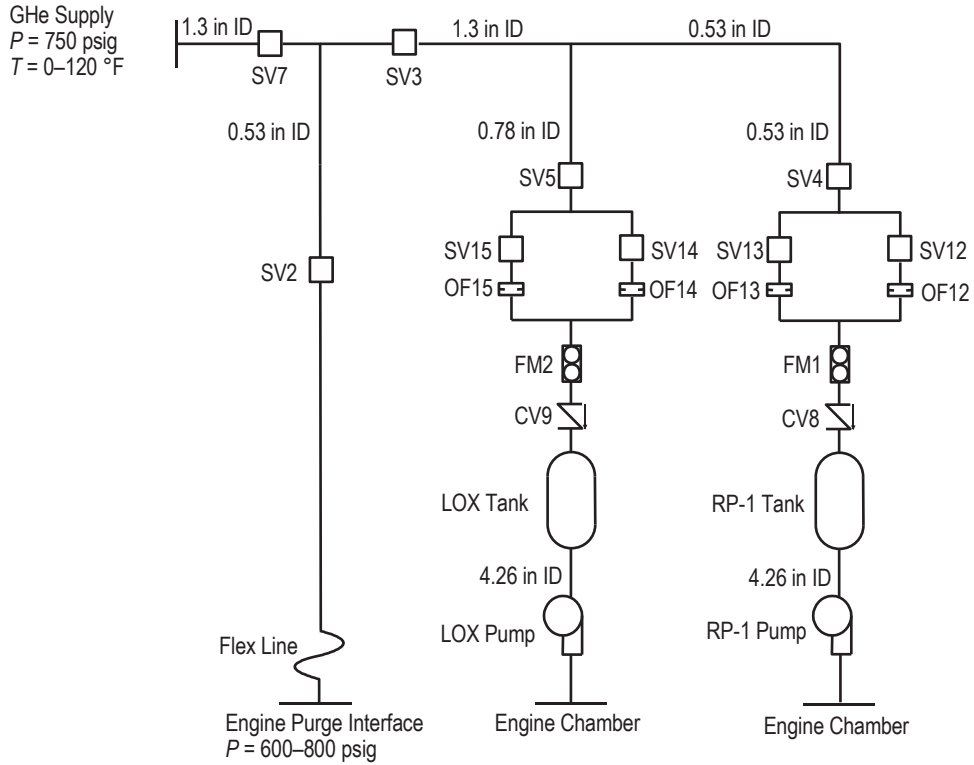
Figure 146. PTA1 helium pressurization system schematic.

6.12.2 GFSSP Model

A GFSSP model of the helium pressurization system of LOX and RP-1 tanks is shown in figure 147(a). The model consists of 65 nodes and 64 branches. The model extends from facility interface to engine purge and engine chamber interfaces. It includes all piping and its fittings, orifices, and valves. Both RP-1 and LOX tanks and pumps are included in the model. Each propellant tank has a diffuser and control valve. Pressure and temperatures are specified at the interfaces, which are represented by six boundary nodes listed in table 26.

Facility Interface
P=765 psia
T=0–120 °F

Pipe
L=128 in
D=1.3 in

SV7
C_L=0.6
A=0.63617 in^2

Tee, Flo Thru
$K1$=200
$K2$=0.1

Pipe
L=17 in
D=1.3 in

SV3
C_L=0.6
A=0.63617 in^2

Pipe
L=288 in
D=1.3 in

Tee, Flo Thru
$K1$=200
$K2$=0.1

Reduction
D_1=1.3 in
D_2=0.53 in

Pipe
L=221 in
D=0.53 in

SV4
C_L=0.6
A=0.2827 in^2

SV2
C_L=0.6
A=0.63617 in^2

Pipe
L=19 in
D=1.3 in

Tee, Flow Thru
$K1$=200
$K2$=0.1

Reduction
D_1=1.3 in
D_2=0.78 in

Tee, Flow Thru
$K1$=200
$K2$=0.1

Reduction
D_1=1.3 in
D_2=0.53 in

Pipe
L=143 in
D=0.53 in

Flex Tube
L=28 in
D=0.53 in

Engine Interface
P=615–815 psia

SV5
C_L=0.6
A=0.63617 in^2

Pipe
L=11 in
D=0.78 in

Tee, Elbow
$K1$=500
$K2$=0.7

Tee, Elbow
$K1$=500
$K2$=0.7

Pipe
L=28 in
D=0.78 in

Pipe
L=12 in
D=0.53 in

Tee, Elbow
$K1$=500
$K2$=0.7

Tee, Elbow
$K1$=500
$K2$=0.7

Pipe
L=7.5 in
D=0.53 in

Pipe
L=14 in
D=0.53 in

SV13
C_L=0.6
A=0.00001 in^2

SV12
C_L=0.6
A=0.2827 in^2

SV15
C_L=0.6
A=0.00001 in^2

Pipe
L=14 in
D=0.53 in

OF13
C_L=0.6
A=0.00785 in^2

OF12
C_L=0.6
A=0.02895 in^2

Pipe
L=9 in
D=0.53 in

For stainless steel tubing, it is assumed that ε=0.0008 in. For steel flex tubing, this roughness is multiplied by 4.

Pipe
L=18 in
D=0.78 in

OF15
C_L=0.6
A=0.01767 in^2

OF14
C_L=0.6
A=0.10179 in^2

Pipe
L=15 in
D=0.78 in

Tee, Elbow
$K1$=500
$K2$=0.7

FM2
C_L=0.77371
A=0.55351 in^2

Tee, Elbow
$K1$=500
$K2$=0.7

FM1
C_L=0.83056
A=0.2255 in^2

Expansion
D_1=0.53 in
D_2=3 in

Pipe
L=14 in
D=0.53 in

CV8
C_L=0.6
A=0.4185 in^2

Pipe
L=14 in
D=0.53 in

Expansion
D_1=0.78 in
D_2=3 in

Pipe
L=21 in
D=0.78 in

CV9
C_L=0.6
A=0.7854 in^2

Pipe
L=13 in
D=0.78 in

Recovery
C_L=0.0
A=7.06858 in^2

V_{ul}=15 ft^3

V_{prp}=285 ft^3

RP1 Diffuser
C_L=0.6
A=37.6991 in^2

RP1 Surface
C_L=0.0
A=3987 in^2

Pipe
C_L=0.181
A=14.25 in^2

RP1 Pump
A_0=168757
B_0=−4.9362

Pipe
C_L=0.0464
A=14.25 in^2

Recovery
C_L=0
A=7.06858 in^2

V_{ul}=25 ft^3

V_{prp}=475 ft^3

LOX Diffuser
C_L=0.6
A=37.6991 in^2

LOX Surface
C_L=0
A=40.15 in^2

Pipe
C_L=0.304
A=14.25 in^2

LOX Pump
A_0=176033
B_0=−2.5799

Pipe
C_L=0.105
A=14.25 in^2

Engine Chamber
P=652 psia

Engine Chamber
P=652 psia

(a)

(b)

Figure 147. GFSSP model of the pressurization system of example 12: (a) Detailed model schematic and (b) VTASC model.

Table 26. Boundary nodes of He pressurization flow circuit.

Boundary Node	Interface
1	Facility interface
65	Engine interface (purge)
55	Ullage-propellant interface (LOX tank)
59	LOX engine chamber interface
30	Ullage-propellant interface (RP-1 tank)
34	RP-1 engine chamber interface

It may be noted that the nodes representing ullage-propellant interface (nodes 55 and 30) are pseudo boundary nodes. The code uses the calculated ullage pressure at the previous time step instead of pressures provided by the user through history files. Helium enters into the system from the facility interface where it is distributed into three parallel branches. The first branching takes place after 128 in of pipeline and this branch supplies helium to the engine for engine purges.

The second branching takes place 305 in downstream of the first branch and this branch supplies helium to the LOX tank. The rest of the helium goes to pressurize the RP-1 tank. The lines leading to the LOX and RP-1 tanks each have two parallel legs, one of which remains closed during operation. The left leg of the circuit is used to pressurize the tank during prepressurization operation while the right leg of the circuit is used to pressurize the tank during pressurization standby and engine operations. In the model discussed in this TM, setting a high resistance in the appropriate branches eliminated the flow to the left leg. Figure 147(b) shows how this model looks in VTASC.

6.12.3 Results

The input and output files including history files of example 12 have been attached in appendix R (on CD inside back cover). The GFSSP model shown in figure 147 was run for a 60-s engine operation period. At the beginning of the model run, the control valve nominal set points are 72 psia for the LOX tank and 55 psia for the RP-1 tank with ±3 psi tolerances. After 3 s they drop 5 psi to 67 psia for the LOX tank and 50 psia for the RP-1 tank with ±3 psi tolerances. The output file contains pressure, temperature, and density at all nodes as well as flow rate, velocity, and pressure drop at all branches for selected time steps.

Figure 148 shows the predicted pressure history of the RP-1 ullage, RP-1 tank bottom, LOX ullage, and LOX tank bottom pressures. The difference in pressure between the tank bottom and ullage is the gravitational head, which slowly reduces as propellant is drained from the tank. The cyclic nature of the pressure profiles is due to the control valves, which are set to close or open as the tank bottom pressures exceed prescribed tolerances. It is observed that the frequency of pressure oscillation is larger in the LOX tank than the RP-1 tank. This observation is attributable to the higher flow rates associated with the LOX tank as compared to those required for the RP-1 tank.
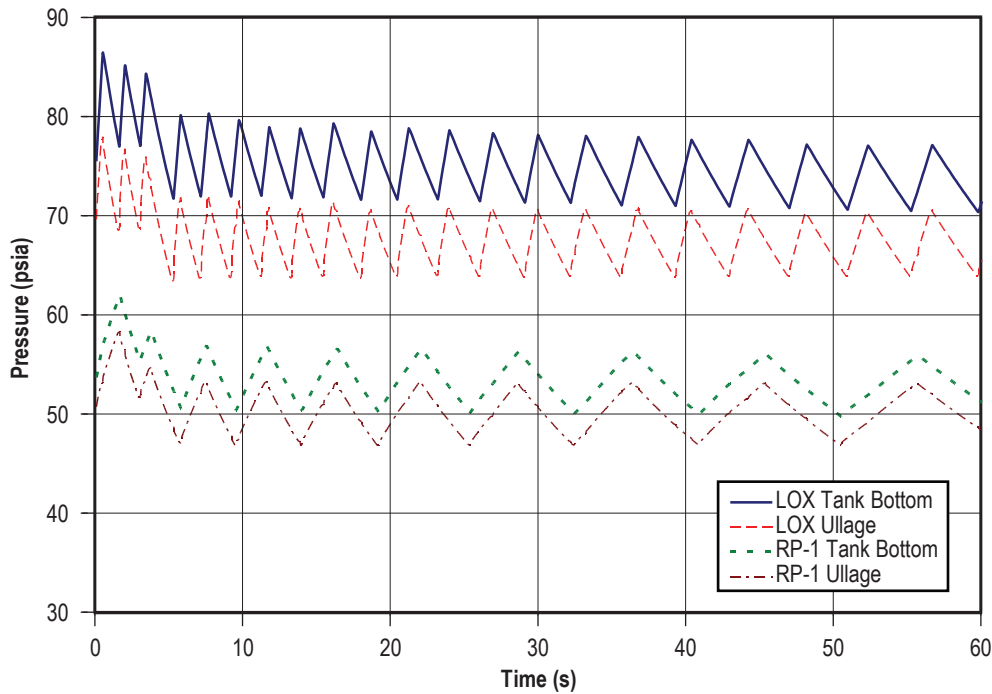
Figure 148. Propellant tank pressure history.

Figure 149 shows the predicted ullage temperature history in the RP-1 tank. Initially, wall and propellant temperatures were assumed equal at 70 °F. Heat transfer between ullage gas and the wall is not very significant in the RP-1 tank and, as a result, the tank wall temperature rises less than a degree during the 60-s engine operation. Ullage temperature, on the other hand, increases by about 42 °F due to mixing and pressurization. Ullage temperature diminishes during the period of valve closure because of the heat transfer from ullage gas to the wall.
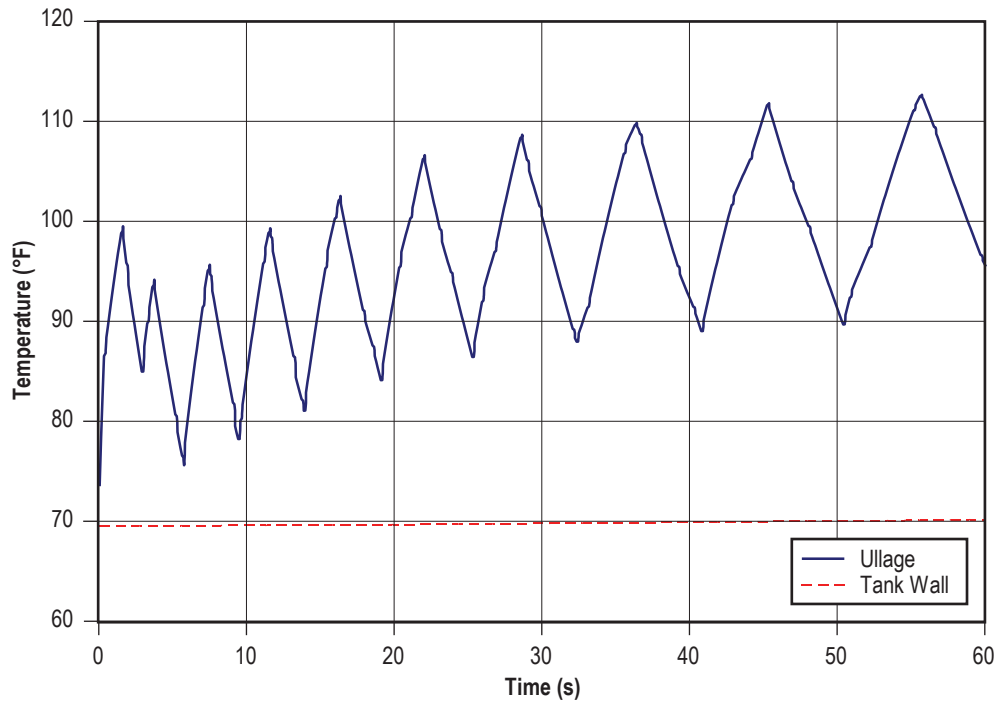
Figure 149. RP-1 temperature history.

Figure 150 shows the heat transfer history for the RP-1 tank. The ullage to propellant heat transfer rises mirrors the RP-1 ullage temperature behavior, reaching a peak value of 0.297 Btu/s. The ullage to wall heat transfer grows continuously throughout engine operation, achieving a maximum value of 1.78 Btu/s. This continuous rise is due to the ever-increasing tank wall area exposed to ullage gas as propellant is expelled from the tank. Conduction from the ullage exposed tank wall to the wetted wall is negligible compared to the heat transfer between the ullage and the wall.
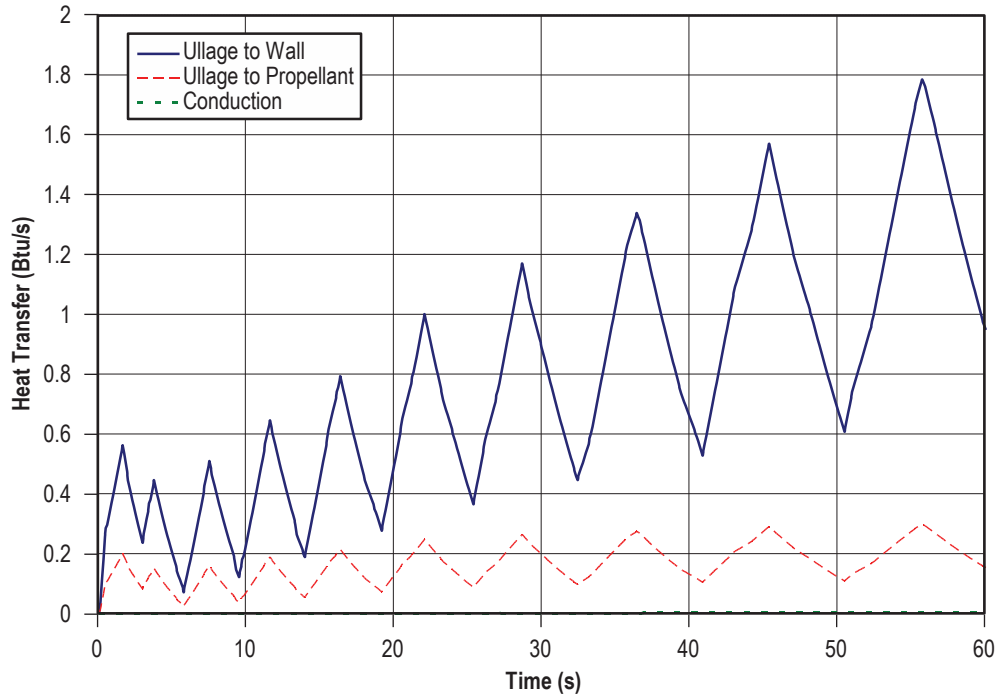
Figure 150. RP-1 heat transfer history.

The predicted ullage temperature history in the LOX tank is shown in figure 151. The LOX ullage temperature is assumed to be initially at –260 °F and the tank wall temperature is assumed to be initially at –300 °F. The tank wall temperature rise is more pronounced in the LOX tank than the RP-1 tank, rising 8 °F over the course of the 60-s run. The ullage temperature, on the other hand, rises about 147 °F. The higher temperature rise in the LOX tank is primarily due to the fact that the LOX ullage is initially assumed to be at –260 °F and mixes with helium at 120 °F. On the other hand, the initial temperature difference in the RP-1 ullage is much smaller. The other contributing factor is the higher helium flow rate into the LOX tank.
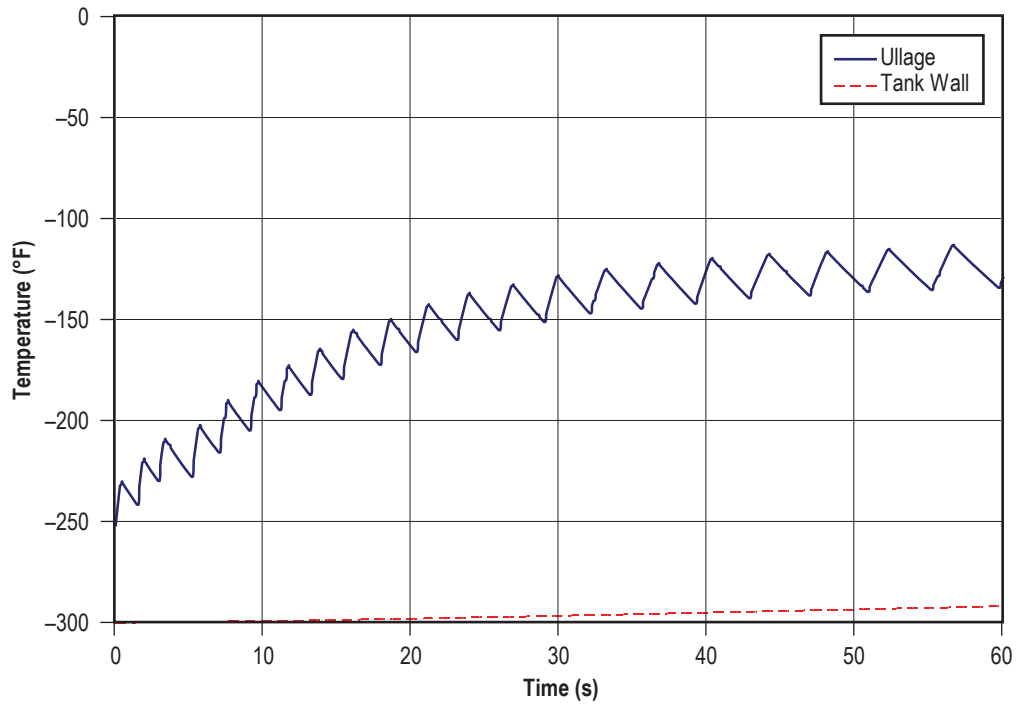
Figure 151. LOX temperature history.

Figure 152 shows the LOX tank heat transfer history. The LOX heat transfer curves follow a similar pattern to the RP-1 tank heat transfer curves, but on a much greater scale. The ullage to propellant heat transfer achieves a maximum value of 2.72 Btu/s and ullage to wall heat transfer peaks at 22.2 Btu/s.
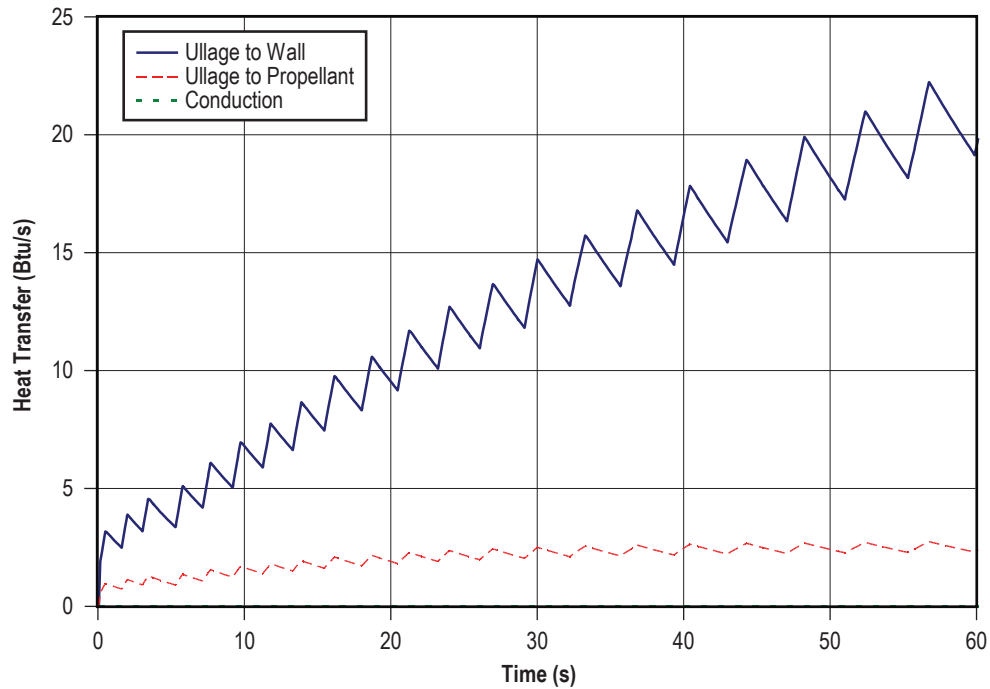
Figure 152. LOX tank heat transfer history.

The mass transfer history of propellant into the ullage for the LOX and RP-1 tanks is shown in figure 153. The mass transfer of propellant to ullage was calculated using the User Sub-routines discussed in section 6.10 (example 10). The mass transfer rate of GOX into the LOX tank ullage is much larger than that of vaporized RP-1 into the RP-1 tank ullage due to the higher heat transfer rates seen in the LOX tank. At the end of the 60-s run, the mass concentration of GOX in the LOX tank ullage is 0.14, while the mass concentration of vaporized RP-1 in the RP-1 tank ullage is 0.002.
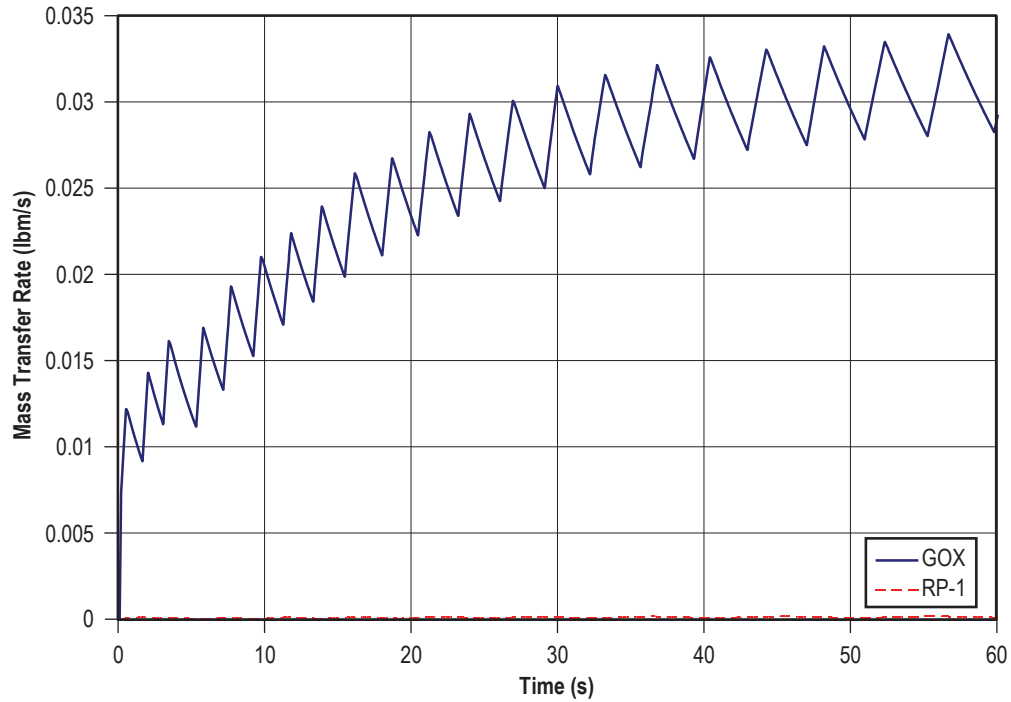
Figure 153. Mass transfer history of propellant.

The propellant flow rates are shown in figure 154 to be 139 lbm/s for LOX and 64 lbm/s for RP-1. The constant propellant flow rate predictions are a result of the RP-1 and LOX pumps. The branches upstream and downstream of the pumps have been adjusted to reproduce the pressure drops associated with the flow paths between the tanks and pump inlets and pump exits and engine chamber. This was done because of a lack of detailed flow path geometry downstream of the propellant tanks.
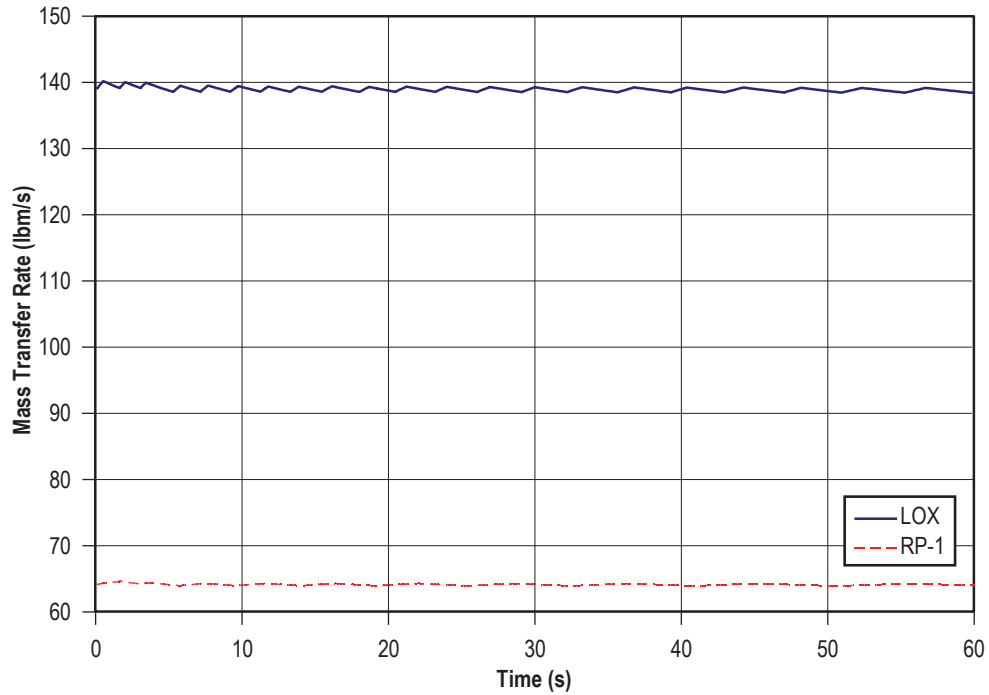
Figure 154. Propellant flow rate history.

While propellant is discharged to the engine, ullage volume increases. The increase in the ullage volume in the RP-1 and LOX tanks is shown in figure 155. The initial ullage volume of the RP-1 tank was assumed to be 15 ft^3 while the LOX tank initial ullage volume was assumed to be 25 ft^3. The ullage volumes increase linearly to 90 ft^3 and 141 ft^3 for the RP-1 and LOX tanks, respectively.
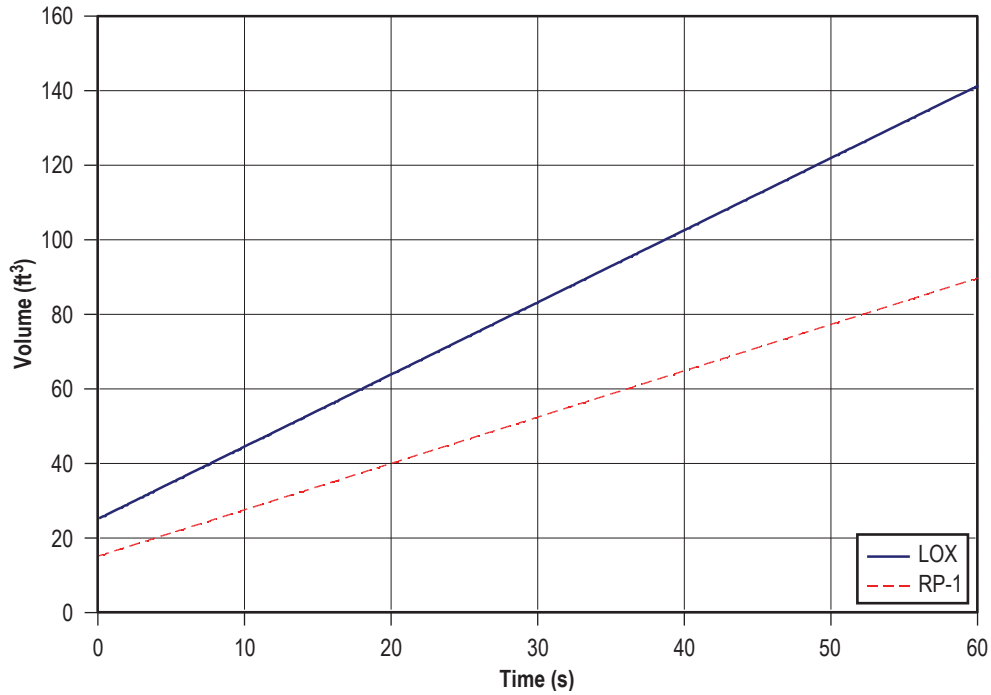
Figure 155. Ullage volume history in propellant tanks.

Figure 156 shows the helium flow rates in the system. Helium flow rate varies over time due to the opening and closing of the control valves during this time period. The flow from the facility interface is distributed to three branches. A nearly constant flow rate (about 0.4 lbm/s) is predicted to the engine purge interface for engine purges. The maximum flow rates to the LOX and RP-1 tanks are about 0.34 lbm/s and 0.085 lbm/s, respectively. Table 27 shows a comparison of GFSSP helium flow predictions with McRight's[47] pressurization analysis model.
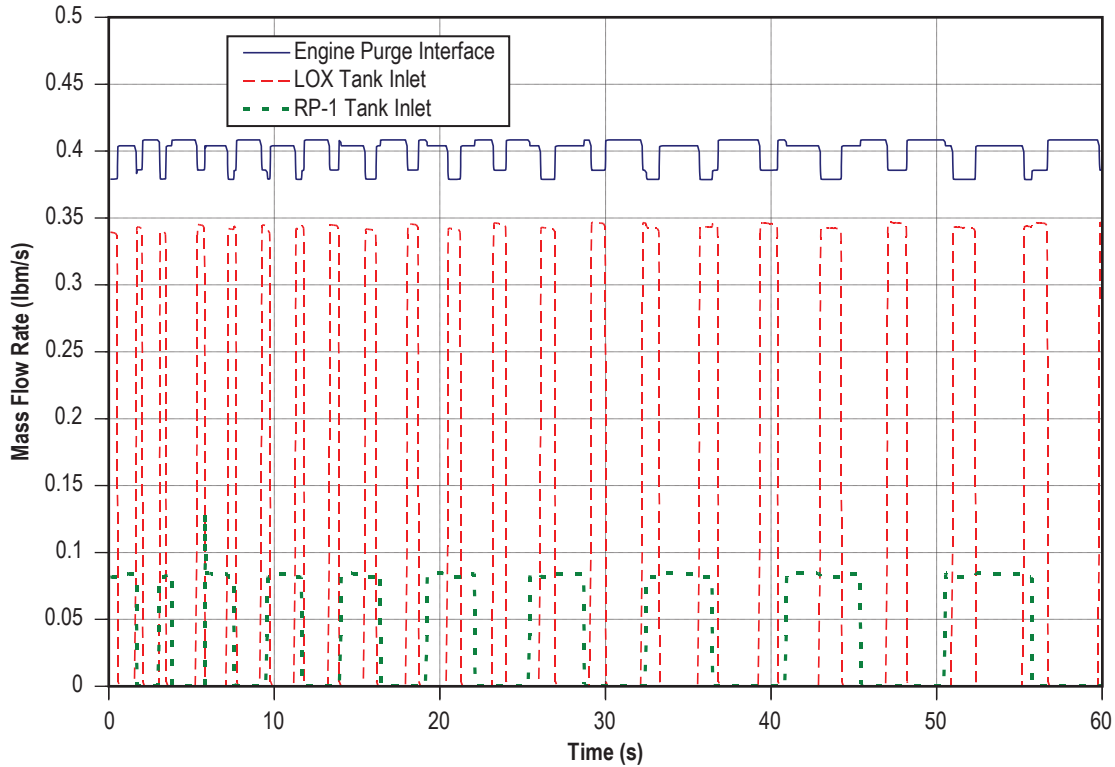
Figure 156. Helium flow rate history.

The comparison shown in table 27 appears reasonable considering the fact that McRight's analysis did not consider pressure loss in lines and fittings and choked flow rate through the orifice was calculated based on a facility pressure of 765 psia. GFSSP calculates pressure drop through the line; therefore, the choked flow rate at lower pressure is evidently less than McRight's prediction.

Table 27. Comparison between GFSSP and McRight's He flow rates.[47]

GFSSP (lbm/s)				McRight (lbm/s)			
Facility	LOX	RP-1	Purge	Facility	LOX	RP-1	Purge
0.825	0.34	0.085	0.4	1	0.35	0.1	0.55

6.13 Example 13—Steady State Conduction Through a Circular Rod, With Convection

6.13.1 Problem Considered

In the previous examples, the focus has been on GFSSP's fluid modeling capabilities. In this example, GFSSP's ability to model applications with conjugate heat transfer is introduced. The verification and validation of GFSSP's conjugate heat transfer capability was performed by

comparing with the known solution of a simple conduction-convection problem.[48] The heat transfer in a homogenous circular rod between two walls was considered (fig. 157). The two walls are held at temperatures of 32 and 212 °F,, respectively. The 0.167-ft-diameter rod is 2 ft in length and is initially at a temperature of 70 °F. The heat transfer coefficient between the rod and the ambient air is 1.14 Btu/ft^2-hr-R and the thermal conductivity of the rod is 9.4 Btu/ft-hr-R.
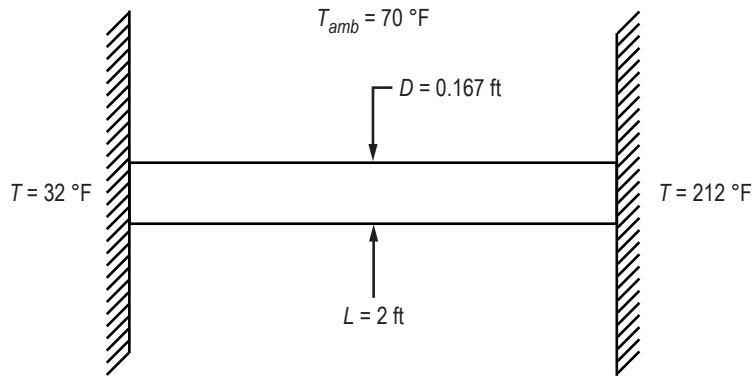
Figure 157. Schematic of circular rod connected to walls at different temperatures.

6.13.2 GFSSP Model

Figure 158(a) shows a GFSSP schematic of the system described by figure 157. The circular rod is represented by eight solid nodes and seven solid-solid conductors. Even though all of the material properties are not used for a steady state model, GFSSP still requires that placeholder values be input at each solid node. The thermal conductivity and specific heat temperature history files, which are shown below in an annotated form, are based on a user-defined material.

(a)

211
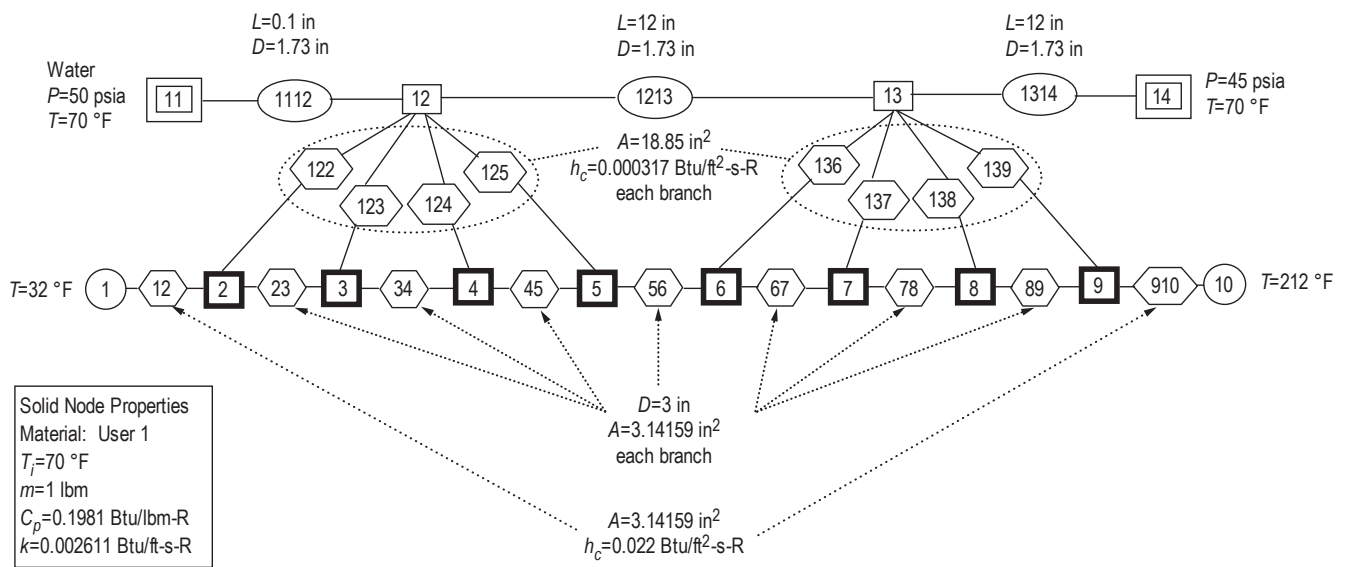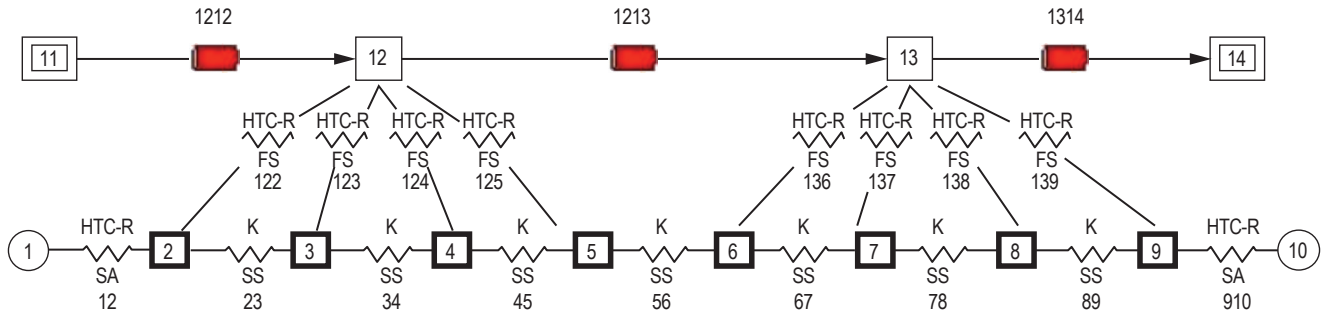
(b)

Figure 158. GFSSP model of circular rod for example 13: (a) Detailed model schematic
and (b) VTASC model.

```
USER1K.PRP

2 – Number of lines of data
Temperature (R)    Thermal Conductivity (Btu/ft-sec-R)
0                  0.002611
1000               0.002611

USER1CP.PRP

2 – Number of lines of data
Temperature (R)    Specific Heat (Btu/lbm-R)
0                  0.1981
1000               0.1981
```

Two ambient nodes are used to model the hot and cold walls, and their interaction with the rod is modeled using two solid-ambient conductors. Because GFSSP is first and foremost a fluid analysis code, it is necessary to include a fluid flow path in any GFSSP model that is being developed. Therefore, a dummy flow circuit consisting of two boundary nodes, two internal nodes, and three pipe flow branches was used to represent the ambient environment of figure 157. The details of the flow path were arbitrarily chosen with the sole intent of maintaining a constant temperature of 70 °F at all points in the flow path to correctly simulate the ambient environment. The two internal flow nodes are connected to the solid rod by eight solid-fluid conductors that represent the heat transfer between the ambient and the rod in figure 157. Figure 158(b) shows how this model looks in VTASC.

6.13.3 Results

The input and output files including history and property files of example 13 have been attached in appendix S (on CD inside back cover).

6.13.3.1 Analytical Solution. From the Thermal Analysis Workbook,[4] the differential equation of energy transport is given by:

$$\frac{d^2T}{dx^2} - \frac{4h}{Dk}\left(T - T_\infty\right) = 0 \tag{104a}$$

with boundary conditions of $T(0)=32$ °F and $T(L)=212$ °F. The closed form of the solution is given by

$$T(x) = T_{amb} + 4.653e^{\sqrt{\frac{4h}{Dk}}x} - 42.65e^{-\sqrt{\frac{4h}{Dk}}x}. \tag{104b}$$

Figure 159 compares GFSSP's predicted temperature distribution along the rod with the closed form solution. The comparison shows very close agreement between GFSSP and the closed form solution.
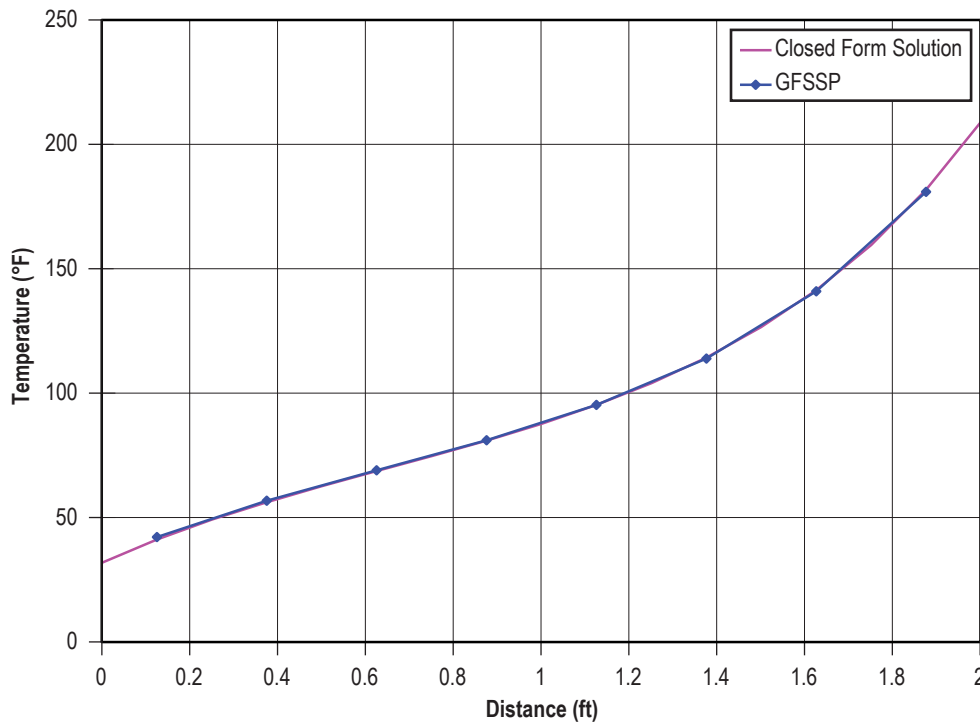
Figure 159. Comparison of GFSSP temperature prediction and closed form solution.

6.14 Example 14—Chilldown of a Short Cryogenic Pipeline

6.14.1 Problem Considered

For this example, the chilldown of a short cryogenic transfer line to validate GFSSP's transient conjugate heat transfer capability has been selected. An aluminum tube with a 26-in length with a 1/2-in outside diameter and a 3/16-in inside diameter is initially at a temperature of 80 °F. The tube is then chilled by liquid hydrogen entering the tube at –425 °F and 14.7 psia. The pressure at the outlet is set at 13.3 psia. Figure 160 shows a simple schematic of the problem.
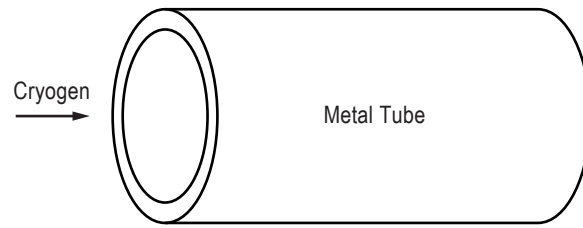
Figure 160. Short cryogenic tube schematic.

6.14.2 GFSSP Model

The tube shown in figure 160 was discretized into 30 nodes and 29 branches as shown in figure 161. Each branch is a pipe flow branch with a length of 0.897 in and a diameter of 0.1875 in. Nodes 1 and 30 are boundary nodes where inlet and outlet conditions were specified as defined in the problem description. Flow temperatures and pressures were calculated at internal nodes 2 through 29, which all have an initial pressure of 14 psi and an initial temperature of 60 °F. Each internal node was connected to a solid node (nodes 31 through 58) by a solid-fluid conductor. The solid nodes were also connected to each other using solid-solid conductors. Each solid node has a mass of 0.0151 lbm, an initial temperature of 80 °F, and has user-defined thermal conductivity and specific heat values (0.002611 Btu/ft-s-R and 0.161 Btu/lbm-R, respectively). Each solid-fluid conductor has a heat transfer area of 0.5283 in^2, while the heat transfer coefficients are calculated using the GFSSP's Miropolskii equation. The conduction area and distance for each solid-solid conductor are 0.1687 in^2 and 0.897 in, respectively.
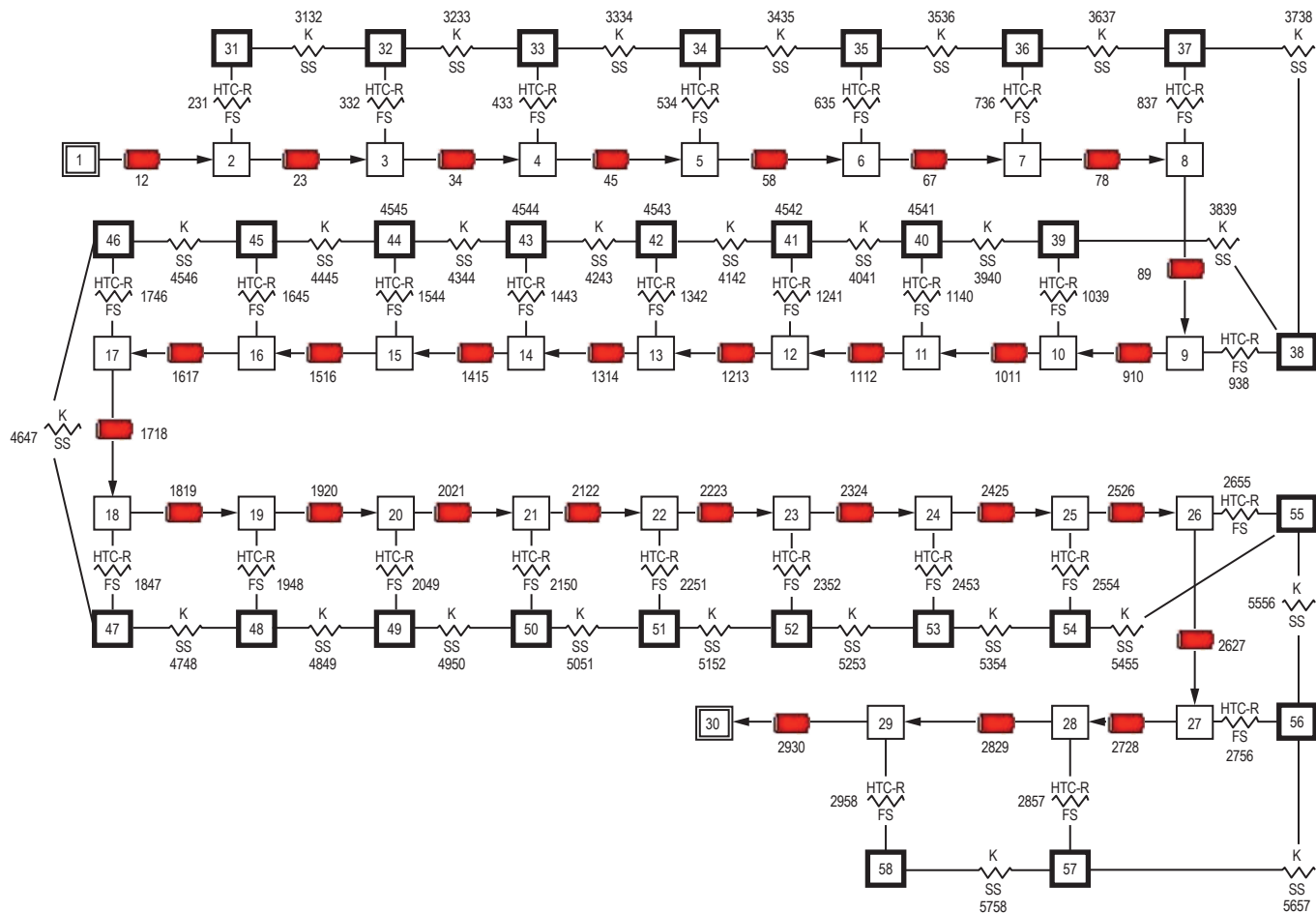
Figure 161. GFSSP model of short cryogenic tube.

6.14.3 Results

The input and output files including history and property files of example 14 have been attached in appendix T (on CD inside back cover). This model was originally created and compared with an analytical solution by Cross et al.[33] Figure 162 shows their original results, which shows excellent comparison between GFSSP and the analytical solution.
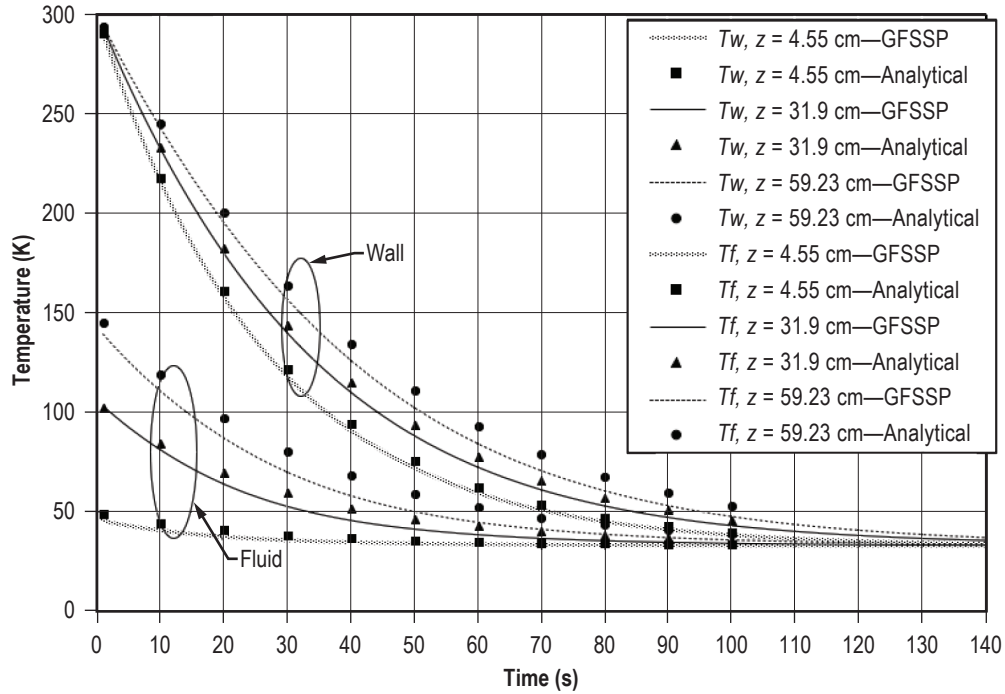
Figure 162. Comparison of GFSSP and analytical solution.[33]

6.15 Example 15—Simulation of Fluid Transient Following Sudden Valve Closure

6.15.1 Problem Considered

This example takes advantage of GFSSP's capability to model fluid transients. Fluid transients, also known as water hammer, can have a significant impact on the design and operation of both spacecraft and launch vehicle propulsion systems. These transients often occur at system activation and shutdown, and they must be predicted accurately to ensure the structural integrity of the propulsion system fluid network.

Consider the system shown in figure 163. LOX at 500 psia and 200 °R flows through a 400-ft-long, 0.25-in inside diameter pipeline at a mass flow rate of 0.1 lbm/s. The corresponding downstream pressure is 450 psia. At time zero, a valve at the end of the pipe begins a 100 ms rapid closure. This example discusses how to predict the liquid's response to the sudden valve closure, including the maximum expected surge pressure in the line.

Figure 163. Schematic of a propellant tank, pipeline, and valve.

6.15.2 GFSSP Model

The GFSSP model of the propellant tank and pipeline schematic of figure 163 is shown in figure 164. The system is represented by seven nodes and six branches. Node 1 represents the propellant tank as a boundary node. Node 7 represents the downstream pressure as a boundary node. Nodes 2 to 6 are internal nodes where pressure, temperature, and density are calculated. Branches 12 to 56 represent pipe segments of 80 ft length and 0.25 in diameter. Branch 67 represents the valve as a flow through a restriction with a flow area of 0.0491 in^2 and a flow coefficient of 0.6. In addition, GFSSP's unsteady valve open/close option is used to model the rapid valve closure in branch 67. The Valve Open/Close characteristics are defined in the Valve Open/Close dialog shown in figure 165. The Valve Open/Close characteristic file is also shown below with annotations to explain the meanings of each value. GFSSP's restart option is used to set the initial conditions of the model. The model is run as a steady state model, and the solution is saved in files FNDEX15.DAT and FBREX15.DAT. This solution is then read in as the initial solution for the transient model run.
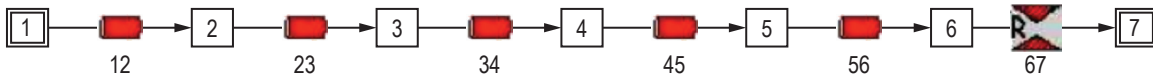
Figure 164. GFSSP model of a propellant tank, pipeline, and valve.
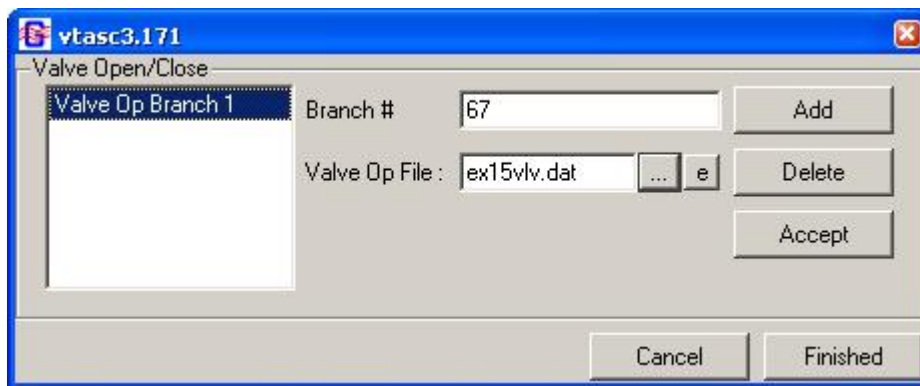
Figure 165. Example 15 Valve Open/Close dialog.

```
EX15VLV.DAT

7 - Number of lines of data
Time (s)    Flow Area (in²)
0.00        0.0491
0.02        0.0164
0.04        0.00545
0.06        0.00182
0.08        0.00061
0.1         1.E-16
100         1.E-16
```

Some consideration should be given to the time step for this model. In order to properly model the fluid transient, a time step must be chosen that is small enough to accommodate the model discretization. This is done by calculating the Courant number, which is the period of oscillation of one branch divided by the time step (eq. (105)). For the model to properly capture the fluid transient phenomena, the Courant number should be greater than unity:

$$\text{Courant Number} = \frac{4L_{branch}}{a_{fluid}\Delta\tau} \geq 1 \ . \tag{105}$$

The speed of sound (a_{fluid}) for LOX is 2,462 ft/s. Choosing a time step of 0.02 s gives a Courant number of 6.5, which satisfies the criteria for this model.

6.15.3 Results

The input and output files including history and restart files of example 15 have been attached in appendix U (on CD inside back cover). Figure 166 compares GFSSP's predicted pressure at node 6 with a method of characteristics (MOC) solution. Both solutions compare very well in the timing and character of the predicted pressure oscillations. The maximum surge pressure predicted by GFSSP is 624 psia compared to a MOC prediction of 636 psia. Additional studies have been performed using varying levels of discretization, different fluids, and even liquid-gas mixtures.[36] A sudden valve opening situation has been modeled and compared with test data.[35]
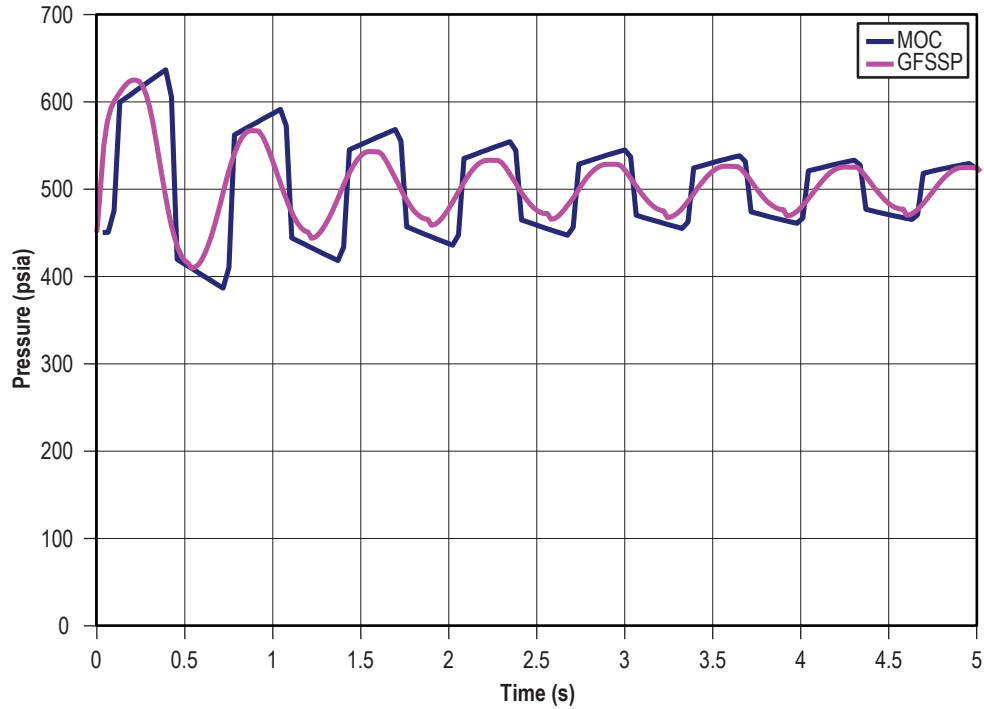
Figure 166. Comparison of GFSSP and MOC predicted pressure oscillations.

6.16 Example 16—Simulation of a Pressure Regulator Downstream of a Pressurized Tank

6.16.1 Problem Considered

This example repeats the blowdown problem of example 8, but simulates the operation of a pressure regulator downstream of the pressurized tank. Consider a tank (fig. 167) with an internal volume of 10 ft^3, containing air initially at a pressure and temperature of 100 psia and 80 °F, respectively. Air passes through a pressure regulator and is then discharged into the atmosphere through an orifice of 0.1 in diameter for a period of 150 s. GFSSP is used to determine the mass flow rate and pressure/temperature history of the tank contents during blowdown. The detailed GFSSP model schematic is shown in figure 168 while the VTASC model is shown in figure 169.
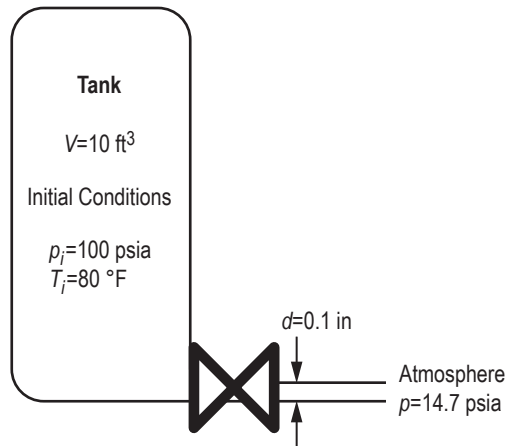
Figure 167. Tank blowdown with pressure regulator VTASC model (example 16).
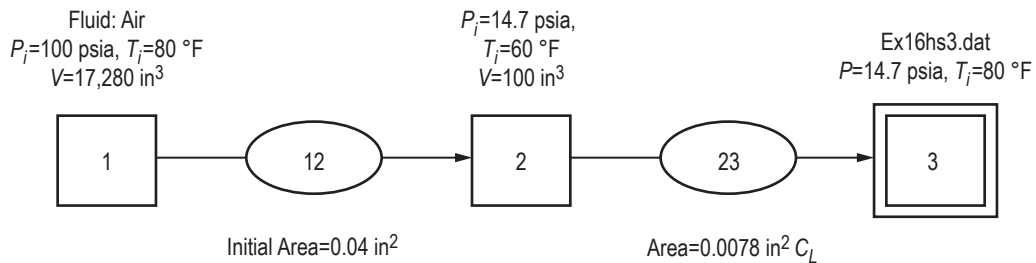
Figure 168. Tank blowdown with pressure regulator detailed model (example 16).
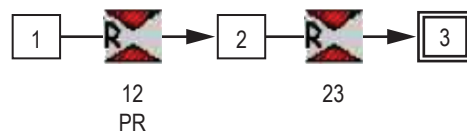
Figure 169. Tank blowdown with pressure regulator VTASC model (example 16).

6.16.2 GFSSP Model

The process is modeled with three nodes and two branches. Node 1 is an internal node that represents the tank volume, with initial pressure of 100 psia and initial temperature of 80 °F. Node 2 represents the small volume between the regulator and the outlet orifice. Node 3 represents the ambient boundary with constant pressure of 14.7 psia. The history file, ex16hs3.dat, is the same as the history file used for example 8. Resistance option 2 (Restriction) was used for branches 12 and 23. Branch 12 represents the pressure regulator with an initial flow area of 0.04 in^2. Branch 23 is the outlet orifice with a flow area of 0.00785 in^2.

The pressure regulator option is activated on the Unsteady Options page, and the dialog box is found on the Advanced menu. The option should be applied to Restriction or Compressible Orifice branches. There are two options for modeling a pressure regulator:

- Option 1 is an iterative algorithm that continuously adjusts the flow area of the branch in each time step, not leaving the time step until the desired downstream pressure is achieved (within the user-specified tolerance level), or until the maximum number of iterations is reached. Only one regulator per model is allowed. This option acts like a regulator that reacts instantly to changes in the flow network. Because it can take many iterations to find the proper area to achieve the desired pressure in each time step, this option may run slowly.

- Option 2 is a time-marching algorithm in which the regulator area is adjusted just once at the beginning of each time step, based on an empirical correlation that compares the downstream pressure from the previous time step to the desired pressure.[38] The pressure may oscillate about the set point for several time steps until convergence is reached. Multiple regulators are allowed with this option. As it takes a finite amount of time to reach the desired pressure, this option can more realistically model the behavior of a real pressure regulator. The time to achieve convergence can be adjusted by modifying the initial regulator area and the relaxation parameter. Because the flow area is corrected just once per time step, this option will generally run more quickly than option 1.

The pressure regulator history file for branch 12 is shown in table 28. It calls for a downstream pressure of 35 psia for the first 10 s, followed by 40 psia thereafter. The user can also set a constant pressure in the dialog box. Air is modeled using the ideal gas option.

Table 28. Pressure regulator history file for branch 12 (example 16).

Number of Data Points—Four		
τ (s)	p (psia)	Note: this line does not appear in dat file
–	35	
10	35	
10.01	40	
1,000	40	

6.16.3 Results

The input and output files of this example are included in appendix V as ex16.dat and ex16. out (on CD inside back cover). The transient test model was run for 150 s with 0.1-s time step, using both pressure regulator options. Calculated tank and downstream pressure, tank temperature, pressure regulator flow area, and flow rate are shown in figures 170–174. Note the gradually shrinking oscillations about the desired pressure when option 2 is used, and how option 1 reacts instantly to the change in set pressure at 10 s.
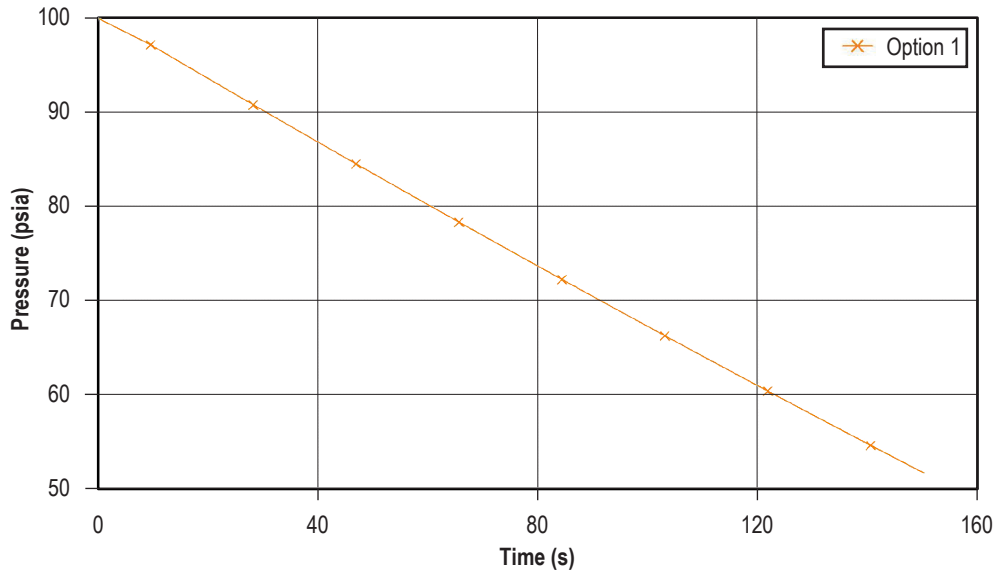
Figure 170. Tank pressure history (example 16).
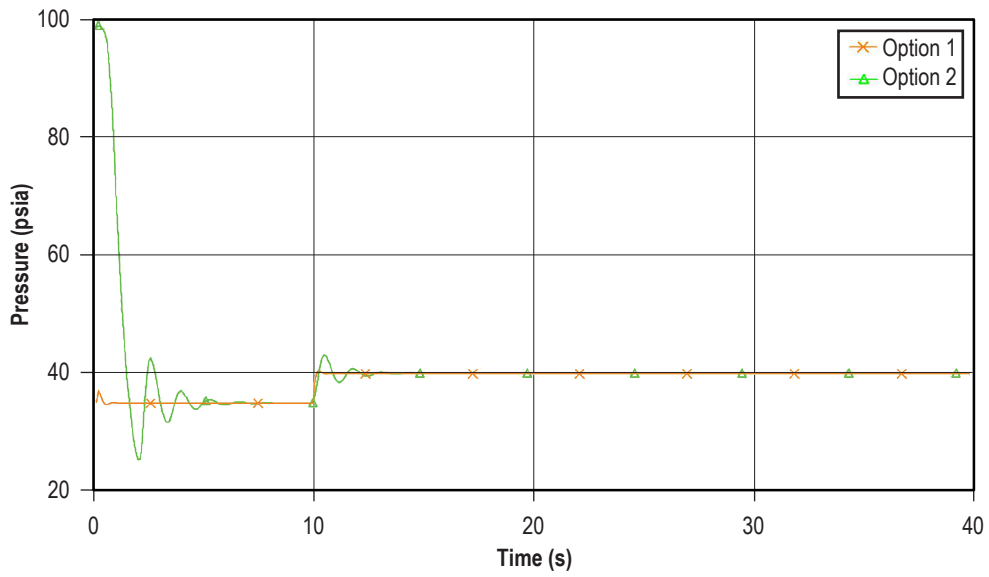
Figure 171. Pressure history downstream of pressure regulator (example 16).
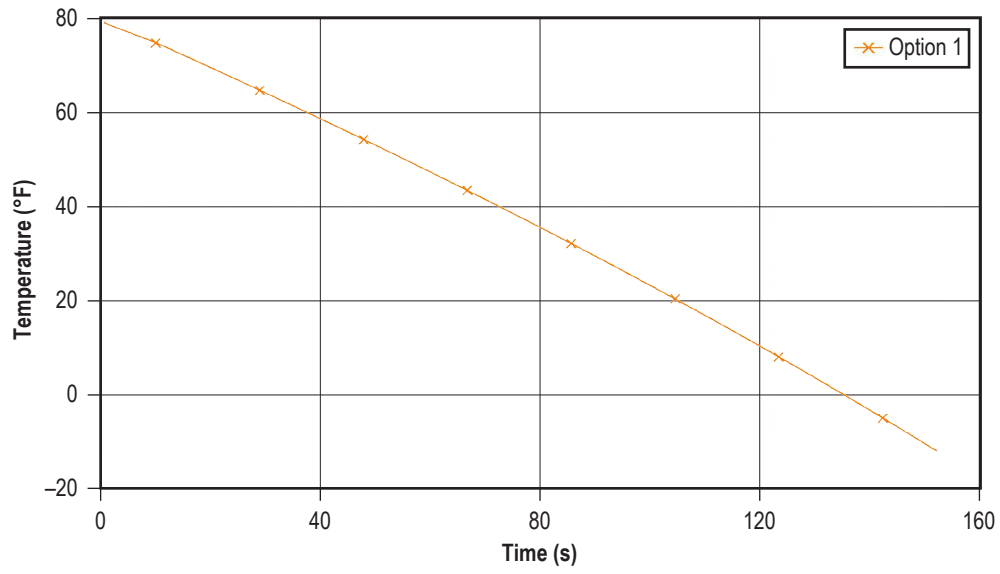
Figure 172. Tank temperature history (example 16).
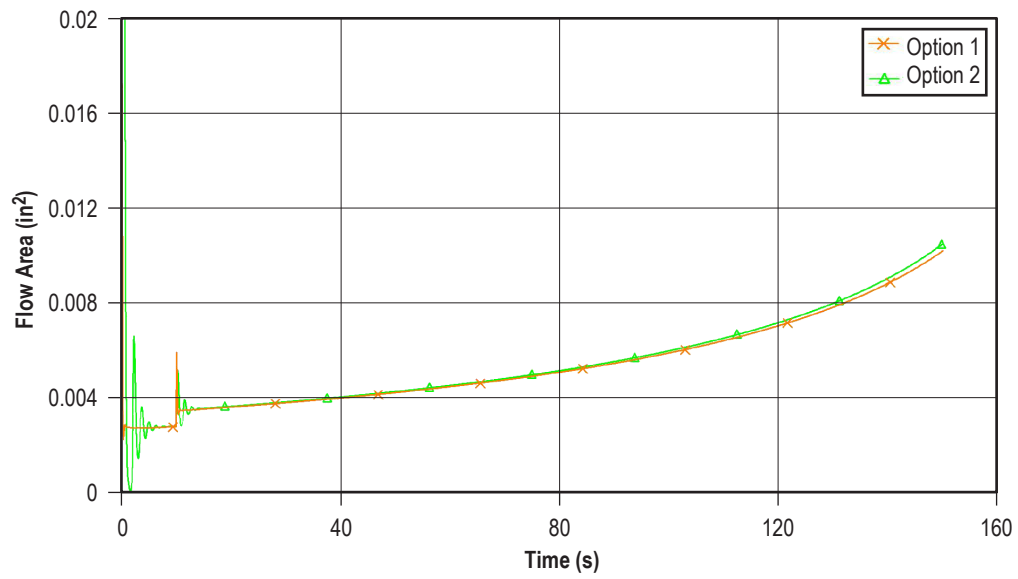
Figure 173. Pressure regulator flow area history (example 16, restrict 12).
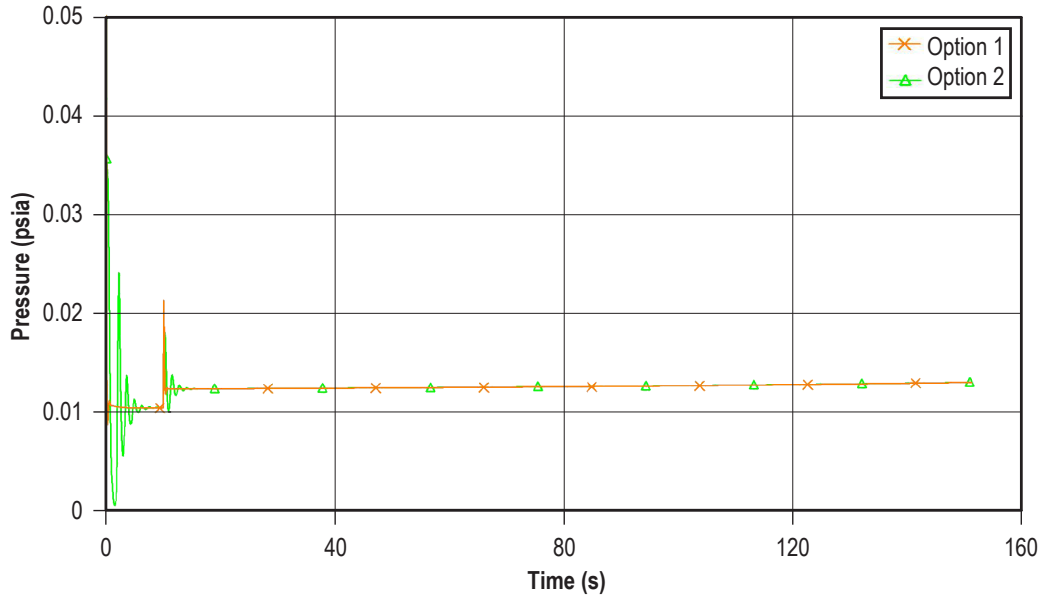
Figure 174. Pressure regulator flow rate history (example 16, restrict 12).

6.17 Example 17—Simulation of a Flow Regulator Downstream of a Pressurized Tank

6.17.1 Problem Considered

This example repeats the blowdown problem of example 8, but simulates the operation of a flow regulator downstream of the pressurized tank. Consider a tank (fig. 175) with an internal volume of 10 ft^3, containing air initially at a pressure and temperature of 100 psia and 80 °F, respectively. Air passes through a flow regulator and is then discharged into the atmosphere for a period of 20 s. GFSSP is used to determine the pressure/temperature history of the tank contents during blowdown. The detailed GFSSP model schematic is shown in figure 176 while the VTASC model is shown in figure 177.
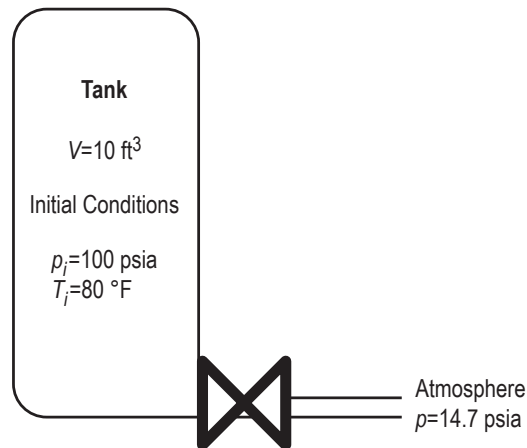
Figure 175. Tank blowdown with flow regulator schematic (example 17).

Fluid: Air
P_i=100 psia, T_i=80 °F
V=17,280 in³

Ex17hs3.dat
P=14.7 psia, T=80 °F

1 — 12 — 2

Initial Area=0.00785 in² C_L=1
Flow=0.012–0.02 lbm/s
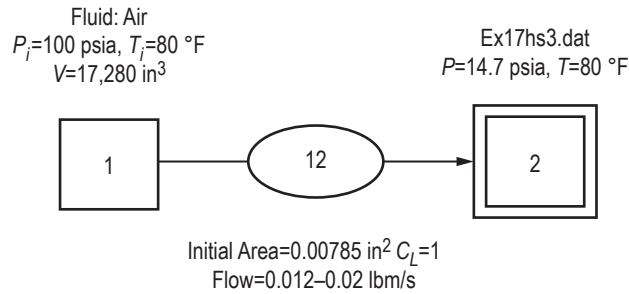
Figure 176. Tank blowdown with flow regulator detailed model (example 17).

1 — C.o 12 FR — 2
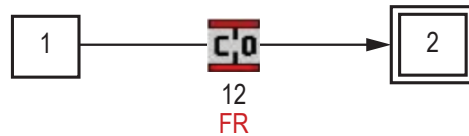
Figure 177. Tank blowdown with flow regulator VTASC model (example 17).

6.17.1 GFSSP Model

The process is modeled with two nodes and one branch. Node 1 is an internal node that represents the tank volume, with initial pressure of 100 psia and initial temperature of 80 °F. Node 2 represents the ambient boundary with a constant pressure of 14.7 psia. The history file, ex17hs3.dat, is the same as the history file used for example 8. Branch 12 represents the flow regulator, modeled with resistance option 22 (Compressible Orifice).

The flow regulator option is activated on the Unsteady Options page, and the dialog box is found on the Advanced menu. The option should be applied to Restriction or Compressible Orifice branches. There are two options for modeling a flow regulator:

• Option 1 is an iterative algorithm that continuously adjusts the flow area of the branch in each time step, not leaving the time step until the desired flow rate is achieved (within the user-specified tolerance level), or until the maximum number of iterations (currently hard-coded to 50) is reached. Only one regulator per model is allowed. This option acts like a regulator that reacts instantly to changes in the flow network. Because it can take many iterations to find the proper area to achieve the desired flow rate in each time step, this option may run slowly.

• Option 2 is a time-marching algorithm in which the regulator area is adjusted just once at the beginning of each time step, based on the numerical derivative of the change in flow rate with respect to the change in flow area, calculated from the results of the previous time steps. The flow rate may oscillate about the set point for several time steps until convergence is reached. Because other elements of the model may also affect the flow rate derivative, this option can be numerically unstable; under-relaxation may be required. Multiple regulators are allowed with this option. As it takes a finite amount of time to reach the desired flow rate, this option can

more realistically model the behavior of a real flow regulator. The time to achieve convergence can be adjusted by modifying the initial regulator area and the relaxation parameter. Because the flow area is corrected just once per time step, this option will generally run more quickly than option 1.

The flow regulator history file is shown in table 29. It calls for a flow rate 0.012 lb/s for the first 10 s, followed by 0.020 lb/s thereafter. The user can also set a constant flow rate in the dialog box. Air is modeled using the ideal gas option.

Table 29. Flow regulator history file for branch 12 (example 17).

Number of Data Points—Four		
τ (s)	$\dot{m}$ (lb/s)	Note: this line does not appear in dat file
1	0.012	
10	0.012	
10.01	0.02	
1,000	0.02	

6.17.3 Results

The input and output files of this example are included in appendix W as ex17.dat and ex17.out (on CD inside back cover). The transient test model was run for 20 s with 0.5-s time step, using both flow regulator options. Calculated tank pressure and temperature, flow regulator area, and flow rate are shown in figures 178–181. Note that option 2 takes several time steps to reach the target flow rate, while option 1 reacts instantly to the change in required flow rate at 10 s.
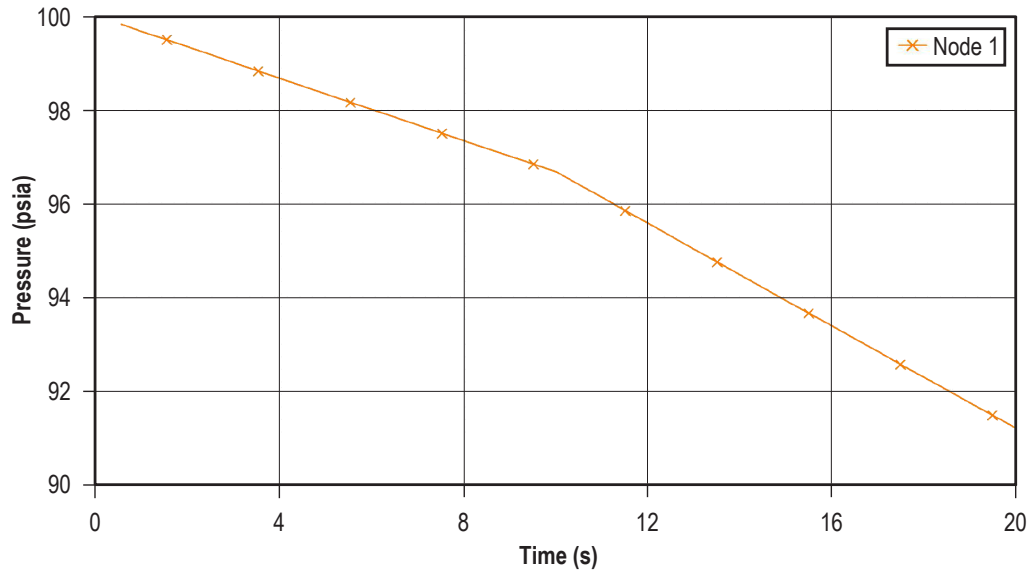
Figure 178. Tank pressure history (example 17).
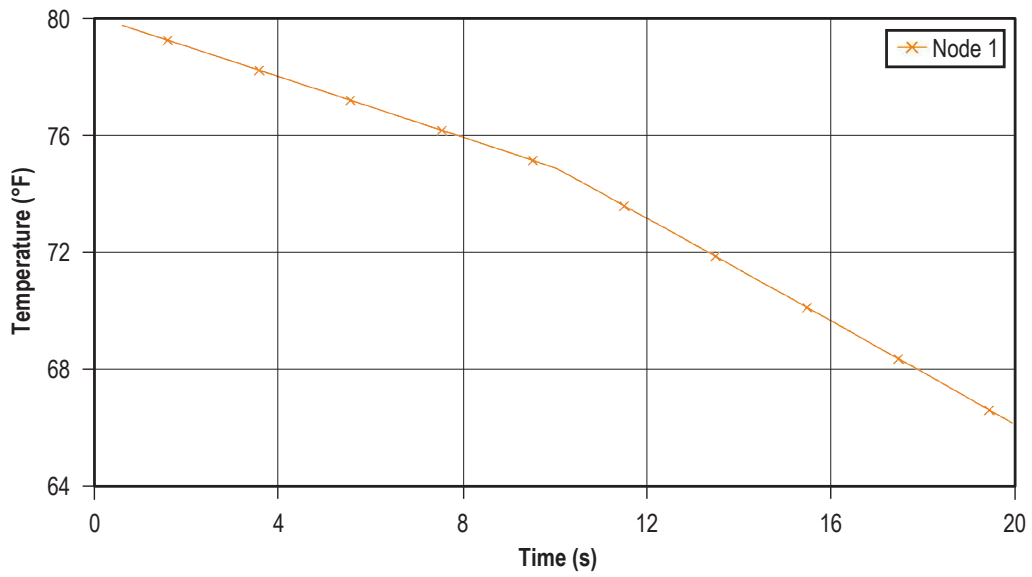
Figure 179. Tank temperature history (example 17).
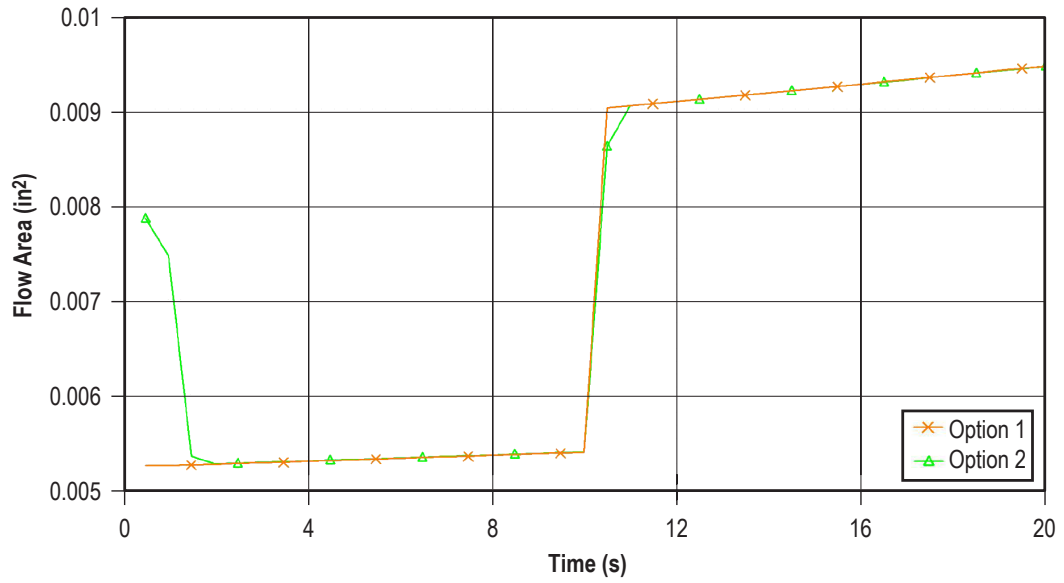
Figure 180. Flow regulator area history (example 17).
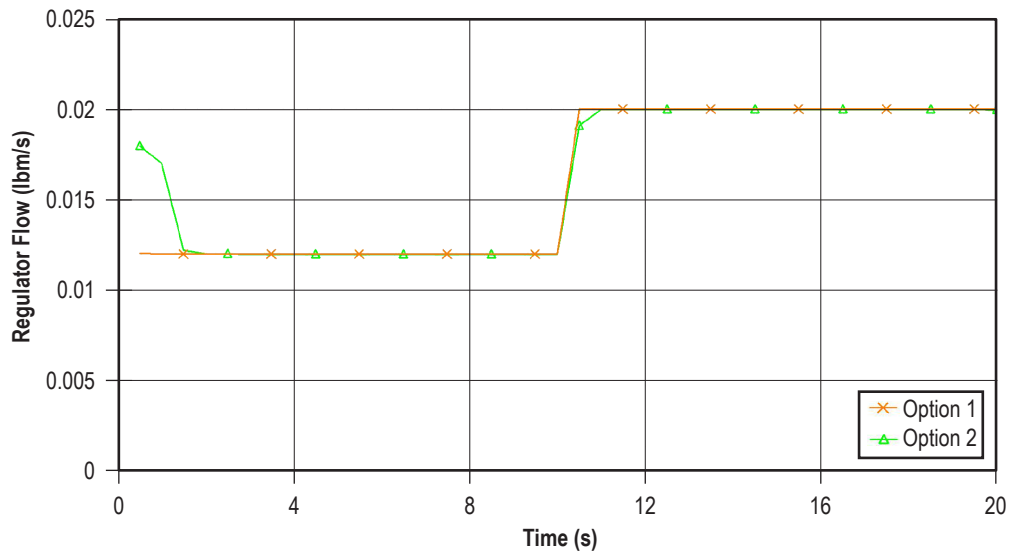
Figure 181. Flow regulator flow rate history (example 17).

228

6.18 Example 18—Simulation of a Subsonic Fanno Flow

6.18.1 Problem Considered

This example illustrates the capabilities of the steady state flow formulation of GFSSP to simulate subsonic Fanno flow. Fanno flow is the flow with friction in an adiabatic constant-area pipe. Consider a supply line (fig. 182) containing nitrogen gas at a pressure and temperature of 50 psia and 80 °F, respectively. Nitrogen gas is discharged into a pipe of 3,207 in length and 6 in diameter. GFSSP will be used to determine the pressure, mass flow rate, and temperature history through pipes using the Fanno flow process. These predicted values are then compared with the analytical solution.
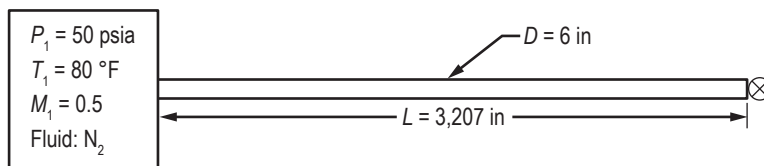
Figure 182. Fanno flow schematic (example 18).

6.18.2 GFSSP Model

GFSSP model schematic is shown in figure 183 and the VTASC model is shown in figure 184.
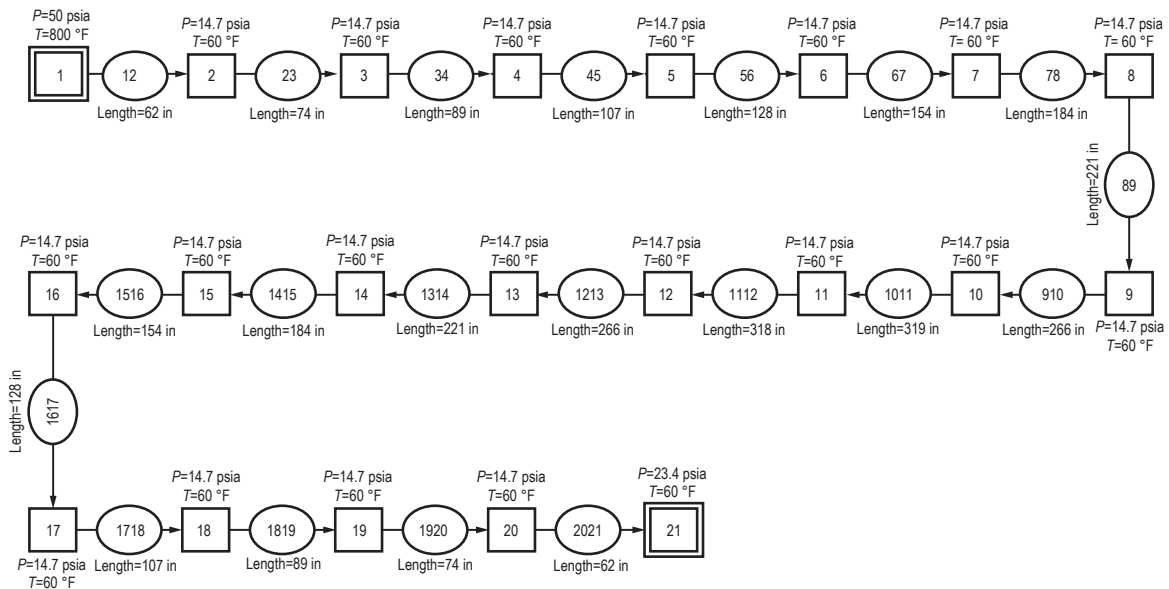
Figure 183. Fanno flow model schematic—fluid: nitrogen.

229
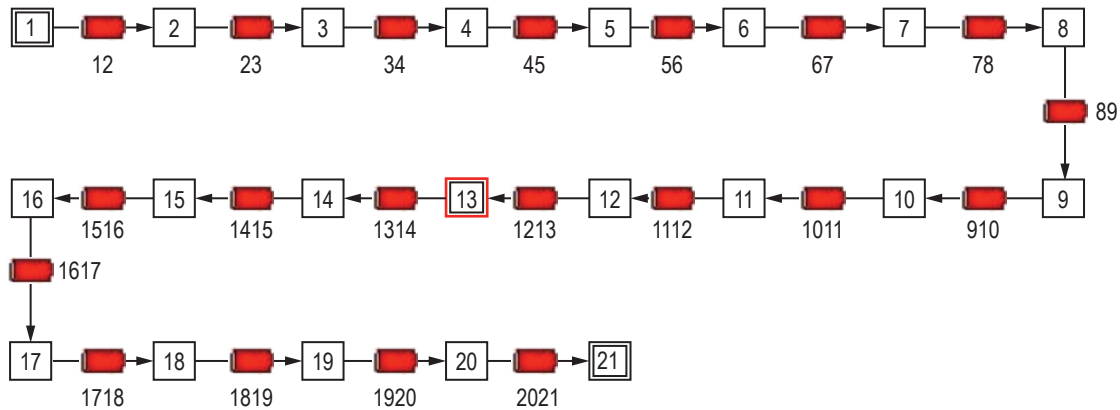
Figure 184. Fanno flow VTASC model.

The process has been modeled with 21 nodes and 20 branches. Node 1 is a boundary node that represents the supply line. For the steady state Fanno flow formulation, the node initial guess must be supplied for each internal node and conditions must be supplied for each boundary node. The initial pressure and temperature for boundary nodes 1 and 21 are shown in table 30. Resistance option 1 (pipe) was used for all branches with different lengths, diameter, and an absolute roughness as shown in table 31. Inertia was activated in all branches to account for fluid acceleration caused by density change. Nitrogen gas is modeled using the general fluid option that is available in VTASC. Table 32 shows the steady state results of pressure, temperature, and density in internal nodes.

Table 30. Example 18 boundary conditions.

Boundary Node No.	Pressure (psia)	Temperature (°F)
1	50	80
21	23.4	60

Table 31. Branch properties of Fanno flow (example 18).

Branch No.	Length (in)	Diameter (in)
12	62	6
23	74	6
34	89	6
45	107	6
56	128	6
67	154	6
78	184	6
89	221	6
910	266	6
1011	319	6
1112	318	6
1213	266	6
1314	221	6
1415	184	6
1516	154	6
1617	128	6
1718	107	6
1819	89	6
1920	74	6
2021	62	6

Table 32. Example 18 internal node results.

Internal Node No.	Pressure (psi)	Temperature (°F)	Density (lbm/ft³)
2	49.81	79.98	0.241
3	49.5	79.74	0.2396
4	49.11	79.39	0.2379
5	48.63	78.94	0.2358
6	48.05	78.37	0.2332
7	47.34	77.65	0.2301
8	46.46	76.73	0.2262
9	45.38	75.54	0.2214
10	44.04	73.97	0.2155
11	42.34	71.82	0.208
12	40.41	68.83	0.1997
13	38.45	65.07	0.1913
14	36.55	60.86	0.1834
15	34.72	56.29	0.1757
16	32.94	51.34	0.1684
17	31.19	45.9	0.1611
18	29.42	39.83	0.1539
19	27.6	32.85	0.1464
20	25.64	24.47	0.1384

This example validates GFSSP's prediction with the analytical solution. The analytical solution, discussed below, is valid for flows with heat transfer and friction. However, the friction factor must be constant throughout the length of the pipe. On the other hand, GFSSP calculates the friction factor with local properties such as density, viscosity, and velocity, and friction factor varies from branch to branch. Therefore, in order to obtain a GFSSP solution with constant friction factor, this model makes use of User Subroutine to set a constant friction factor in all branches. Subroutine KFADJUST was used to adjust K_f for a given friction factor. The analytical solution assumes a friction factor of 0.002. The listing of Subroutine KFADJUST in appendix X shows how to implement the constant friction factor in GFSSP (on CD inside back cover).

6.18.2.1 Analytical Solution of Fanno Flow. The analytical solution of compressible flow with friction requires simultaneous solution of the following mass and momentum conservation equations:

Mass Conservation:

$$\frac{d\rho}{\rho} + \frac{dA}{A} + \frac{dV}{V} = 0 \,.$$

(106)

Momentum Conservation:

$$\frac{dp}{p} + \frac{\gamma M^2}{2} + \frac{f dx}{D} + \gamma M^2 \frac{dV}{V} = 0 , \qquad (107)$$

where, M = Mach No. = $\dfrac{V}{C} = \dfrac{V}{\sqrt{\gamma \dfrac{p}{\rho}}}$.

Using the definition of Mach number, stagnation temperature, equationss (106) and (107) can be expressed as an ordinary differential equation of first order:

$$\frac{dM}{dx} = \frac{M\left(1 + \frac{\gamma-1}{2}M^2\right)}{\left(1 - M^2\right)}\left[\gamma M^2 \frac{f}{D} + \frac{\left(1 + \gamma M^2\right)}{2T_0}\frac{dT_0}{dx} - \gamma M^2 \frac{1}{A}\frac{dA}{dx}\right] , \qquad (108)$$

with boundary value,

$$M(x=0) = M_1 . \qquad (108a)$$

dT_0/dx in equation (108) can be determined from the energy equation which can be expressed as:

$$q(\pi D)dx = \dot{m}C_p dT_0 . \qquad (109)$$

Given the inlet conditions (eq. (108)), the first order differential equation (eq. (108)) in M is solved to find the Mach number at any x location. As this equation is a nonlinear equation in M, this equation is solved by using the fourth order Runge-Kutta method.[27]

From Mach number, the static temperature and pressure can be determined from the following equations:

$$\frac{T(x)}{T(0)} = \frac{T_0(x)}{T_0(0)}\frac{1 + \frac{\gamma-1}{2}(M(0))^2}{1 + \frac{\gamma-1}{2}(M(x))^2} \qquad (110)$$

and

$$\frac{p(x)}{p(0)} = \frac{A(0)}{A(x)}\frac{M(0)}{M(x)}\sqrt{\frac{T(x)}{T(0)}} . \qquad (111)$$

No heat transfer implies the stagnation temperature is constant and $dT_0/dx = 0$. Before presenting the results for Fanno flow, choice of pipe length as 3,207 in is explained below. From the analytical solution for Fanno flow, the critical length of the pipe (L^*) is determined from the following equation:

$$\frac{fL^*}{D} = \frac{1 - M^2}{\gamma M^2} + \frac{1 + \gamma}{2y} \ln\left(\frac{(1 + \gamma)M^2}{2 + (\gamma - 1)M^2}\right). \tag{112}$$

M is the inlet Mach number. The critical length of the pipe is the length required for the flow to choke at the exit (i.e., $M = 1$ at exit). With an inlet Mach number of 0.5 and a friction factor of 0.002, and pipe diameter of 6 in, the critical length is calculated to be 3,207 in. This length is kept fixed for the cases of constant area pipes.

6.18.3 Results

Figure 185 shows a plot of the p/p^* ratio with different types of node distribution for the numerical solution compared with the analytical solution. A nonuniform node distribution with a total of 21 nodes (20 control volumes) is sufficient to get a grid-independent solution. The corresponding temperature distribution is shown in figure 186. The plots also show that the numerical solution using GFSSP agrees very well with that of the analytical solution.
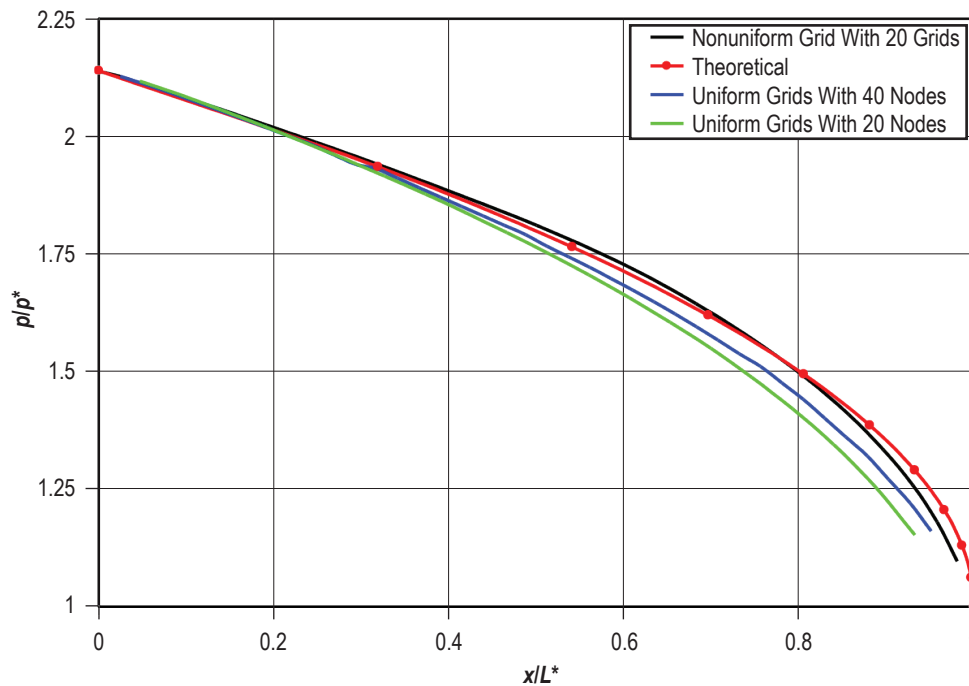
Figure 185. Pressure distribution for Fanno flow with various grid distributions.
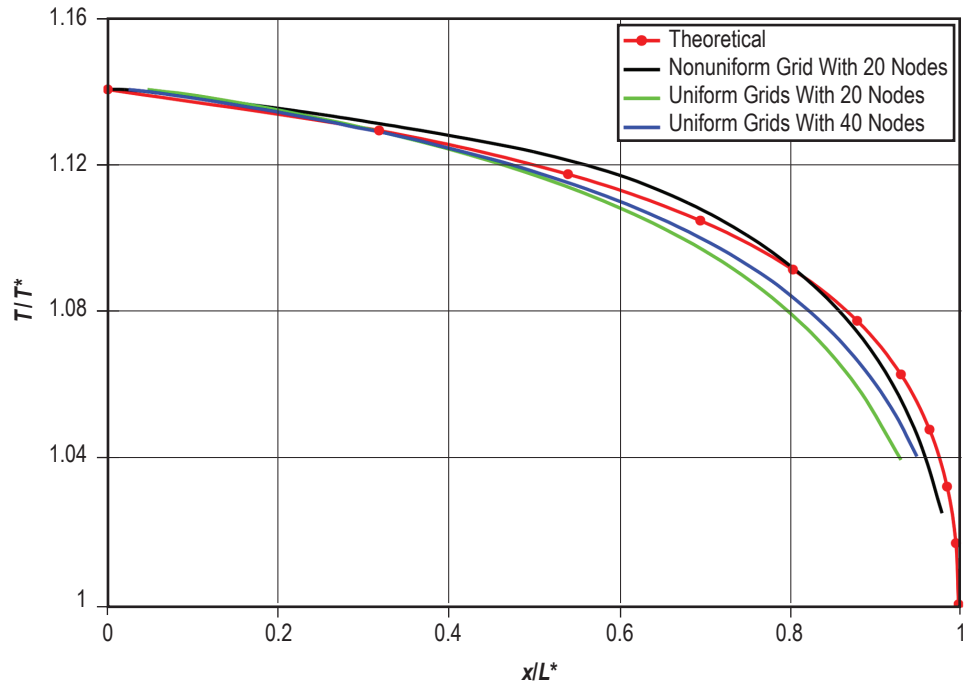
Figure 186. Temperature distribution for Fanno flow with various grid distributions.

Figure 187 shows a plot of the Mach number along the axial direction, and again the agreement with the analytical solution is quite good. The slight difference even at the inlet is because, in GFSSP, the mass flow rate is not prescribed, rather the pressure boundary condition is specified, and the flow rate is computed.
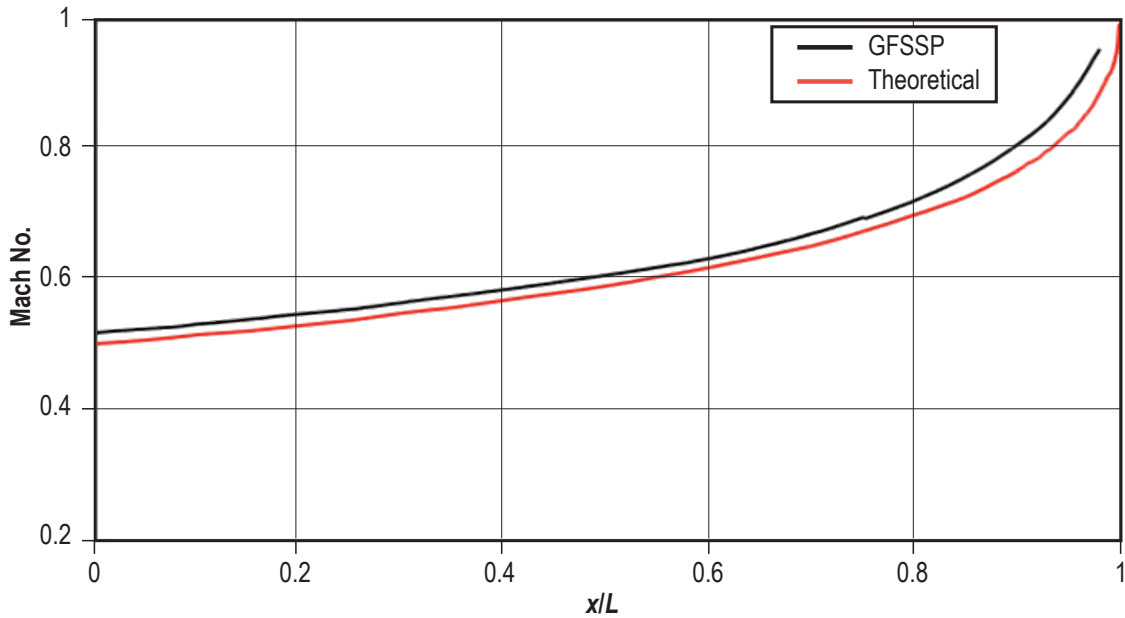
Figure 187. Fanno flow plot of Mach number along the pipe length.

6.19 Example 19—Simulation of a Rayleigh Flow

6.19.1 Problem Considered

This example illustrates the capabilities of the steady state flow formulation of GFSSP to simulate subsonic Rayleigh flow. Rayleigh flow is the frictionless flow in a constant area pipe with heat transfer. Consider a supply line (fig. 188) containing nitrogen gas at a pressure and temperature of 50 psia and 80 °F, respectively. Nitrogen gas is flowing with an inlet Mach number of 0.5 through a pipe that is 3,207 in long and has a 6 in diameter. A heat input rate of 1,800 Btu/s has been applied along the length of the discharge pipe. GFSSP will be used to determine the pressure, mass flow rate, and temperature history through pipes using the Rayleigh flow process. These predicted values are then compared with the analytical solution. The detailed model schematic is shown in figure 189 while the VTASC model is shown in figure 190.
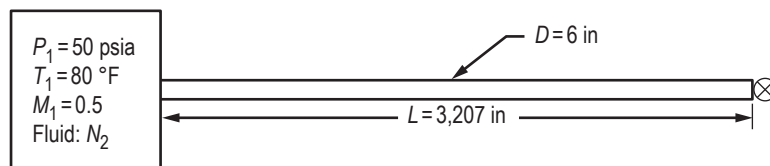
Figure 188. Tank blowdown schematic (example 19).

6.19.2 GFSSP Model

The GFSSP model schematic is shown in figure 189 and the VTASC model is shown in figure 190. The process has been modeled with 21 nodes and 20 branches. Node 1 is a boundary node that represents the supply line. For the steady state Rayleigh flow formulation, the node heat rate and the initial guess must be supplied for each internal node and conditions must be supplied for each boundary node. The initial pressure and temperature for boundary nodes 1 and 21 are shown in table 33. The initial guess pressure, temperature, and heat rate for internal nodes 2 through 20 are shown in table 34. Resistance option 1 was used for all branches with different lengths, heat rate, and an absolute roughness as shown in table 35. However, to simulate frictionless flow the friction factor is set to zero in the User Subroutine. Inertia was activated in all branches to account for fluid acceleration caused due to density change. Nitrogen gas is modeled using the general fluid option that is available in VTASC. Table 36 shows the steady state results of the internal node.
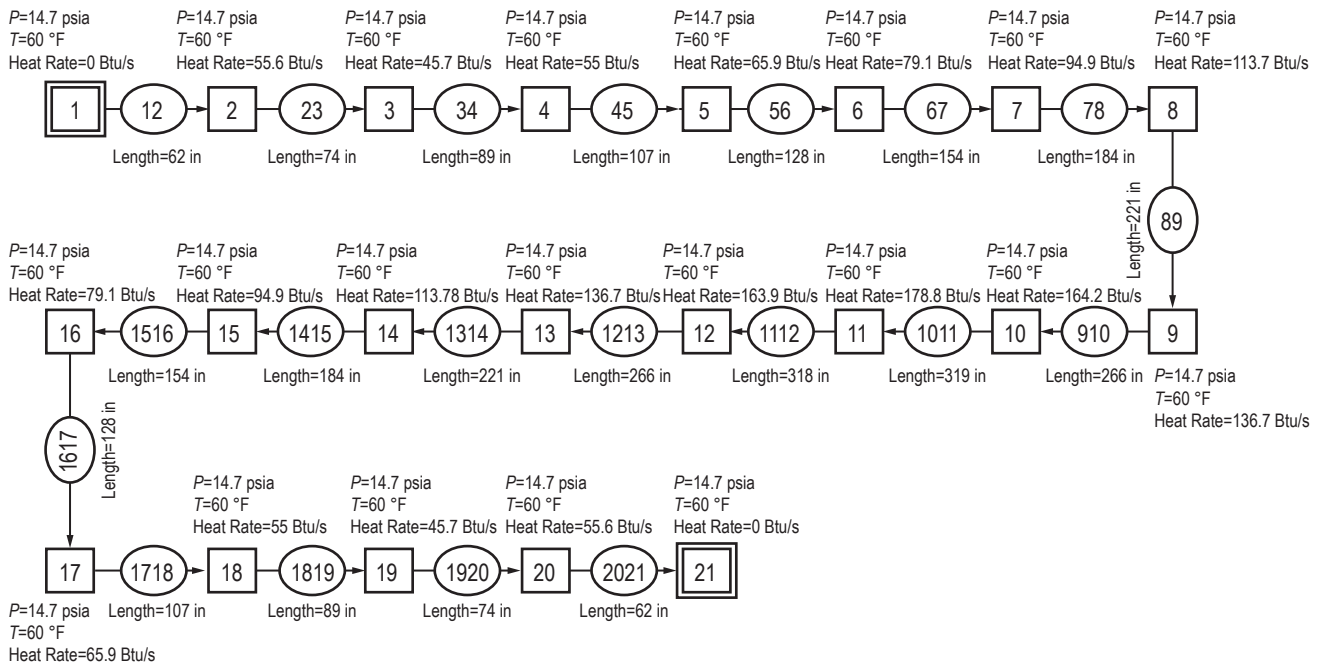
Figure 189. Pressurized tank detailed model schematic (example 19)—fluid: nitrogen.
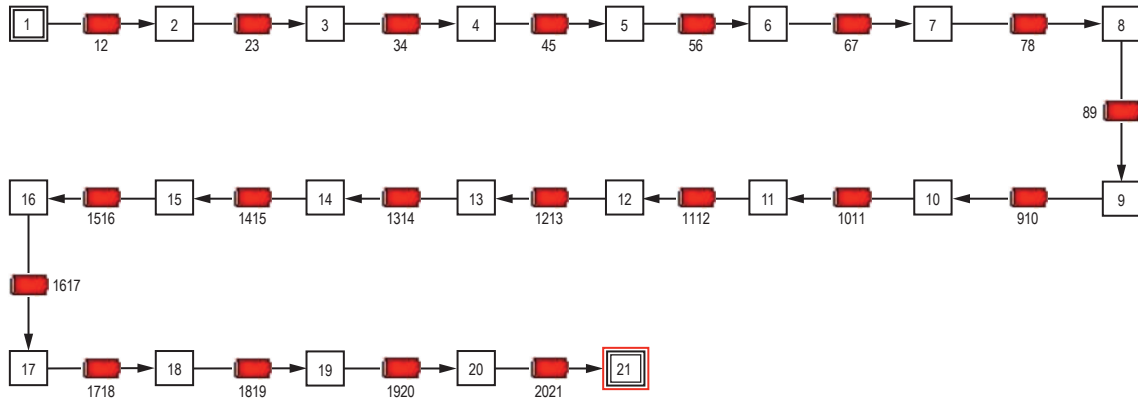
Figure 190. Rayleigh flow VTASC model (example 19).

Table 33. Boundary conditions (example 19).

Boundary Node No.	Pressure (psia)	Temperature (°F)
1	50	80
21	35	40

Table 34. Internal node initial conditions (example 19).

Internal Node No.	Pressure (psia)	Temperature (°F)	Heat Rate (Btu/s)
2	14.7	60	55.6
3	14.7	60	45.7
4	14.7	60	55
5	14.7	60	65.9
6	14.7	60	79.1
7	14.7	60	94.9
8	14.7	60	113.7
9	14.7	60	136.7
10	14.7	60	164.2
11	14.7	60	178.8
12	14.7	60	163.9
13	14.7	60	136.7
14	14.7	60	113.7
15	14.7	60	94.9
16	14.7	60	79.1
17	14.7	60	65.9
18	14.7	60	55
19	14.7	60	45.7
20	14.7	60	55.6

Table 35. Branch initial conditions (example 19).

Branch No.	Length (in)	Diameter (in)
12	62	6
23	74	6
34	89	6
45	107	6
56	128	6
67	154	6
78	184	6
89	221	6
910	266	6
1011	319	6
1112	318	6
1213	266	6
1314	221	6
1415	184	6
1516	154	6
1617	128	6
1718	107	6
1819	89	6
1920	74	6
2021	62	6

Table 36. Internal node results (example 19).

Internal Node No.	Pressure (psi)	Temperature (°F)	Density (lbm/ft³)
2	50	88.62	0.2381
3	49.75	94.94	0.2342
4	49.49	102.7	0.2297
5	49.18	111.9	0.2246
6	48.81	122.9	0.2187
7	48.35	136	0.2118
8	47.79	151.6	0.204
9	47.1	171.1	0.1951
10	46.25	192	0.1851
11	45.18	215	0.1746
12	43.92	234.8	0.1649
13	42.61	249.8	0.1566
14	41.36	261.4	0.1496
15	40.21	270.3	0.1437
16	39.16	277.1	0.1386
17	38.2	282.1	0.1343
18	37.32	285.7	0.1306
19	36.51	288.2	0.1273
20	35.78	292.6	0.124

6.19.3 Results

The main objective of this example is to validate GFSSP's prediction with the analytical solution described in section 6.18 in the context of Fanno flow. Equation (108) is valid for both heat transfer and friction. In order to obtain the solution for Rayleigh flow, friction factor in equation (108) is set to zero. Subroutine KFADJUST was used to set the zero friction factor (app. Y—on CD inside back cover).

Figures 191 and 192 show the distribution of temperature and Mach number for a heat input rate of 1,800 Btu/s on the same pipe geometry that has been used for Fanno flow (example 18). The inlet Mach number is chosen as 0.5 and it has been analytically calculated that with this inlet Mach number, a heat input of 1,800 Btu/s makes the flow choked at the exit (i.e., Mach number = 1). Further increase in the heat rate will make the flow supersonic and in the present work, only subsonic flow is being considered.[27] Again, the numerical results show quite good agreement (within 5%) with the analytical solution.
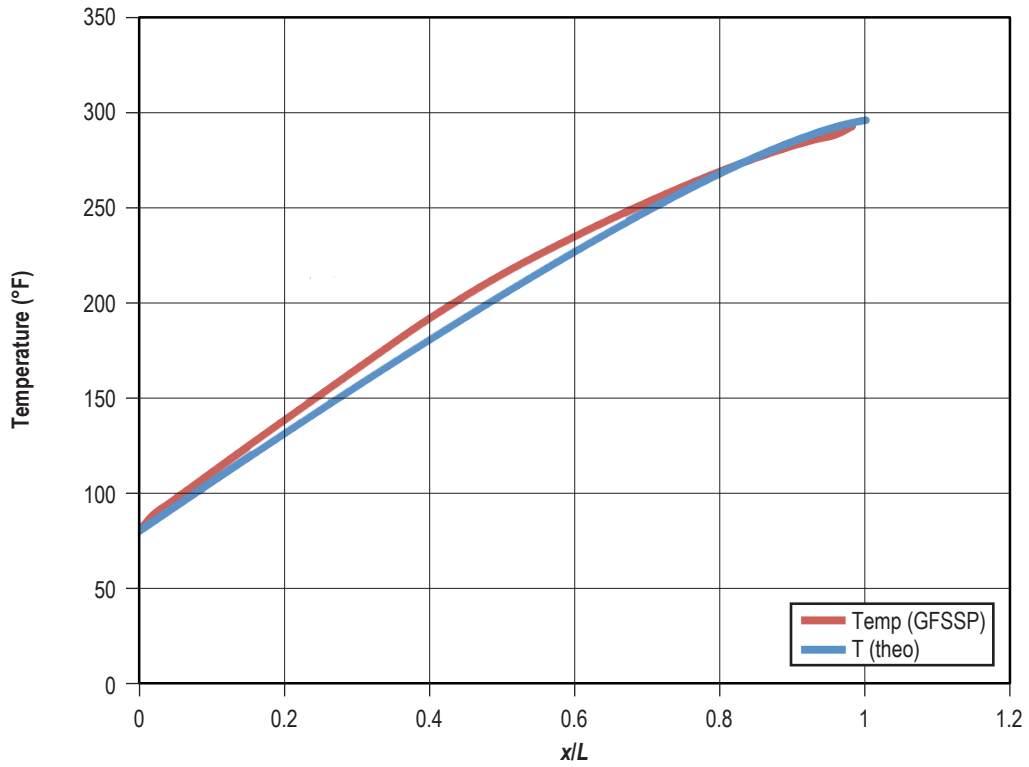
Figure 191. Temperature distribution for Rayleigh flow ($Q = 1,800$ Btu/s).
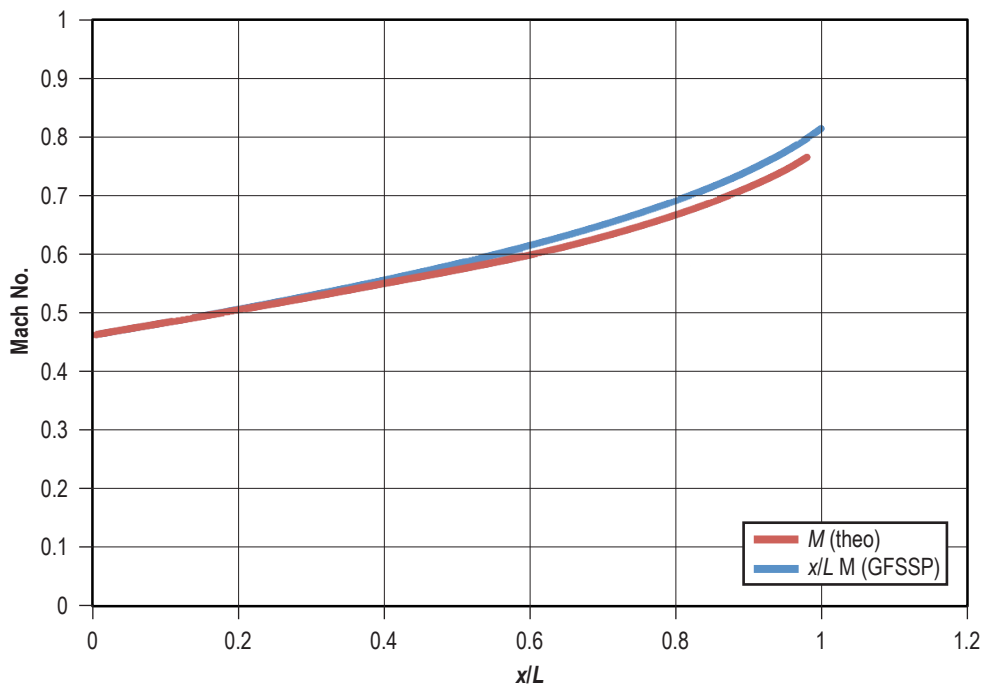
Figure 192. Mach No. distribution for Rayleigh flow ($Q = 1,800$ Btu/s).

6.20 Example 20—Modeling of Closed Cycle Liquid Metal (Lithium) Loop With Heat Exchanger to Heat Helium Gas

6.20.1 Problem Considered

A lithium loop with a counter flow heat exchanger system configuration, as shown in figure 193, was chosen for this example. The specific objective of the lithium circuit analysis using GFSSP was to model a closed loop with heat addition and rejection typical for a thermodynamic cycle. The model includes the pump, reactor core, heat exchanger with liquid lithium and nitrogen gas. This example also employs cyclic boundary to model a closed cycle, and illustrates the use of user-defined fluid by providing property tables because lithium properties are not available in GFSSP's thermodynamic property package. The model calculates the pressure and temperature of the fluid in each component during steady state operation.
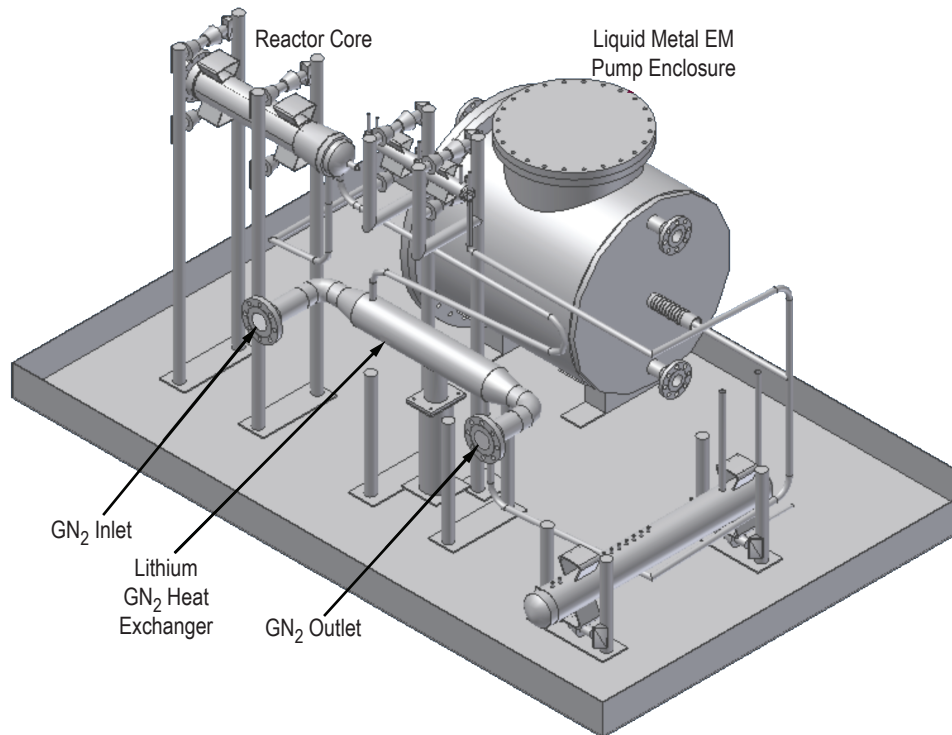
Figure 193. Lithium loop model schematic (example 20).

As shown in figure 194, counter flow heat exchange occurs when the hot branch of the heat exchanger has a flow that is propagating in a direction opposite of the cold branch. This counter flow heat exchanger configuration consists of liquid lithium, at 7 psi and 932 °F, flowing through a 46.56-in concentric annulus duct of 2.8-in inner radius and 5-in outer radius pipe. Also, from the cold side of the heat exchanger, nitrogen is flowing at 200 psi and 477 °F through a 13-in section

of 3.26-in inner diameter pipe, through the heat exchanger, and out through compressible orifice of area of 0.13 in^2 and flow coefficient of 1 at 14.7 psi and 60 °F. All of the initial guess values for this problem are shown in figure 194. The style-IV pump performance is shown in figure 195, which was modeled in a User Subroutine. The lithium properties were introduced through look-up property tables.
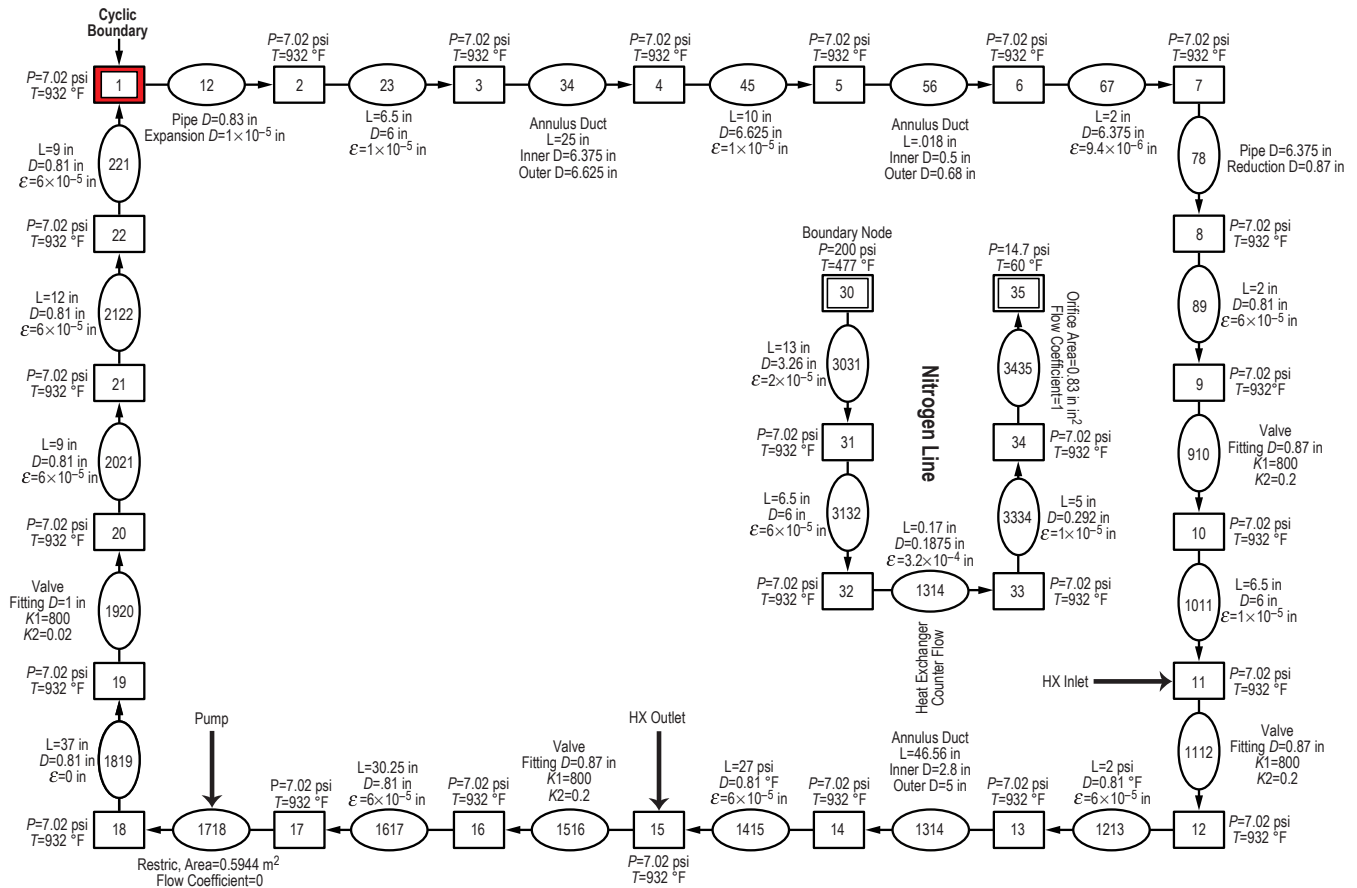
Figure 194. GFSSP model information of example 20.
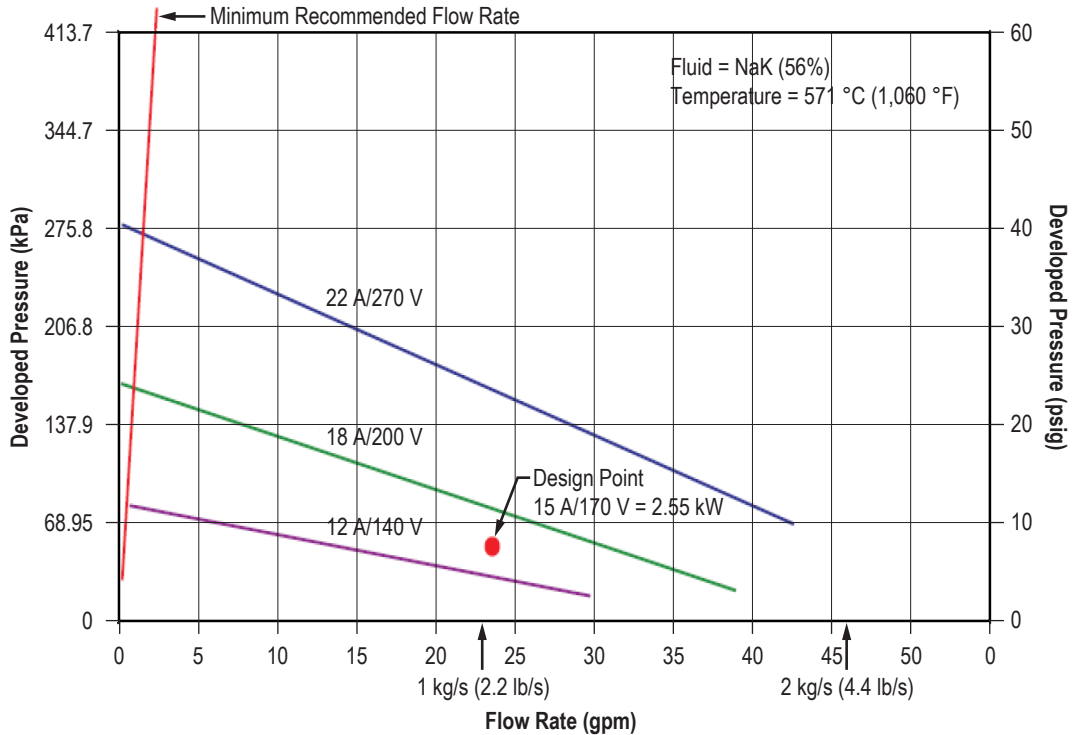
Figure 195. Style-IV ac conduction pump performance curve (reproduced from MSA
Research Corp. dwg. No. C-510356).

The GFSSP analysis provides numerical predictions of pressure and temperature at various locations in the flow circuit.

6.20.2 GFSSP Model

A GFSSP model consisting of 28 nodes and 27 branches is used to represent the NaK test loop and counter flow heat exchanger system shown in figure 194. Nodes 1, 30, and 35 are boundary nodes. Node 1 is a cyclic boundary node at 7.02 psi and 932 °F. This implies that the temperature at node 22 must be equal to the temperature at node 1 and this must be achieved by iteration. Node 30 represents the inlet boundary node at 200 psi and 477 °F for the nitrogen cold flow of Heat Exchanger. Option 2 (Flow Through Restriction) was used for branch 1718 in place of the pump because the flow rate versus developed pressure curve has multiple curves depending on voltage, and therefore must be modeled as a momentum source in a User Subroutine. Branches 1314 and 3233 represent the hot and cold sides, respectively, of the heat exchanger. Figure 196 shows how this model appears in VTASC. GFSSP output results are shown in figure 197.
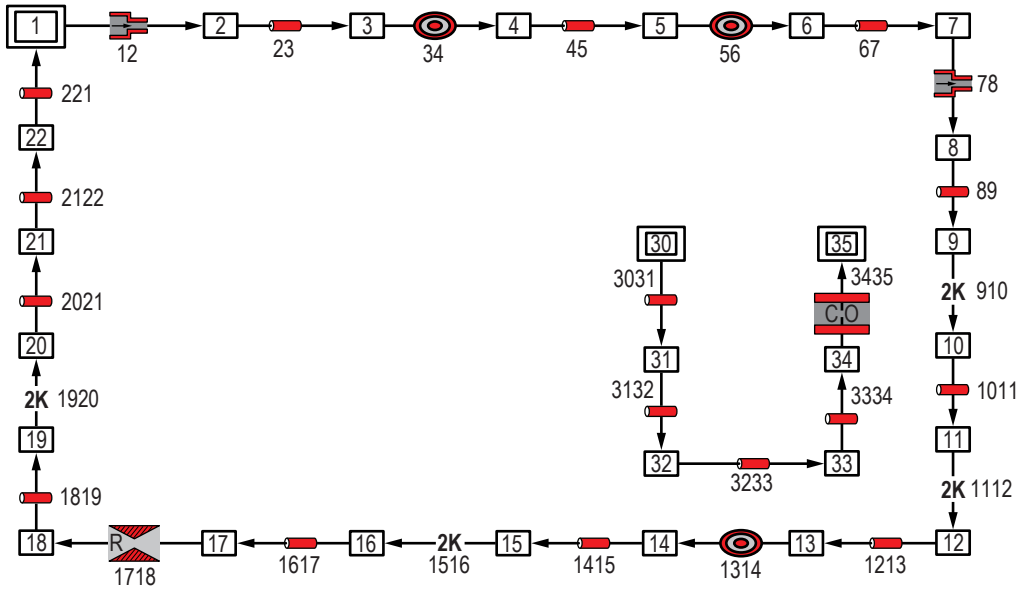
Figure 196. VTASC model of lithium loop with Heat Exchanger (example 20).
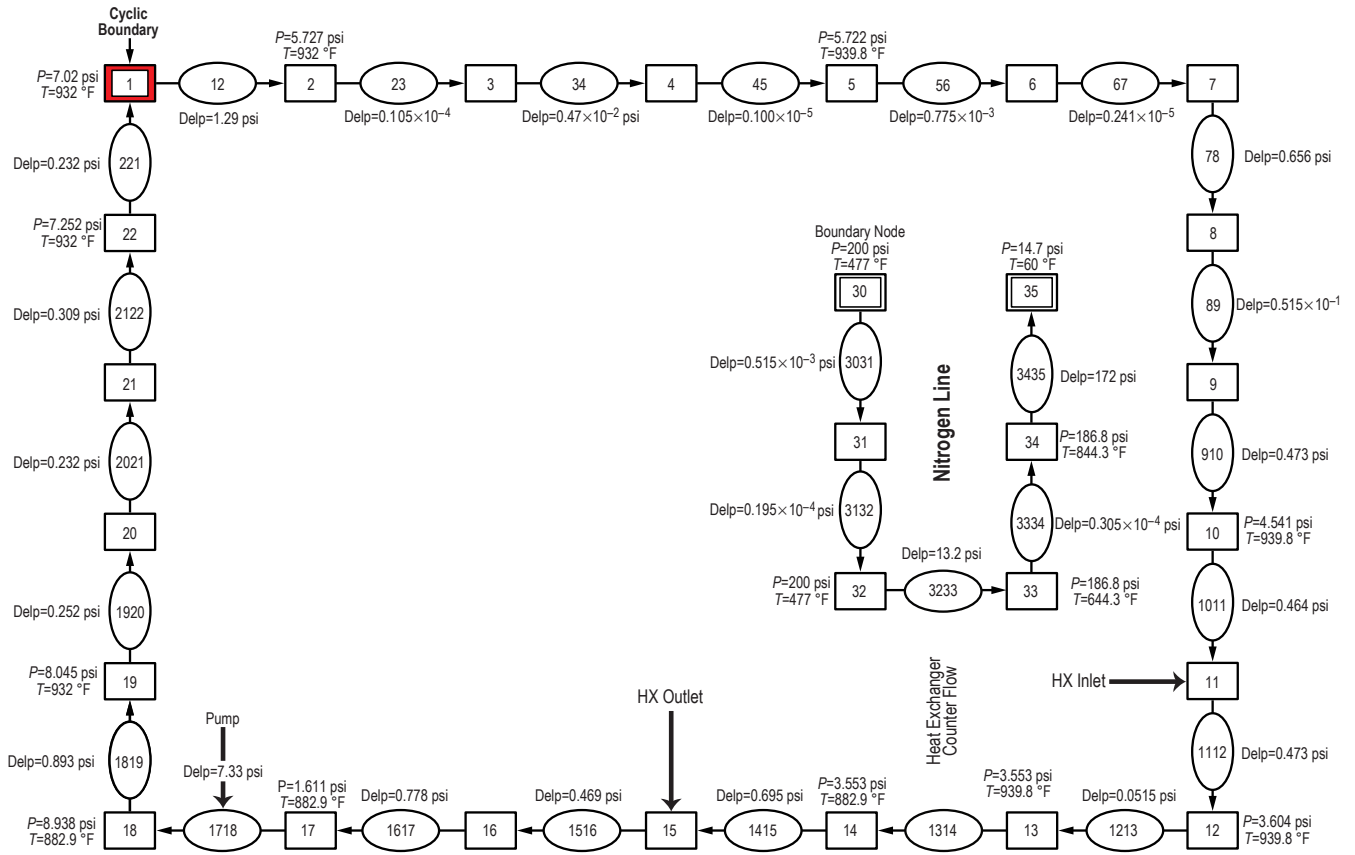
Figure 197. Model results of example 20.

6.20.3 Results

The User Subroutine input and output files of example 20 have been given in appendix Z (on CD inside back cover). The model results are also shown in figures 197 and 198.
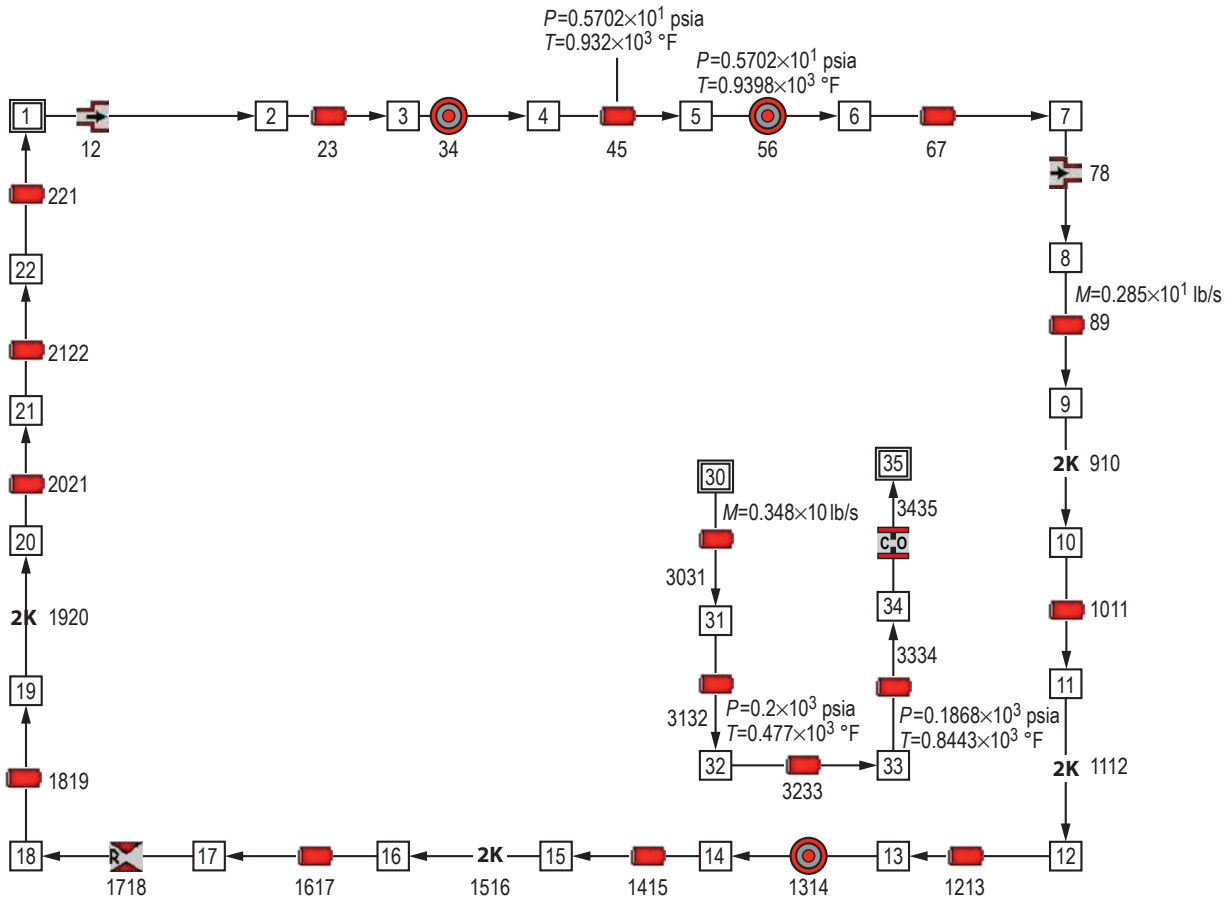
Figure 198. Sample results of example 20 is displayed in the flow circuit.

6.21 Example 21—Internal Flow in a Turbopump

6.21.1 Problem Considered

It is desired to calculate the axial thrust of the SIMPLEX turbopump operating at 25,000 rpm with LOX as the operating fluid.[29] In order to calculate axial thrust, the pressure distribution throughout the secondary flow system of the turbopump must be obtained. Figure 199 shows a schematic of the turbopump. The pressure is known at the inducer inlet, the inducer discharge, on the back face of the impeller (upstream of the Labyrinth seal), on the front face of the impeller shroud (upstream of the Labyrinth seal), at the end of the atmospheric dump lines (two dump lines), and on the front and back face of the turbine. The pressure at the exit of the impeller is not known, due to flow conditions at the pressure tap corresponding to the impeller discharge. (The value of the pressure at the impeller discharge must be estimated.) The values of pressure

and temperature at these positions are listed in table 37, and are boundary conditions for the GFSSP model. Axial thrust and pressures throughout the internal flow circuit are to be calculated using GFSSP.
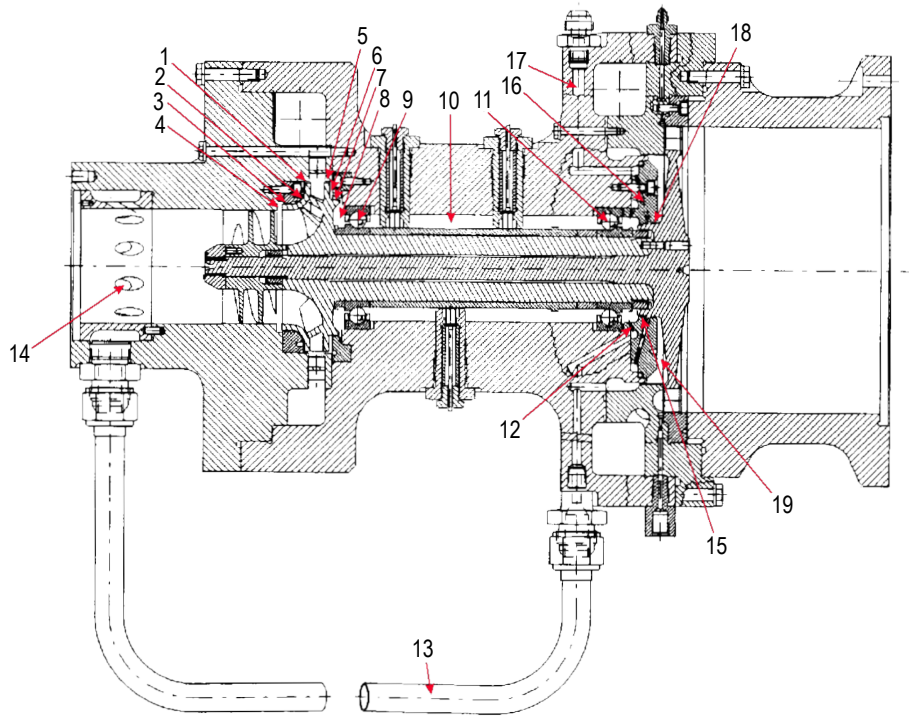
Figure 199. Schematic of the simplex turbopump with secondary flow passages indicated.

Table 37. Boundary conditions.

Location	Node No.	Pressure (psia)	Temperature (°F)
Impeller discharge	100	1,100	–286.6 (assumed)
Impeller shroud	107	1,078	–286.6 (assumed)
Impeller inlet	105	346.2	–286.6
Impeller back face	102	1,025	–286.6
Turbine front face	116	62.6	–265.6
Dump	130	14.7 (assumed)	–286.6 (assumed)
Inducer inlet	140	93.7	–286.6
Turbine back face	180	14.7 (assumed)	–286.6 (assumed)

6.21.2 GFSSP Model

Figure 199 indicates the flow passages that will be modeled using GFSSP. The modeled passages are numbered, with each number corresponding to the passage as follows:

(1) Axial flow between the impeller shroud and the housing flowing from the impeller discharge.
(2) Radially inward flow between the impeller shroud and the housing.
(3) Flow through the labyrinth seal at the end of the impeller shroud.
(4) Radially inward flow between the end of the impeller shroud and the housing flowing into the impeller inlet.
(5) Axial flow between the impeller and the housing flowing from the impeller discharge.
(6) Radially inward flow between the impeller back face and the housing.
(7) Flow through the labyrinth seal at the lip on the back face of the impeller.
(8) Radially inward flow between the impeller back face and the housing flowing into the first bearing.
(9) Flow through the first rolling element bearing.
(10) Axial flow along the impeller shaft between the bearings.
(11) Flow through the second rolling element bearing.
(12) Flow through eight radially outward holes (for return lines).
(13) Flow through two external return lines.
(14) Flow through eight radially inward holes flowing into inducer inlet.
(15) Flow through the first turbine-end labyrinth seal.
(16) Flow through 22 radially outward holes (for dump lines).
(17) Flow through two external dump lines.
(18) Flow through the second turbine-end labyrinth seal,
(19) Radially outward flow between the front face of the turbine and the housing.

Additionally, a dummy branch connects the front face and the back face of the turbine for the calculation of axial thrust.

The detailed model of the secondary flow passages for this turbopump is shown schematically in figure 200, and the VTASC model is shown in figure 201. Heat is added to nodes 110 and 112 to account for the heat transferred from the bearings. Heat is also added to node 109 to match the temperature at that node with the experimental data. As seen in figure 200, branches 2401 through 2408 and 2131 through 2138 were originally modeled as eight separate but identical pipe branches; branches 2801 through 2822 were originally 22 separate branches. In the current VTASC model seen in figure 201, these are now modeled using the Parallel Tube branch option (option 21). Because the Parallel Tube option does not include entrance and exit loss K-factors, branches 2132 and 2802 were added with equivalent lengths to produce an identical K_f.
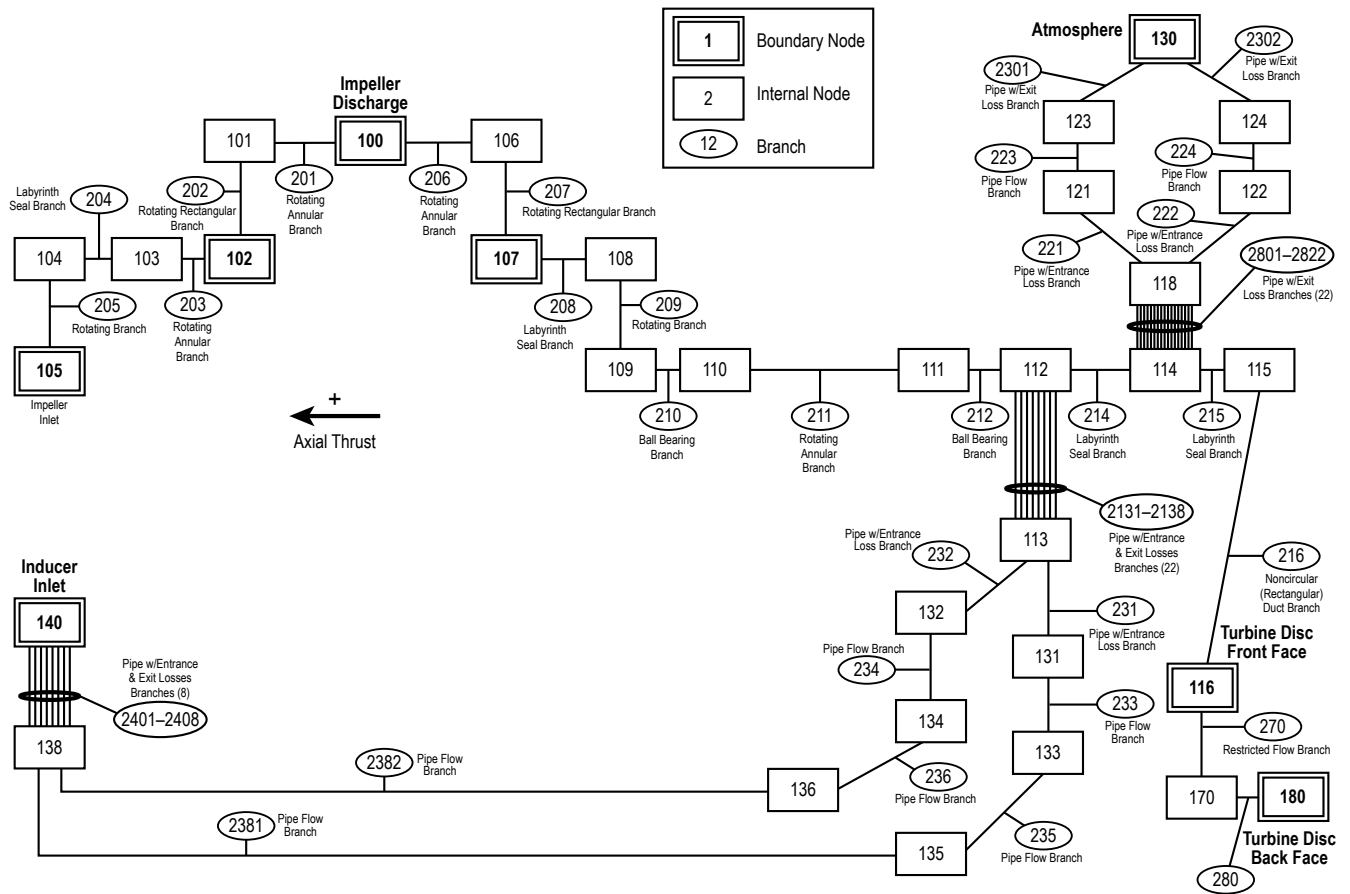
Figure 200. Simplex turbopump detailed model.
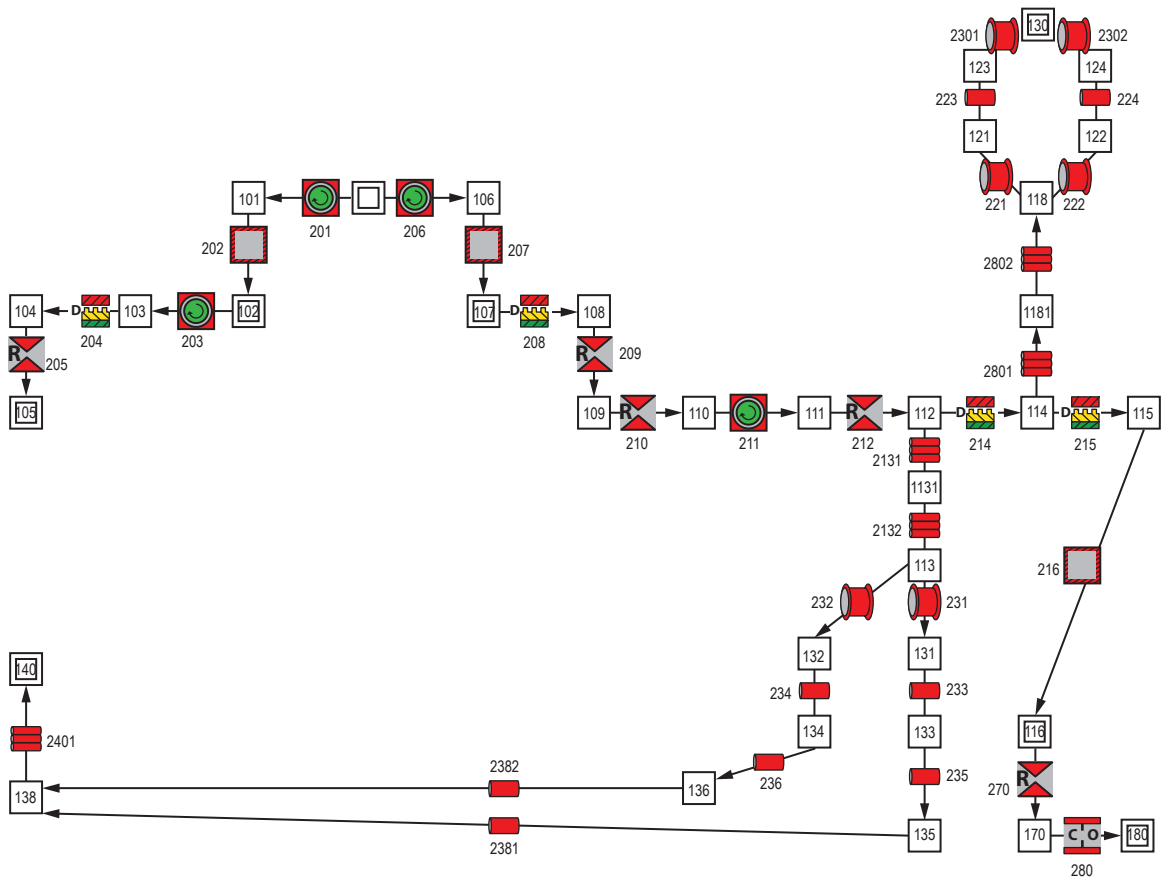
Figure 201. Simplex turbopump VTASC model.

6.21.3 Results

The input and output files of this example are included in appendix AA as ex21.dat and ex21.out (on CD inside back cover). The model predicts that the axial thrust will be 567 lb_f. Figure 202 shows the calculated pressures at nodes 108 through 112, 114, and 115 comparing them to the available experimental data at nodes 109 and 112. Figure 203 shows the calculated and experimental temperatures.
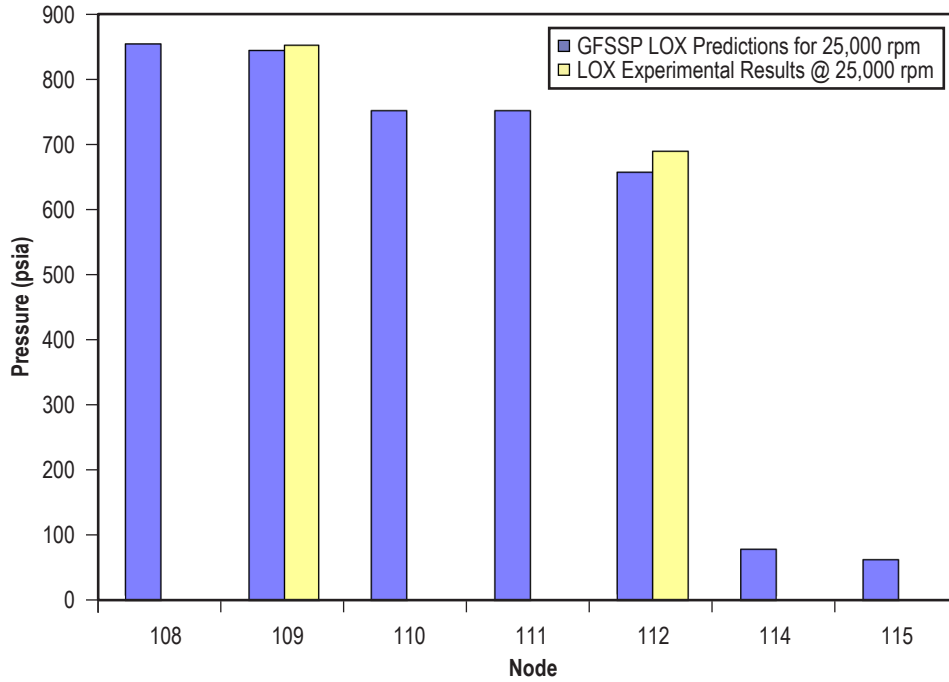
Figure 202. Simplex pressure predictions compared to experimental data.
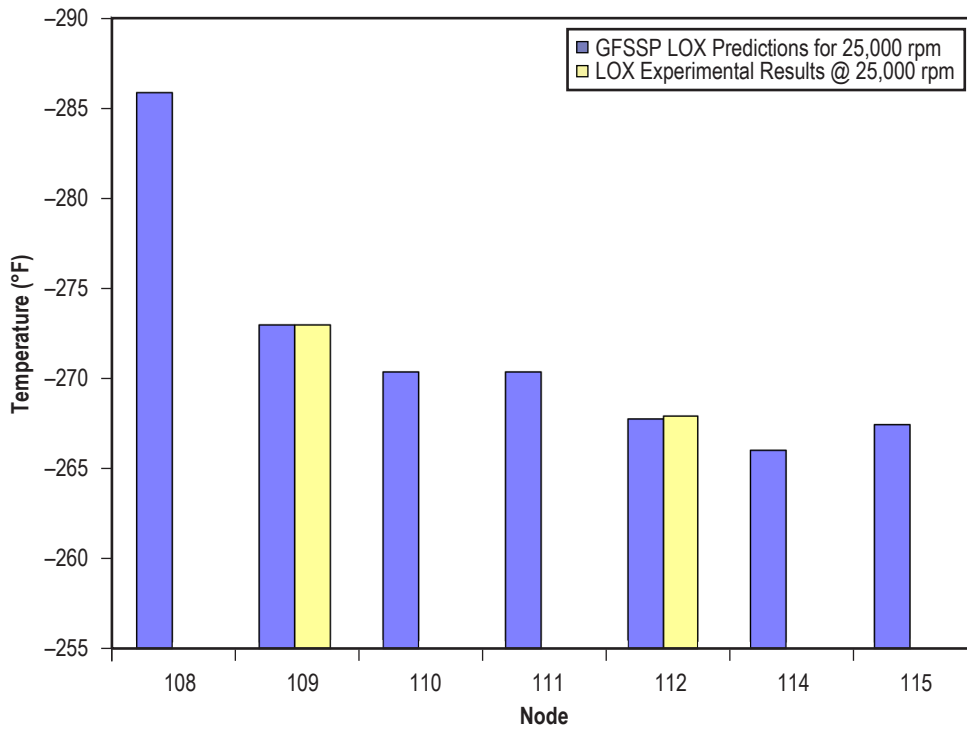
Figure 203. Simplex temperature predictions compared to experimental data.

6.22 Example 22—Simulation of a Fluid Network With Fixed Flow Rate Option

6.22.1 Problem Considered

Sometimes the boundary condition of a fluid network is given as a flow rate instead of a boundary pressure. Besides using an estimated boundary pressure and a flow regulator branch, the user could also attach a Fixed Flow branch (option 24) to a boundary node. GFSSP is always solving the conservation equations for flow rate, so the Fixed Flow branch does not set the desired flow rate, per se. Instead, it uses a nearly vertical pump curve to trick the code into solving for the branch pressure rise necessary to achieve the required flow rate (see sec. 3.1.7.24).

6.22.2 GFSSP Model

Figure 204 shows the VTASC canvas of a model with two Fixed Flow branches. Note that Fixed Flow branches must be attached to a boundary node. Water passes from node 2 through a gate valve and a 1,500-ft-long pipe to the exit boundary (node 4) at 14.7 psia. Pressures in boundary nodes 1 and 5 are arbitrarily set to 14.7 psia, and the boundary temperature is 60 °F.
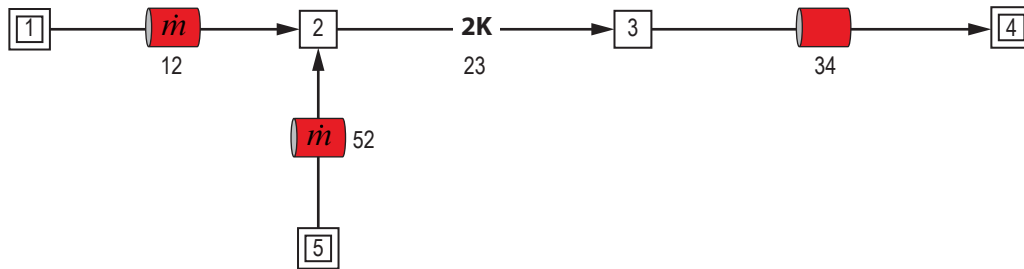
Figure 204. VTASC model of a fluid network with fixed flow rate option.

Both steady state and transient cases are modeled. In the steady state case, the required flow rates in branches 12 and 52 are 100 and –10 lb/s, respectively. Note that a negative flow rate can be handled to use a Fixed Flow branch as an exit branch.

In the transient case, history files are used to set the required flow rates. Table 38 is the history file for branch 12. In the first 10 s the required flow decreases from 100 to 50 lb/s, after which the flow rate remains constant.

Table 38. Example 22 history file for fixed flow branch 12.

Number of Data Points—Three			
τ (s)	f (lb/s)	Area (in^2)	Note: This line does not appear in dat file
2	100	200	
10	50	200	
20	50	200	

Table 39 is the history file for branch 52. For the first 10 s, flow must exit (negative flow rate) at 10 lb/s, after which flow enters the network at 10 lb/s.

Table 39. Example 22 history file for fixed flow branch 52.

Number of Data Points—Four			
τ (s)	f (lb/s)	Area (in^2)	Note: This line does not appear in dat file
0	−10	200	
10	−10	200	
10.1	10	200	
20	10	200	

6.22.3 Results

The steady state input and output files of this example are included in appendix BB as ex22ss.dat and ex22ss.out (on CD inside back cover). In the steady state case, the required 100 lb/s flows into the network via branch 12, while 10 lb/s flows out branch 52. The remaining 90 lb/s exits at branch 34, satisfying conservation of mass. It is recommended that the user always verify that the program has converged on the required flow rates; if it does not succeed, rerunning with a tighter convergence criteria will usually solve the problem.

The transient input and output files of this example are included in appendix CC as ex22tr. dat and ex22tr.out (on CD inside back cover). Note that although the program ran with the default convergence criteria of 1×10^{-4}, a plot of the sum of the inlet and outlet flow rates showed that conservation of mass was not achieved in every time step. This was remedied by tightening the convergence criteria to 1×10^{-5} and increasing the maximum number of iterations to 2,000.

Figure 205 is a plot of the flow rates in Fixed Flow branches 12 and 52, showing that they meet their target flow rates set by the history files. Figure 206 compares the sum of the flow rates in branches 12 and 52 with the flow rate in branch 34, to demonstrate that mass is being conserved in the system.
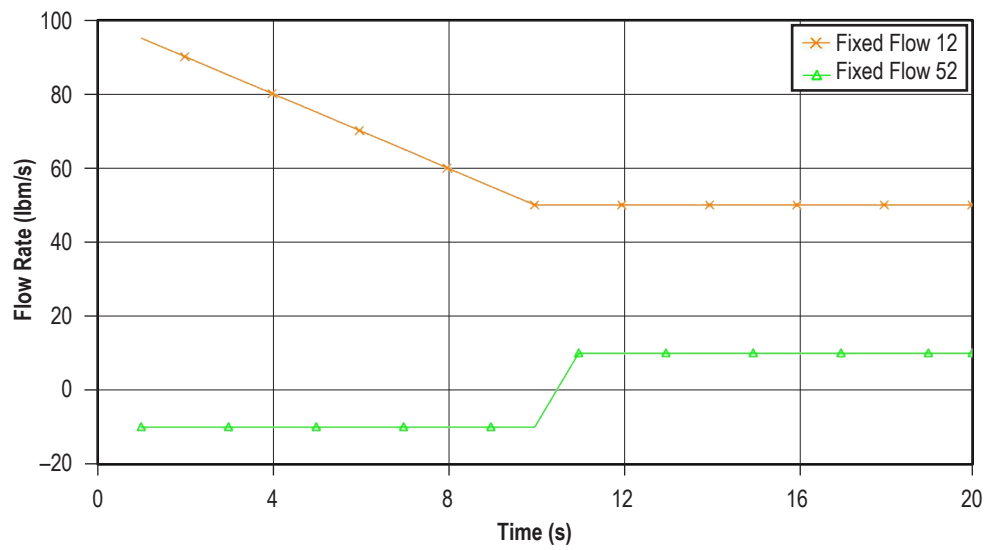
Figure 205. Flow rates in fixed flow branches.
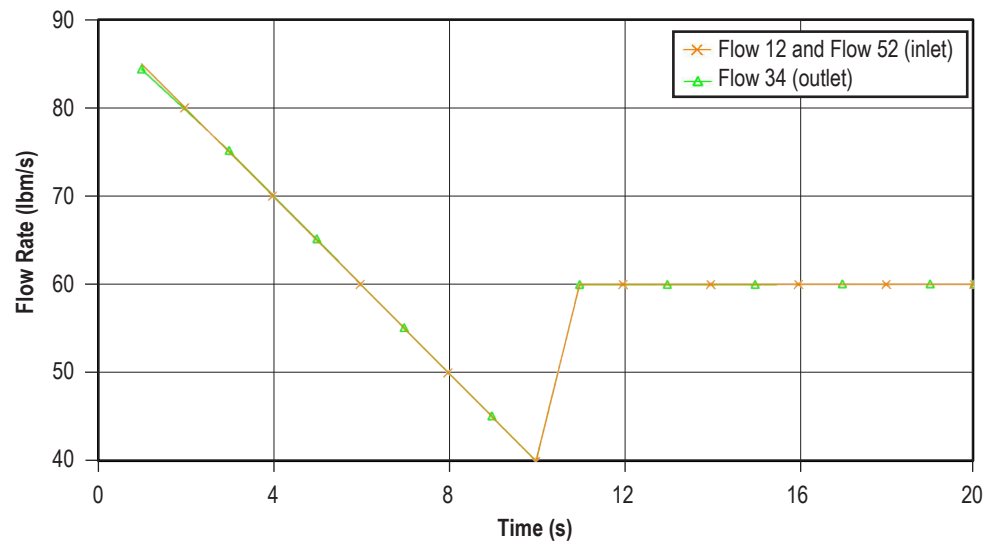
Figure 206. Comparison of inlet and outlet flow rates.

6.23 Example 23—Helium-Assisted, Buoyancy-Driven Flow in a Vertical Pipe Carrying Liquid Oxygen With Ambient Heat Leak

6.23.1 Problem Considered

This example models a vertical pipe carrying LOX, as might be the case with a propellant recirculation line. To drive the flow without the aid of a pump, one can use buoyancy effects from heat transfer and/or helium injection. This example includes three GFSSP models in three VTASC files: a steady state model of a recirculation line, a steady state model of a helium injector, and a transient model of a recirculation line with helium injection. The third model demonstrates the use of the VTASC's file import feature to combine the first two models. It is also an example of one of GFSSP's mixture options: separate energy conservation equations for separate species.

6.23.2 GFSSP Model

Figure 207 shows the VTASC canvas of the Recirculation Line model. LOX at 55.78 psia and –272.5 °F flows from boundary node 1 up to boundary node 8 at 53 psia. The pressure differential of 2.78 psid represents the gravitational head of 6 ft of LOX, so that in the absence of heat transfer, the fluid will remain stagnant. This can be confirmed by running the model with Conjugate Heat Transfer deactivated; GFSSP will predict a flow rate near zero. The recirculation line is 6 ft long with a diameter of 1.87 in. It is discretized into six pipe branches. Five of the 6 ft are assumed to be well insulated, but 1 ft is left uninsulated, to allow heat transfer from the ambient. Solid nodes 9 and 10 represent the mass of the Inconel 718 pipe, separated by a conductor of 0.225-in thickness. There is convection from the outside wall of the pipe to a 70 °F ambient temperature with an assumed convection coefficient of 2 Btu/hr-ft^2-°F. There is also convection between the inside wall and the LOX, with the convection coefficient calculated by GFSSP using the Dittus-Boelter correlation.
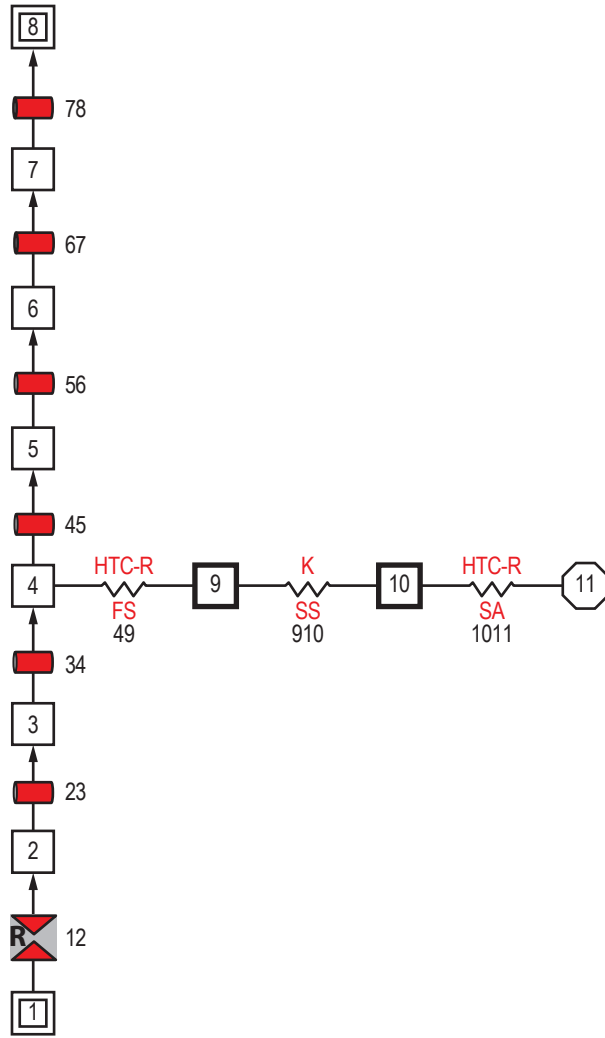
Figure 207. VTASC model of LOX recirculation line with heat transfer.

Figure 208 shows the VTASC canvas of the Helium Injector model. Helium at 425 psia and 100 °F flows from boundary node 1 to boundary node 4 at 14.7 psia. The injector line is broken into two pipes of 0.152 in diameter. The pipes are connected by a small orifice with an area of 0.0012566 in^2, corresponding to a diameter of 0.04 in.

Figure 208. VTASC model of a helium injector.

Figure 209 shows the VTASC canvas of the combined models. The Recirculation Line model was imported into the Helium Injector model using the File Import feature on the VTASC's File menu. An offset of 100 was used to renumber the nodes and branches in the Recirculation Line that conflicted with node numbers in the Helium Injector.
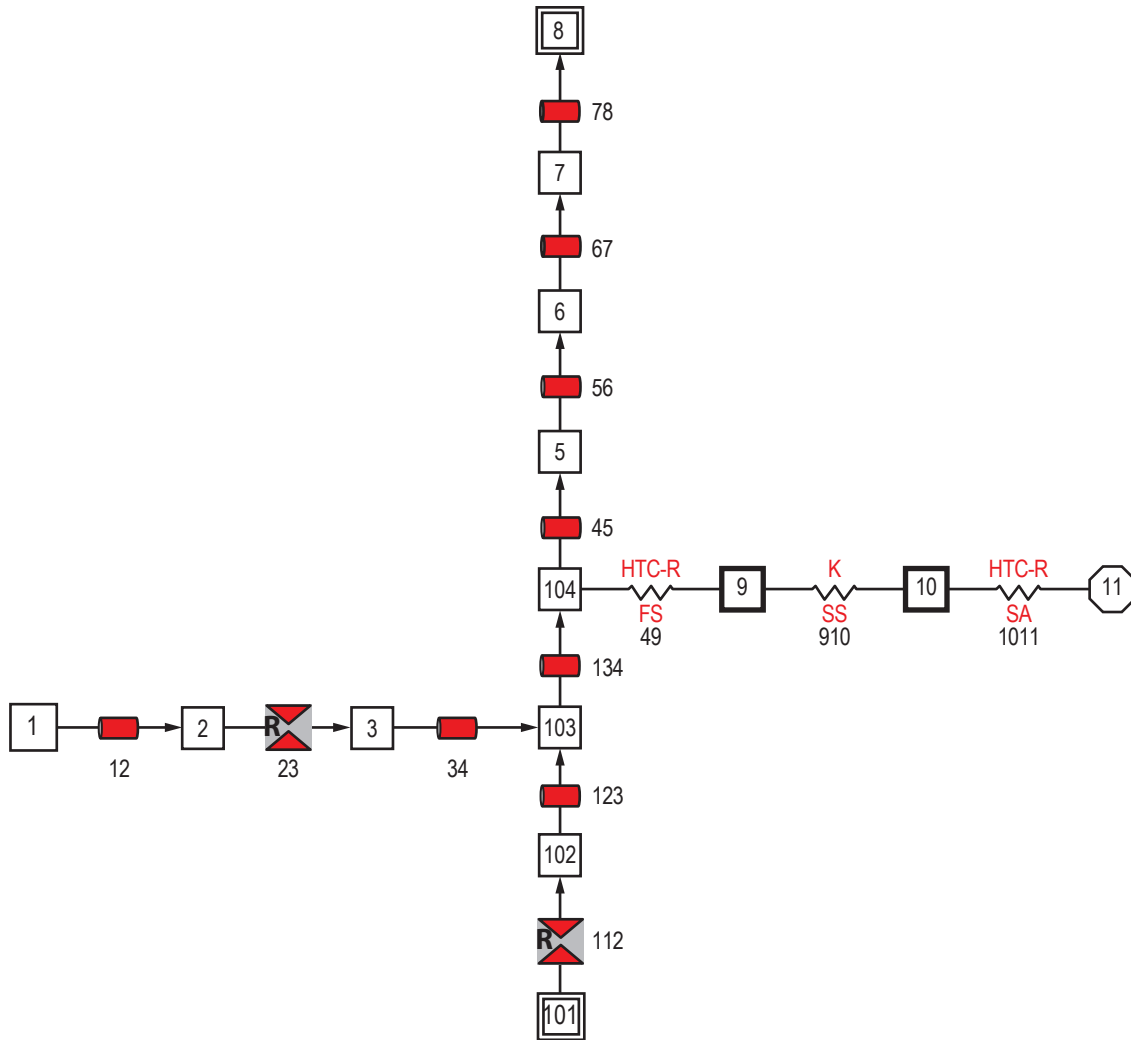
Figure 209. Combined VTASC model of LOX recirculation line model imported into helium injector model.

Helium Injector branch 34 has been connected to LOX Recirculation Line node 103, and boundary node 4 has been deleted. This allows the helium to bubble into the LOX line just below node 104, where heat transfers into the fluid. The problem is changed from steady to unsteady, with a run of 100 s at 0.1 s intervals.

Because there are two fluids, oxygen and helium, the mixture option is enabled on the Circuit Options page. The mixture option is changed from the default temperature solution of the energy equation (see eq. (16)) to option Enthalpy 2: Separate Energy Equations for Separate Species (see sec. 3.1.3.2). This is the only mixture option capable of handling phase change and is required to model the oxygen vapor that forms in the flow.

6.23.3 Results

The LOX recirculation line steady state model files are included in appendix DD as ex23A.dat and ex23A.out (on CD inside back cover). GFSSP predicts that the heat transfer drives a flow of 0.172 lb/s in the vertical pipe. The temperature of the fluid rises from –272.5 to –271.3 °F, while its density decreases from 66.7 to 63.3 lb/ft^3. Part of the density decrease is caused by formation of oxygen vapor, indicated by the exiting fluid having a quality of 0.0014. The conjugate heat transfer calculations indicate a heat rate into the fluid of 0.106 Btu/s. The Dittus-Boelter correlation predicts the internal convection coefficient to be 33.3 Btu/hr-ft^2-°F. The solution of the Solid Energy Conservation equation gives an inner wall temperature of –248 °F.

An interesting exercise is to evaluate how efficient the heat transfer is at driving the fluid flow. The work required to achieve the predicted flow rate is:

$$\dot{W} = \frac{\dot{m}\Delta P}{\rho} = \frac{\left(0.172\ \frac{\text{lb}}{\text{s}}\right)\left(2.78\ \frac{\text{lb}_\text{f}}{\text{in}^2}\right)\left(144\ \frac{\text{in}^2}{\text{ft}^2}\right)}{\left(66.4\ \frac{\text{lb}}{\text{ft}^3}\right)\left(778\ \frac{\text{lb}_\text{f}-\text{ft}}{\text{Btu}}\right)} = 0.001333\ \frac{\text{Btu}}{\text{s}}\ . \tag{113}$$

Since 0.106 Btu/s of heat leak are driving this flow, the efficiency of the process is just 1.26%.

The helium injector steady state model files are included as ex23B.dat and ex23B.out in appendix EE (on CD inside back cover). The predicted flow rate is 0.00163 lb/s.

The files for the unsteady model that combines the recirculation line with the helium injector are included as ex23C.dat and ex23C.out in appendix FF (on CD inside back cover). Figure 210 is a plot of the flow rate of helium in the injector (F12) and of the LOX-helium mixture in the recirculation line (F56). It is seen that a small flow rate of 0.005 lb/s of helium, combined with the heat transfer into the uninsulated portion of the line, increases the flow rate to a steady value of 1.80 lb/s. Initially the flow rate is even larger because the pipe is chilling down from 70 °F, and heat transfer is greater.
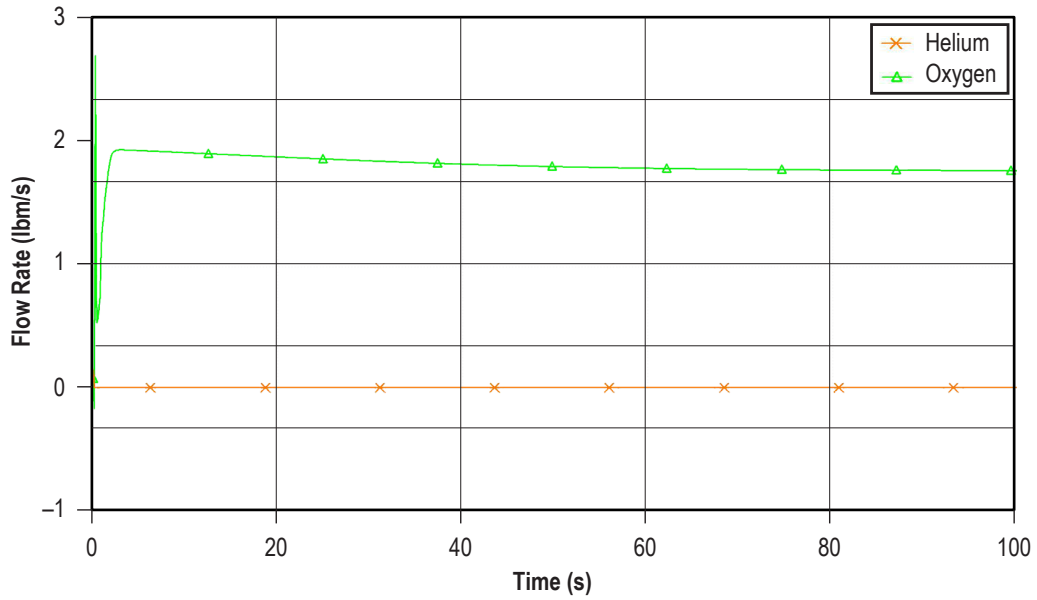
Figure 210. Helium and mixture flow rates.

Figure 211 plots the mass fractions of the two species in node 7. The exiting fluid is predicted to be 99.7% oxygen and 0.3% helium by mass.
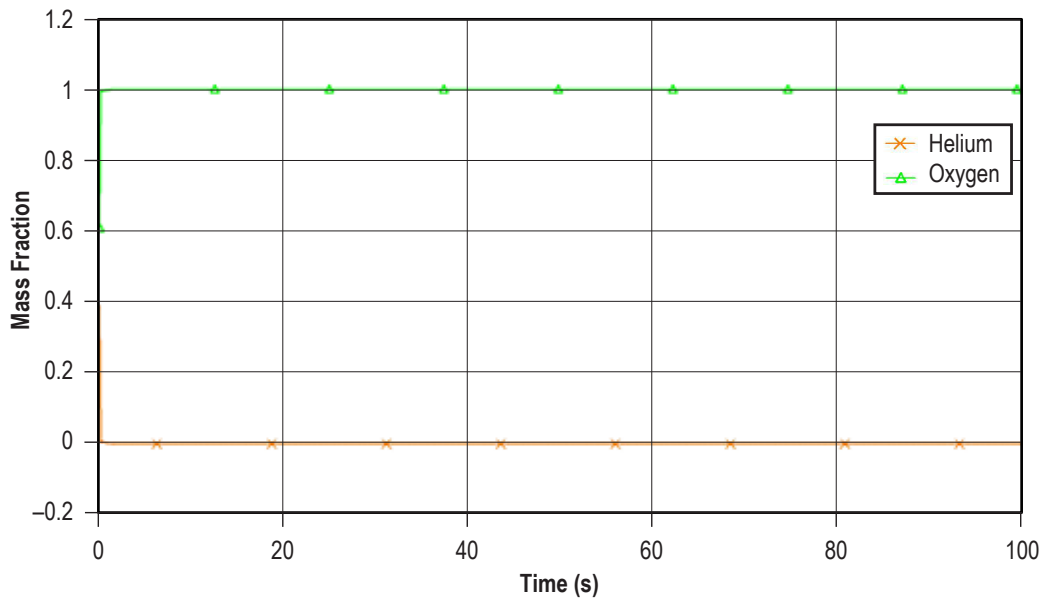
Figure 211. Mass fractions of He and O_2 at node 7.

Figure 212 plots the temperatures of solid nodes 9 and 10, representing the inner and outer uninsulated pipe walls. The temperatures fall to their steady state values in about 80 s. These steady state temperatures are slightly colder than those predicted by the recirculation model without helium injection, because the higher flow rate with helium injection increases the convection coefficient.
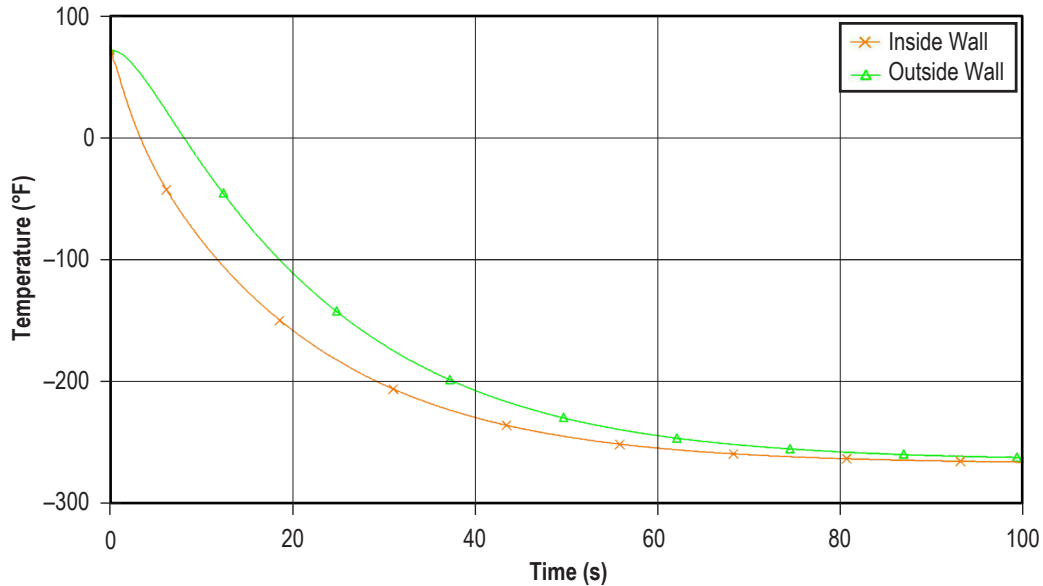
Figure 212. Inside and outside temperature of pipe wall.

6.24 Example 24—Simulation of Relief Valve in a Pressurized Tank

6.24.1 Problem Considered

This example demonstrates the use of a pressure relief valve to allow flow out of a node that has exceeded a defined pressure. This advanced option is different from the Control Valve branch option (option 18). The control valve monitors pressure in a node and opens to allow fluid to enter the node when pressure gets too low. In contrast, the relief valve monitors pressure in a node and opens to allow fluid to exit the node when pressure gets too high.

Figure 213 shows a schematic of the system being modeled. Air at 35 psia and 70 °F flows through a 0.316-in-diameter orifice in and into a 10 ft^3 tank, initially at 14.7 psia. A relief valve on top of the tank has a cracking pressure of 9.5 psid above the ambient pressure of 14.7 psia. As the tank fills with air, the relief valve will open whenever the tank pressure exceeds 24.2 psia.
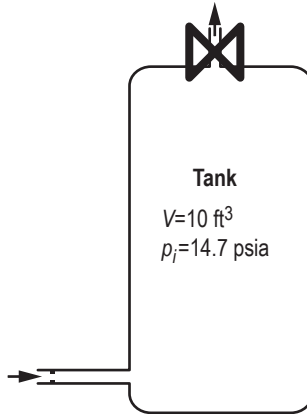
Figure 213. Example 24 relief valve schematic.

6.24.2 GFSSP Model

Figure 214 shows the VTASC canvas of the model. Boundary node 1 represents the source of high pressure air. Interior node 2 represents the 10 ft^3 tank. Boundary node 3 is the ambient environment. Compressible orifice 23 represents the pressure relief valve.
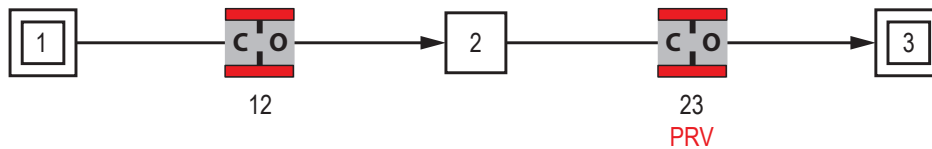
Figure 214. Example 24 relief valve VTASC model.

Table 40 shows the file that controls how far the relief valve opens as a function of the pressure differential across the valve. If a Restriction or Compressible Orifice option is used as the relief valve, then area in square inches is given as a function of pressure in psid. If a Valve with Cv Branch option is used, then area is replaced with Cv values. The first point should be the reseating pressure, which uses a very small area or Cv to represent a closed valve. The last point should be the pressure at which the valve is opened to the maximum area.

Table 40. Example 24 pressure relief valve control file.

4		(No. of ΔP versus area points in interpolation table (max=20)
7	1×10^{-16}	ΔP (psi), A (in^2) (reseat pressure with small area))
8	0.24	
9	0.48	
10	0.72	(Max possible area for fully open valve)

6.24.3 Results

The pressure relief valve model files are included in appendix GG as ex24.dat and ex24.out (on CD inside back cover). Figure 215 is a plot of the pressure in the tank over time. Whenever the pressure exceeds the cracking pressure of 9.5 psid (24.2 psia), the relief valve opens, allowing air to escape. When the pressure falls below the reseat pressure of 7 psid (21.7 psia), the relief valve closes and the tank begins to pressurize again.
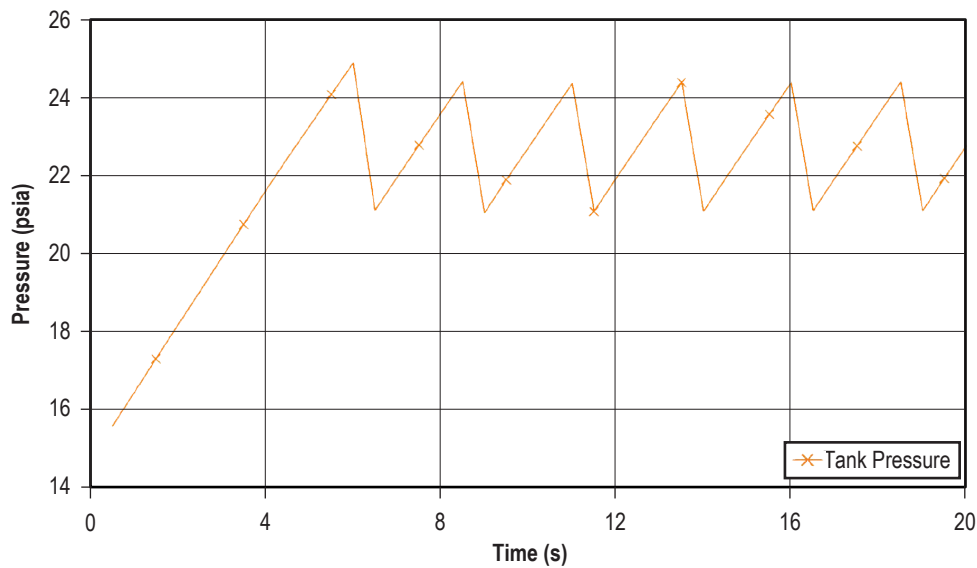
Figure 215. Pressure inside tank.

Figure 216 plots the flow rate into and out of the tank. The orange line shows that the flow into the tank is relatively steady around 0.06 lb/s. The green line plots flow out of the tank. This is normally zero, but jumps to approximately 0.3 lb/s whenever the relief valve opens.
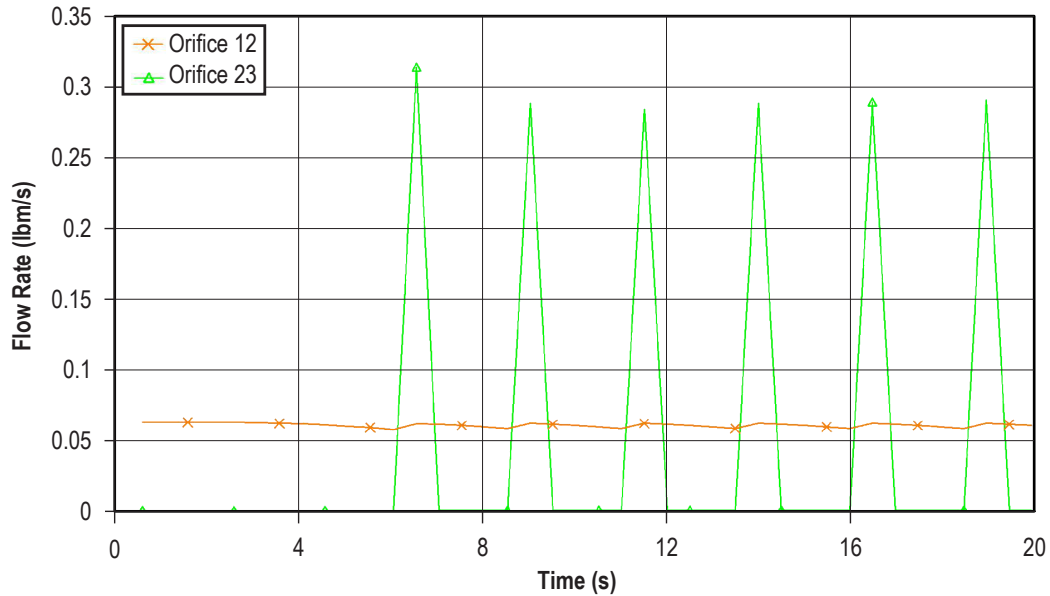
Figure 216. Flow in tank and out of tank.

6.25 Example 25—Two-Dimensional Recirculating Flow in a Driven Cavity

6.25.1 Problem Considered

In this example, two-dimensional recirculating flow in a square cavity[49] has been modeled using GFSSP's multidimensional flow calculation capability. In a square cavity the flow is induced by shear interaction at the top wall as shown in figure 217. The length of each wall is 12 in. The density of the fluid is assumed constant at 1 lb_m/ft^3, and the viscosity of the fluid is assumed to be 1 $lb_m/(ft \cdot s)$. The bottom and side walls are fixed. The top wall is moving to the right at a constant speed of 100 ft/s. The corresponding Reynolds number for this situation is Re = 100.
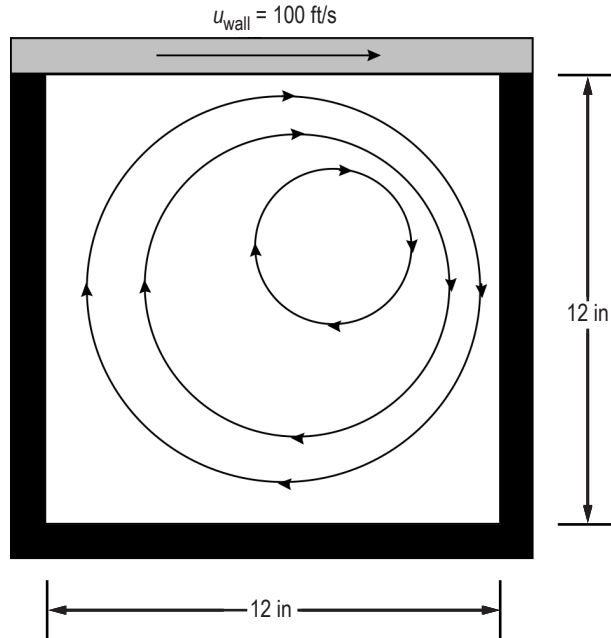
Figure 217. Flow in a shear driven square cavity.

6.25.2 GFSSP Model

The GFSSP model of the driven cavity consists of 50 nodes (49 of which are internal) and 84 branches. For numerical stability, one boundary node with an arbitrary pressure of 14.7 psi was introduced. A unit depth (1 in) was assumed for the required areas. It is assumed that the shear area for each branch is 2 in^2. The shear distance between adjacent branches is 1.71429 in and the shear distance between walls and their adjacent branches is 0.85714 in. For transverse momentum, only adjacent parallel branches were considered, and only connecting branches not associated with the boundary node were used. The constant density option was used for the fluid, and the corresponding density and viscosity used were 1 lb$_m$/ft^3 and 1 lb$_m$/(ft·s), respectively. The bottom and side walls are fixed. The top wall moves to the right at 100 ft/s. All parallel angles are 0°, and all transverse angles are 90°.

The grid generation is performed in two steps: (1) Grid Generation is activated through Advanced options and the system network is built with multidimensional elements as shown in figure 218 (a) and (b), and (2) the grid is then generated (fig. 218(b)) through a dialog box (sec. 5.4.10).
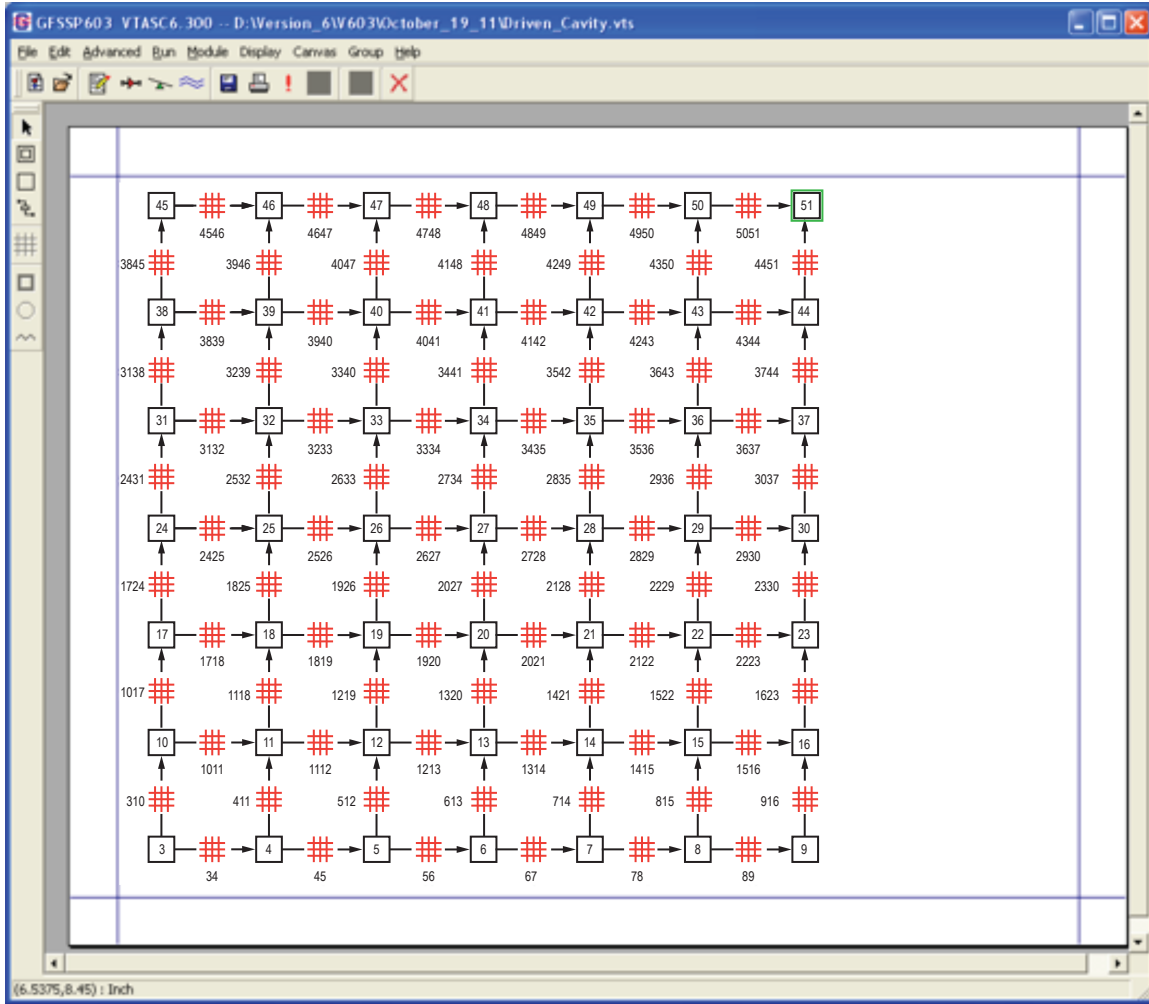
Figure 218. Two-dimensional cartesian grid generation in VTASC—element 1: (a) System network with expandable grid and (b) expandable two-dimensional cartesian grid.

6.25.3 Results:

Figure 219 shows a comparison between the benchmark numerical solution and the GFSSP 7×7 node model velocity profiles along a vertical plane at the horizontal midpoint. As can be seen in figure 219, the results of this crude GFSSP model compare very favorably with the benchmark numerical solution of Burggraf.[50]
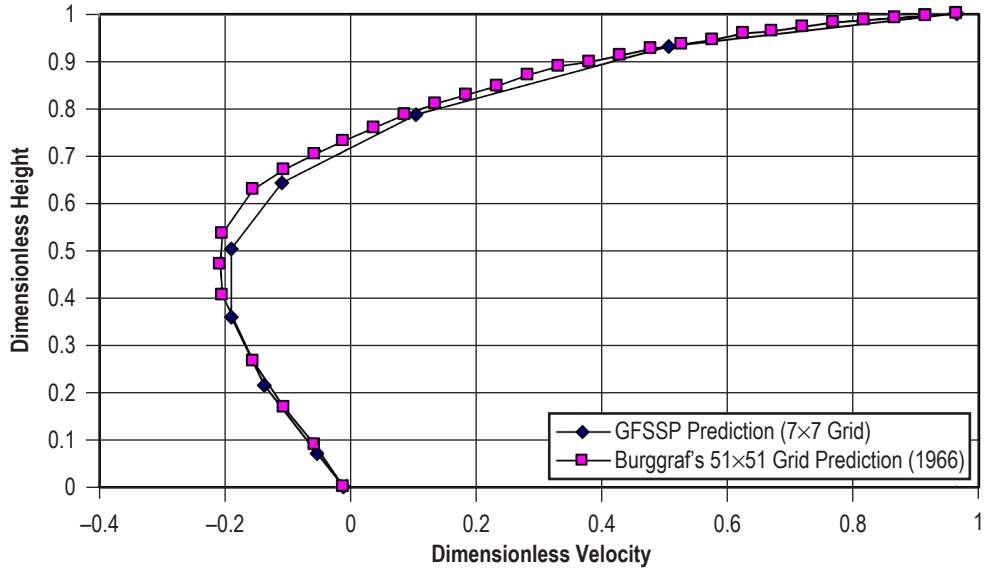
Figure 219. Shear driven square cavity centerline velocity distribution.

The predicted velocity field and pressure contours are shown in figure 220. The recirculating flow pattern and stagnation of flow near the top right corner are clearly shown in the figure. The predicted stream traces from calculated velocity field is shown in figure 221.
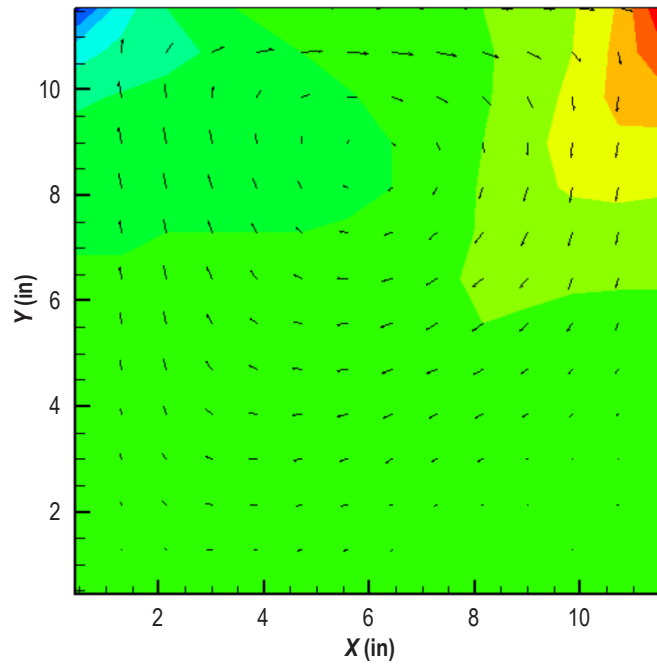
Figure 220. Predicted velocity field and pressure contours.
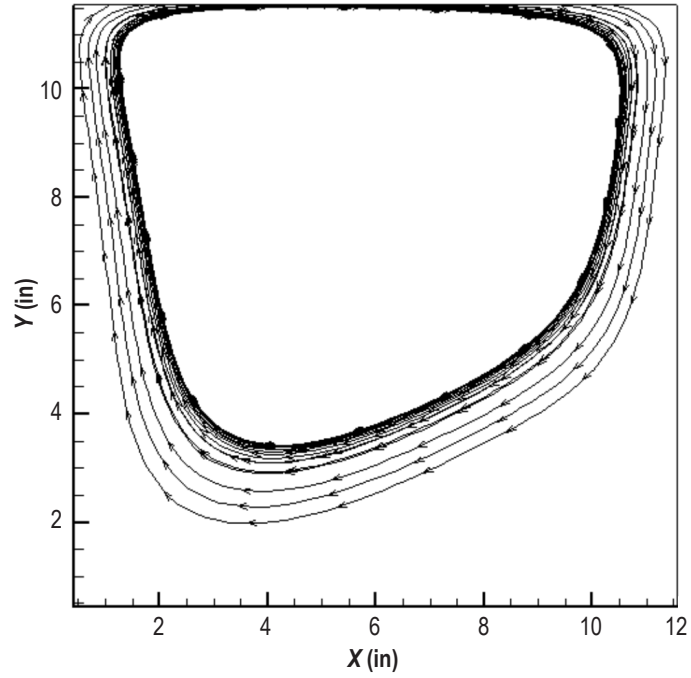
Figure 221. Predicted stream traces in the driven cavity.

Appendix HH has the example 25 data as ex25.dat and ex25.out (on CD inside back cover).

APPENDIX A—DERIVATION OF K_f FOR PIPE FLOW

It is assumed that there is a dynamic equilibrium that exists between the friction and the pressure forces. Therefore, the momentum conservation equation can be expressed as:

$$P_u - P_d = K_f \dot{m}^2 \; , \tag{114}$$

where K_f is a function of f, L, D, and ρ.

For a fully developed pipe flow, the momentum conservation equation can be written as:

$$\tau \, \pi \, DL = \left(P_u - P_d \right) \frac{\pi D^2}{4} \; . \tag{115}$$

The Darcy friction factor (f) can be expressed as:

$$f = \frac{8 \, \tau \, g_c}{\rho u^2} \; . \tag{116}$$

From the continuity equation,

$$u = \frac{4\dot{m}}{\rho \, \pi \, D^2} \; . \tag{117}$$

Substituting equations (116) and (117) into equation (115) gives:

$$P_u - P_d = \frac{8 f L \dot{m}^2}{g_c \, \rho \pi^2 D^5} \; . \tag{118}$$

Therefore,

$$K_f = \frac{8 f L}{g_c \rho \pi^2 D^5} \; . \tag{119}$$

APPENDIX B—SUCCESSIVE SUBSTITUTION METHOD OF SOLVING COUPLED NONLINEAR SYSTEMS OF ALGEBRAIC EQUATIONS

The application of the successive substitution method involves the following six steps:

(1) Develop the governing equations:

$$x_1 = f_1(x_1, x_2, x_3, ..., x_n)$$

$$x_2 = f_2(x_1, x_2, x_3, ..., x_n)$$

$$...$$

$$x_n = f_n(x_1, x_2, x_3, ..., x_n) \; . \tag{120}$$

If there are n number of unknown variables, there are n number of equations.

(2) Guess a solution for the equations. Guess $x_1^*, x_2^*, x_3^*, ..., x_n^*$ as an initial solution for the governing equations.

(3) Compute new values of $x_1^*, x_2^*, x_3^*, ..., x_n^*$ by substituting $x_1^*, x_2^*, x_3^*, ..., x_n^*$ in the right-hand side of equation (124).

(4) Under-relax the computed new value, $x = (1-\alpha)x^* + \alpha x$, where α is the under-relaxation parameter.

(5) Replace $x_1^*, x_2^*, x_3^*, ..., x_n^*$ with the computed value of $x_1^*, x_2^*, x_3^*, ..., x_n^*$ from step (4).

(6) Repeat steps (3)–(5) until convergence.

APPENDIX C—NEWTON-RAPHSON METHOD OF SOLVING COUPLED NONLINEAR SYSTEMS OF ALGEBRAIC EQUATIONS

The application of the Newton-Raphson Method involves the following seven steps:

(1) Develop the governing equations. The equations are expressed in the following form:

$$f_1(x_1, x_2, x_3, ..., x_n) = 0$$

$$f_2(x_1, x_2, x_3, ..., x_n) = 0$$

$$...$$

$$f_n(x_1, x_2, x_3, ..., x_n) = 0 \ .$$

(121)

If there are n number of unknown variables, there are n number of equations.

(2) Guess a solution for the equations. Guess $x_1^*, x_2^*, x_3^*, ..., x_n^*$ as an initial solution for the governing equations.

(3) Calculate the residuals of each equation. When the guessed solutions are substituted into equation (120), the right-hand side of the equation is not zero. The nonzero value is the residual:

$$f_1(x_1^*, x_2^*, x_3^*, ..., x_n^*) = R_1$$

$$f_2(x_1^*, x_2^*, x_3^*, ..., x_n^*) = R_2$$

$$...$$

$$f_n(x_1^*, x_2^*, x_3^*, ..., x_n^*) = R_n \ .$$

(122)

The intent of the solution scheme is to correct $x_1^*, x_2^*, x_3^*, ..., x_n^*$ with a set of corrections $x_1', x_2', x_3', ..., x_n'$ such that $R_1, R_2, R_3, ..., R_n$ are zero.

(4) Develop a set of correction equations for all variables. First construct the matrix of influence coefficients:

$$
\begin{matrix}
\dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \dfrac{\partial f_1}{\partial x_3} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\[2ex]
\dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \dfrac{\partial f_2}{\partial x_3} & \cdots & \dfrac{\partial f_2}{\partial x_n} \\[2ex]
\cdots & \cdots & \cdots & \cdots & \cdots \\[1ex]
\dfrac{\partial f_n}{\partial x_1} & \dfrac{\partial f_n}{\partial x_2} & \dfrac{\partial f_n}{\partial x_3} & \cdots & \dfrac{\partial f_n}{\partial x_n}
\end{matrix} \quad .
\tag{123}
$$

Then construct the set of simultaneous equations for corrections:

$$
\begin{bmatrix}
\dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \dfrac{\partial f_1}{\partial x_3} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\[2ex]
\dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \dfrac{\partial f_2}{\partial x_3} & \cdots & \dfrac{\partial f_2}{\partial x_n} \\[2ex]
\cdots & \cdots & \cdots & \cdots & \cdots \\[1ex]
\dfrac{\partial f_n}{\partial x_1} & \dfrac{\partial f_n}{\partial x_2} & \dfrac{\partial f_n}{\partial x_3} & \cdots & \dfrac{\partial f_n}{\partial x_n}
\end{bmatrix}
\begin{bmatrix}
x_1' \\[1ex] x_2' \\[1ex] \\ x_n'
\end{bmatrix}
=
\begin{bmatrix}
R_1 \\[1ex] R_2 \\[1ex] \\ R_n
\end{bmatrix} \quad .
\tag{124}
$$

(5) Solve for $x_1', x_2', x_3', \ldots, x_n'$ by solving the simultaneous equations.

(6) Apply correction to each variable.

(7) Iterate until the corrections become very small.

APPENDIX D—GLOSSARY OF FORTRAN VARIABLES IN THE COMMON BLOCK

GFSSP contains 16 common blocks of variables. The following is a listing of the common blocks and the variables contained within them along with a description of each including units where applicable. Note that the units of several variables described in the common block are different in the input and output files. GFSSP converts the units of the following variables prior to solving the conservation equations:

Variable	Input/Output Unit	GFSSP Unit
Length	inches	feet
Area	inches^2	feet^2
Volume	inches^3	feet^3
Angle	degree	radian
Pressure	lb$_f$/in^2	lb$_f$/ft^2
Temperature	°F	°R

Variable	Description
1. COMMON/CFILNUM/	**Common block for file numbers**
NWRTE	File number for the command line preprocessor output file
NPRNT	File number for the main output file
NREAD	File number for the input deck
NGSPK	File number for the GASPAK swap file
NFNOD	File number for the node restart file
NGFSOUT	File number for the GFSSP.OUT output file
NFBR	File number for the branch restart file
NGASP	File number for the GASP.OUT output file
NHSTN	File number for the HISTN.XLS output file
NHSTB	File number for the HISTBR.XLS output file
NHSTF	File number for the input history files (except rotational and control valve history files)
NCVHST	File number for maximum and minimum pressure history of control valves
NCVCHR1	File number for time schedule history of open valve characteristics of control valve sub-options
NCVCHR2	File number for time schedule history of closed valve characteristics of control valve sub-options
NHSTROT	File number for the input rotational history file

Variable	Description
NERROR	File number for the ERROR.XLS output file
NRP1DAT	File number for the RP-1 property files
NUSR1	User defined file number/integer
NUSR2	User defined file number/integer
NUSR3	User defined file number/integer
NUSR4	User defined file number/integer
NUSR5	User defined file number/integer
NUSR6	User defined file number/integer
NUSR7	User defined file number/integer
NUSR8	User defined file number/integer
NUSR9	User defined file number/integer
NUSR10	User defined file number/integer
NBRPLT	File number for branch results for VTASC post-processing
NBRWINP	File number for branch results for Winplot post-processing
NCOND	File number for thermal conductivity property data
NCP	File number for specific heat property data
NCVHST	File number for control valve history file
NDPLT	File number for node results for VTASC post-processing
NDWINP	File number for node results for Winplot post-processing
NVOPFILE	File number for history file of valve opening/closing
NSLDPLT	File number for solid node results for VTASC post-processing
NSSCPLT	File number for solid to solid conductor results for VTASC post-processing
NSFCPLT	File number for solid to fluid conductor results for VTASC post-processing
NSACPLT	File number for solid to ambient conductor results for VTASC post-processing
NSSRCPLT	File number for solid to solid radiation conductor results for VTASC post-processing
NSLDWIN	File number for solid node results for Winplot post-processing
NSSCWIN	File number for solid to solid conductor results for Winplot post-processing
NSFCWIN	File number for solid to fluid conductor results for Winplot post-processing
NSACWIN	File number for solid to ambient conductor results for Winplot post-processing
NSSCRWIN	File number for solid to solid radiation conductor results for Winplot post-processing

Variable	Description
NBWPLOT	File number for storing data in binary for Winplot post-processing
NPRGFILE	File number for pressure regulator history data
NFRGFILE	File number for flow regulator history data
NAHSTF	File number for ambient node history data
NSTDERR	File number for writing error information to VTASC
NUVWINP	File number for writing user variables for plotting
NFIXFHST	File number for fixed flow information
NTECPLT	File number for storing 2-D data for Tecplot
2. COMMON/CNODEI/	**Common block for node index information**
NNODES	Total number of nodes
NINT	Total number of internal nodes
NBND	Total number of boundary nodes
NODREF	Reference node number
NODE( )	User assigned node number
INDEX( )	Index number to distinguish between internal and boundary node (1-internal node, 2-boundary node)
INODE( )	Internal node number
IBNODE( )	Boundary node number
NUMBR( )	Number of branches connected to an internal node
NAMEBR(,)	Name of branches connected to an internal node
NMNODE	Number of nodes with moving boundary
NRNODE	Number of nodes with reaction
IVERS	Version number of the code
IBU( )	Not used
IBD( )	Not used
NUMUB( )	Number of upstream branches of a given branch
NUMDB( )	Number of downstream branches of a given branch
INDEXUB(,)	Index number of upstream branches in a given branch
INDEXDB(,)	Index number of downstream branches in a given branch
3. COMMON/CNODEF/	**Common block for node variable informatio**
P( )	Pressure (lb_f/ft^2)
PM( )	Pressure (lb_f/ft^2) at previous time step
RHO( )	Density (lb_m/ft^3)
RHOM( )	Density (lb_m/ft^3) at previous time step
ZL( )	Not used
H( )	Specific enthalpy (Btu/lb_m)
HM( )	Specific enthalpy (Btu/lb_m) at previous time step
TF( )	Temperature (°R)

Variable	Description
TM( )	Temperature (°R) at previous time step
CX(,)	Mass specie concentration
CXM(,)	Mass specie concentration at previous time step
CONDF( )	Thermal conductivity (Btu/ft-s-°R)
EMU( )	Absolute viscosity (lb_m/ft-s)
GAMA( )	Ratio of specific heats
CM(,)	Molar specie concentration
RNODE( )	Gas constant (lb_f-ft /lb_m-°R)
RNODEM( )	Gas constant (lb_f-ft /lb_m-°R) at previous time step
XV( )	Mass fraction of vapor in mixture
Z( )	Compressibility factor
ZM( )	Compressibility factor at the previous time step
AREAN( )	Surface area of node (ft^2) for thrust calculation
VOLUME( )	Node volume (ft^3)
VOLUMEM( )	Node volume (ft^3) at previous time step
CVNODE( )	Specific heat at constant volume (Btu/lbm-°R)
EM( )	Mass (lb_m)
EMM( )	Mass (lb_m) at previous time step
U( )	Specific internal energy (Btu/lbm-°R)
UM( )	Specific internal energy (Btu/lbm-°R) at previous time step
EMS( )	Mass source (lb_m/s) at node
HSORCE( )	Heat source (Btu/s or Btu/lb_m) at node
NAMEND(,)	Neighboring nodes of an internal node for conduction calculation
DISTC(,)	Distance between internal and neighboring node (ft)
AREAC(,)	Surface area of heat conduction between internal and neighboring node (ft^2)
NABOR( )	Same as NUMBR(I)
VBOUND( )	Velocity of moving boundary (ft/s), for moving boundary option
HSORCR( )	Heat source (Btu/s) due to chemical reaction
H2( )	Hydrogen concentration in the reaction product
O2( )	Oxygen concentration in the reaction product
H2O( )	Water concentration in the reaction product
ENTROPY( )	Specific entropy (Btu/lb_m-°R)
ENTRPYM( )	Specific entropy (Btu/lb_m-°R) at previous time step
CPNODE( )	Specific heat at constant pressure (Btu/lb_m-°R)
GIBBS( )	Not used.
EXERGY( )	Not used
PR( )	Prandtl Number
SGEN( )	Entropy generation rate (Btu/s-°R)

Variable	Description
TSORCE( )	Not used
CVNODEM( )	Specific heat at constant volume in previous time step (Btu/lb$_m$-°R)
HSDENOM( )	Not used
VELIN( )	Average velocity at node inlet (ft/s)
AREAIN( )	Average area at node inlet (ft^2)
TFMAX( )	Not used
HMAX( )	Not used
QDOTSU( )	Not used
QDOTSP( )	Not used
RHOL( )	Liquid density (lb$_m$/ft^3)
RHOV( )	Vapor density (lb$_m$/ft^3)
EMUL( )	Liquid viscosity (lb$_m$/ft-s)
EMUV( )	Vapor viscosity (lb$_m$/ft-s)
CPNODEL( )	Liquid specific heat (Btu/lb$_m$-°R)
CPNODEV( )	Vapor specific heat (Btu/lb$_m$-°R)
CONDFL( )	Liquid thermal conductivity (Btu/ft-s-°R)
CONDFV( )	Vapor thermal conductivity (Btu/ft-s-°R)
RE( )	Node Reynolds number
PMM( )	Pressure at two time steps prior to current period (lb$_f$/ft^2)
RHOMM( )	Density at two time steps prior to current period (lb$_m$/ft^3)
HMM( )	Sp. Enthalpy at two time steps prior to current period (Btu/lb$_m$)
EMMM( )	Mass at two time steps prior to current period (lb$_m$)
VELINMM( )	Node Velocity at two time steps prior to current period (ft/s)
VELMM( )	Branch Velocity at two time steps prior to current period (ft/s)
TMM( )	Temperature at two time steps prior to current period (°R)
CVNODEMM( )	Sp. heat at two time steps prior to current period (Btu/lb$_m$-° R)
CXMM( )	Mass specie concentration two time steps prior to current period
HLIQ( )	Specific enthalpy of liquid in two-phase mixture (Btu/lb$_m$)
HVAP( )	Specific enthalpy of vapor in two-phase mixture (Btu/lb$_m$)
GAMAL( )	Not used
GAMAV( )	Not used
ELEV( )	Not used
4. COMMON/CBRANCH/	**Common block for branch information**
NBR	Total number of branches in a model
IBRANCH( )	User defined branch number
IBRUN( )	Upstream node number for a given branch
IBRDN( )	Downstream node number for a given branch
FLOWR( )	Mass flow rate (lb$_m$/s)
FLOWRM( )	Mass flow rate (lb$_m$/s) at the previous time step
AK( )	Flow resistance coefficient, K_f, (lb$_f$-s^2/(ft-lb$_m$)^2) for the branch
AKM( )	Flow resistance coefficient, K_f, (lb$_f$-s^2/(ft-lb$_m$)^2) for the branch at the previous time step

Variable	Description
AREA( )	Branch cross-stional area (ft^2)
AREAM( )	Branch cross-stional area (ft^2) at the previous time step
CLF( )	Not used
EL( )	Not used
D( )	Not used
SR( )	Not used
IOPT( )	Branch resistance option number (1-pipe flow, etc.)
DELP( )	Pressure differential across the branch
AREAUP( )	Not used
AREADN( )	Not used
ANGLE( )	Angle (radians) between branch and the gravity vector. Used when gravity is activated
NONBR( )	Not used
NOUBR( )	Number of upstream branches for a given branch
NODBR( )	Number of downstream branches for a given branch
NMUBR(,)	Name of each upstream branch for a given branch
NMDBR(,)	Name of each downstream branch for a given branch
ANGUBR(,)	Angle (radians) between current branch and each upstream branch. Used in the longitudinal inertia option
ANGDBR(,)	Angle (radians) between current branch and each downstream branch. Used in the longitudinal inertia option
VEL( )	Velocity (ft/s) of the fluid in a given branch
VELM( )	Velocity (ft/s) of the fluid in a given branch at the previous time step
RADU( )	Upstream node radius from axis of rotation (for rotation option)
RADD( )	Downstream node radius from axis of rotation (for rotation option)
RPM( )	Rotational speed (in RPM) for rotation option
AKROT( )	Fluid slip factor, K_{rotation}, for rotation (ratio of fluid rotational speed to solid rotational speed)
NRBR	Number of rotating branches
NMBR	Number of branch with momentum source
NIBR	Number of branches with longitudinal inertia
AREAS(,)	Not used
DISTS(,)	Not used
NMNBR(,)	Not used
BRPR1( )	Branch resistance input variable 1 (used in all branch resistance options)
BRPR1M( )	Branch resistance input variable 1 at the previous time step
BRPR2( )	Branch resistance input variable 2
BRPR2M( )	Branch resistance input variable 2
BRPR3( )	Branch resistance input variable 3
BRPR3M( )	Branch resistance input variable 3 at the previous time step

Variable	Description
BRPR4( )	Branch resistance input variable 4
BRPR4M( )	Branch resistance input variable 4 at the previous time step
BRPR5( )	Branch resistance input variable 5
BRPR5M( )	Branch resistance input variable 5 at the previous time step
BRPR6( )	Branch resistance input variable 6
BRPR6M( )	Branch resistance input variable 6 at the previous time step
SORCE( )	Momentum source (lb$_f$)
VOLBRN( )	Branch volume (ft^3)
VOLBRNM( )	Branch volume (ft^3) at previous time step
PIPET( )	Not used
EMOD( )	Not used
EMACH( )	Branch Mach number
SOLID( )	Not used
AREASB( )	Not used
DISTSB( )	Not used
VELSB( )	Not used
REYN( )	Reynolds number in a branch
TWOPHM( )	Not used
VIODF( )	Not used
SRATIO( )	Not used
IBRPR1( )	Index to activate X-momentum source in multi-D option
IBRPR2( )	Index to activate Y-momentum source in multi-D option
IBRPR3( )	Not used
MDGEM( )	Not used
5. COMMON/CPROP/	**Common block for property information**
NF	Total number of fluids in a given model
NF1	NF – 1
NFL	Number of fluids available in the library
WM( )	Molecular weights of fluids in the library
RGAS( )	Gas constants (lb$_f$-ft/ lb$_m$-°R) of fluids in the library
DELH( )	Reference enthalpy (Btu/ lb$_m$) of fluids in the library
DELH1( )	Reference enthalpy (Btu/ lb$_m$) of fluids with respect to 1st reference point
DELH2( )	Reference enthalpy (Btu/ lb$_m$) of fluids with respect to 2nd reference point
NAMEF( )	Character identifier for fluids in the library
NHREF	Index for specifying reference point for enthalpy calculation for mixture
IFLUID( )	Character identifier for fluid index in a model
NFLUID( )	Character identifier for fluid index in the fluid library
NDATA( )	Not used

Variable	Description
PREF	Reference pressure (lb_f/ft^2)
TREF	Reference temperature (°R)
RHOREF	Reference density (lb_m/ft^3)
EMUREF	Reference absolute viscosity (lb_m/ft-s)
G	Gravitational acceleration (32.174 ft/s^2)
GC	Force conversion factor (32.174 ft-lb_m/lb_f-s^2)
PI	π (3.1415926)
HEQ	Energy conversion factor (778.16 ft-lb_f/Btu)
GAMREF	Reference ratio of specific heats
RREF	Reference gas constant
CPREF	Reference specific heat at constant pressure (Btu/lb_m-°R)
AKREF	Reference conductivity (Btu/ft-s-°R)
DELHGP( )	Reference enthalpy (Btu/ lb_m) of fluids in the GASPAK library
GSPMIN( )	Minimum allowable pressure for fluids in GASP library
GSPMAX( )	Maximum allowable pressure for fluids in GASP library
GSTMIN( )	Minimum allowable temperature for fluids in GASP library
GSTMAX( )	Maximum allowable temperature for fluids in GASP library
NCVFL( )	Identifier of fluids in GASPAK library that do not have thermophysical properties
GPPMIN( )	Minimum allowable pressure for fluids in GASPAK library
GPPMAX( )	Maximum allowable pressure for fluids in GASPAK library
GPTMIN( )	Minimum allowable temperature for fluids in GASPAK library
GPTMAX( )	Maximum allowable temperature for fluids in GASPAK library
AKNBP	Not used
DNBP	Not used
INDFLCP	Not used
INDFLV	Not used
SIGMAR	Stephan-Boltzman constant = 4.7611E-13 Btu/ft^2-R^4-s
TCRIT	Not used
TNBP	Not used
HREF	Reference enthalpy (Btu/lb_m)
SREF	Reference entropy (Btu/lb_m-°R)
RCONST	Universal Gas Constant = 1545 lb_f-ft/lb_{mol}-°R)
DCRIT	Critical Density (lb_m/ft^3)
EMUNBP	Viscosity at Normal Boiling Point ((lb_m/ft-s)
FRACM	Fraction of Hydrogen Peroxide in aqua mixture
INDFLR	Fluid index in GASPak
PCRIT	Critical Pressure (lb_f/ft^2)
6. COMMON/CNUM/	**Common block for control variables**
NVAR	Total number of variables to be solved in a model

Variable	Description
TLRNCE	Convergence criteria of Newton-Raphson scheme
ITMAX	Maximum allowable number of Newton-Raphson iterations
ITER	Number of outer loop iterations (reset every time step)
ITERNR	Number of Newton-Raphson iterations (reset every time the Newton-Raphson loop is called)
ITERT	Total number of iterations (reset every time step)
ISTEP	Number of time steps
DIFK	Fractional change in flow resistance coefficient between successive iterations
TAU	Time (s)
DTAU	Time step (s)
TIMEF	Start time (s)
TIMEL	End time (s)
AFACT	Not used
GFACT	Not used
V( )	Variable array for Newton-Raphson scheme
NAME( )	Name of variable array for Newton-Raphson scheme
NPSTEP	Interval of printout for an unsteady calculation.
DIFD	Fractional change in density between successive iterations
RELAXK	Under-relaxation parameter for the K-factor, K_f. NOTE: User must specify that 0<RELAXK<1
RELAXD	Under-relaxation parameter for the density. NOTE: User must specify that 0<RELAXD<1
RELAXH	Under-relaxation parameter for the enthalpy/entropy. NOTE: User must specify that 0<RELAXH<1
NITER	Maximum number of outer iterations if SIMULA is false (set in the input deck)
ITERMIN	Minimum number of iterations if SIMULA is false (currently set at 5)
CC	Convergence criteria
DIFMAX	Maximum normalized correction in Newton-Raphson iteration
GREAT	An arbitrary large number (= 10^{25})
TINY	An arbitrary small number (= 10^{-25})
RELAXTS	Under relaxation parameter in solid temperature equation
NPWSTEP	Interval for Winplot data dump for an unsteady calculation

Variable	Description
GREAT	Arbitrary large number (=1.E25)
SMALL	Arbitrary small number (=1.E-25)
ALPHA1	1st coefficient for sond-order differencing scheme
ALPHA2	2nd coefficient for sond-order differencing scheme
ALPHA3	3rd coefficient for sond-order differencing scheme
IBDF	Index to select 1st or 2nd order differencing scheme
RELAXNR	Under relaxation parameter for Newton-Raphson scheme
RELAXHC	Under relaxation parameter for heat transfer coefficient
ITMAXSLD	Maximum no. of iteration of solid temperature equation
WPLSTEP	Timestep interval of updating Winplot Bianary File
BWFMARGIN	Additional file space allocation for Winplot Binary File
IFRMIX	Index for selecting mixture option
LPP	Not used
AKMIN	Minimum value of K_f (= 1.E-10)
ISOLVE	Index to choose Newton or Broyden Solver
CCMULT	Multiplier for Convergence Criterion (CC)
7. COMMON/CHEX/	**Common block for Heat Exchanger related variables**
NHEX	Number of heat exchangers in a model
NODHIN( )	Upstream node of branch carrying hot fluid
NODHEX( )	Downstream node of branch carrying hot fluid
NODCIN( )	Upstream node of branch carrying cold fluid
NODCEX( )	Downstream node of branch carrying cold fluid
IBRHOT( )	Branch carrying hot fluid
IBRCLD( )	Branch carrying cold fluid
HEXEFF( )	Heat exchanger effectiveness
ITYPHX( )	Index number to describe the type of heat exchanger: 1- Counter flow, 2-Parallel flow
ARHOT( )	Heat transfer area in hot side (ft^2)
ARCOLD( )	Heat transfer area in cold side (ft^2)
UA( )	Product of overall heat transfer coefficient and area (Btu/s-°R)
8. COMMON/CTABLE	
FLAK( )	Filename for thermal conductivity data table
FLRHO( )	Filename for density data table
FLEMU( )	Filename for viscosity data table
FLGAM( )	Filename for specific heat ratio data table
FLH( )	Filename for enthalpy data table
FLS( )	Filename for entropy data table
FLCP( )	Filename for specific heat data table
9. COMMON/CTPA/	**Common block for Turbopump related variables**
NTPA	Number of turbopumps in a model

Variable	Description
IBRPMP( )	Pump branch number
IBRTRB( )	Turbine branch number
NODPMP( )	Node number upstream of pump
NODTRB( )	Node number upstream of turbine
SPEED( )	Operating speed (RPM) of the turbopump
EFFTRB( )	Turbine efficiency
TORQUE( )	Calculated required torque
HPOWER( )	Horsepower of the pump
DIATRB( )	Turbine diameter (ft). Input file uses in
PSITRD( )	Flow coefficient of the turbine at the design point
ETATRB( )	Efficiency of turbine at design point
PSITR( )	Flow coefficient of the turbine at the operating point
10. COMMON/CTVM/	**Common block for Transverse Momentum variables**
NTM	Number of branches for which transverse momentum is calculated
IBRANCHT( )	Name of branch for which transverse momentum will be calculated
NUMBERL( )	Number of parallel branches used to calculate transverse momentum for a given branch
NAMEL(,)	Name of each parallel branch for a given branch
ANGLEL(,)	Angle (radians) between each parallel branch and the current branch. NOTE: If the branches are perfectly parallel and in opposite directions, this angle is π)
NUMBERT(,)	Number of branches connecting each parallel branch and the current branch
NAMELT(, ,)	Name of each connecting branch, corresponding to each parallel branch for the current branch
ANGLELT(, ,)	Angle (radians) between each connecting branch and the current branch
11. COMMON/CSHR/	**Common block for Shear variables**
NSHR	Number of branches for which shear will be calculated
IBRNCHSH( )	Name of branch for which shear will be calculated
NUMBRSH( )	Number of parallel branches, which will contribute to the shear of the current branch
NAMESH(,)	Names of the parallel branches, which will contribute to the shear of the current branch
ANGLESH(,)	Angle (radians) between each parallel branch and the current branch. NOTE: If the branches are perfectly parallel and in opposite directions, this angle is π)
AREASH(,)	Shear area (ft^2) between each parallel branch and the current branch

Variable	Description
DISTSH(,)	Distance (ft) between the each parallel branch and the current branch
NSOLID( )	Number of solid wall adjacent to the current branch
VSOLID(,)	Velocity (ft/s) of each solid corresponding to the current branch
ANGSOLID(,)	Angle (radians) between each solid walls corresponding to the current branch
AREASOL(,)	Shear area (ft^2) between each solid wall corresponding to the current branch
DISTSOL(,)	Distance (ft) between each solid wall corresponding to the current branch
ENTSHR( )	Entropy generated due to shear for the current branch
NAMELU( )	Name of upstream branch of a branch in multi-D option
NAMELD( )	Name of downstream branch of a branch in multi-D option
NBRMOM	Number of branches with momentum source
IBRMOMS( )	Index of branches with momentum source
IDIR( )	Index to define branch direction
NMUBRMS( )	Upstream branch number that supplies inlet momentum
12. COMMON/CTRANS/	**Common block for Transient variables**
VOLN(,)	Node volume (in^3) - time (s) array used in the input history file for the variable geometry option
VOLB(,)	Branch volume (in^3) - time (s) array used in the input history file for the variable geometry option
AREAB(,)	Branch area (in^2) - time (s) array used in the input history file for the variable geometry option
HEIGHT(,)	Branch height (in) - time (s) array used in the input history file for the variable geometry option. This array is only used for branch resistance option 3 (noncircular duct)
WIDTH(,)	Branch width (in) - time (s) array used in the input history file for the variable geometry option. This array is only used for branch resistance option 3 (noncircular duct)
TIMEG( )	Time (s) array used in history files
NGSTEP	Number of lines of time-data information in a history file
ARNMB(,)	Nodal normal area (in^2) - time (s) array used in the input history file for the variable geometry option when the moving boundary option is used
VELMB(,)	Nodal normal velocity (ft/s) - time (s) array used in the input history file for the variable geometry option when the moving boundary option is used
AREANB( )	Area of the moving boundary (ft^2) of an internal node for the variable geometry option when the moving boundary option is used
TIMER( )	Time (s) used in history file for variable rotation option
RPMT( )	Rotational speed (RPM) used in history file for variable rotation option
NRSTEP	Number of data points in rotational history file

Variable	Description
13. COMMON/CPRESS/	**Common block for Pressurization variables**
NTANK	Number of pressurization tanks in a model
NODUL( )	Node number representing ullage in a given tank
NODULB( )	Pseudo boundary node representing the interface between ullage and propellant
NODPRP( )	Node number representing propellant tank pressure and temperature
IBRPRP( )	Branch number representing propellant flow rate
TNKAR( )	Tank surface area (ft^2) for heat transfer with ullage gas
TNKTH( )	Tank thickness (ft) for heat conduction calculation in tank wall
TNKRHO( )	Tank density (lb_m/ft^3) for heat conduction calculation in tank wall
TNKCP( )	Tank specific heat (Btu/lb_m-°R) for heat conduction calculation in tank wall
ELHC( )	Length scale (ft) for computing Grashoff number
ARHC( )	Surface area for heat transfer (ft^2) between ullage and propellant
FCTHC( )	Factor controlling the magnitude of heat transfer coefficient (Default value = 1)
TNKTM( )	Tank temperature (°R)
TNKTMM( )	Tank temperature (°R) at the previous time step
QULWAL( )	Heat transfer rate (Btu/s) between ullage and wall
QULPRP( )	Heat transfer rate (Btu/s) between ullage and propellant
EMDPRP( )	Not used
HFG( )	Not used
TSAT( )	Not used
TNKCON( )	Tank conductivity (Btu/ft-s-°R)
QCOND( )	Heat transfer rate (Btu/s) between ullage exposed tank surface and propellant exposed tank surface
ITTYPE( )	Index number to distinguish between type of propellant tank (1 – Cylindrical tank, 2 – Spherical tank)
CIP( )	Constant in the correlation for ullage to propellant heat transfer coefficient
FNIP( )	Power law index in the correlation for ullage to propellant heat transfer coefficient
CIW( )	Constant in the correlation for ullage to wall heat transfer coefficient
FNIW( )	Power law index in the correlation for ullage to wall heat transfer coefficient
ITTYPE( )	Index to define cylindrical or spherical tank
CIP	Constant for heat transfer correlation (ullage to propellant)
FNIP	Index for heat transfer correlation (ullage to propellant)
CIW	Constant for heat transfer correlation (ullage to wall)
FNIW	Index for heat transfer correlation(ullage to wall)

284

Variable	Description
14. COMMON/CVALVE/	**Common block for Control Valve variables**
DTAUIN	A temporary variable that stores the data file time step (sec) input by the user. Used by subroutine CTRLVLV when determining minimum time step
NVALVE	Number of control valves in the model
DVTAU( )	User prescribed time step (sec) for each sub-option 2 or 3 control valve
IVOPT( )	Sub-option of each control valve (1=instantaneous, 2=linear, 3=nonlinear
NOVDAT( )	Number of open characteristics data points for suboption 2 and 3 control valves. Used for reading open characteristics data
NCVDAT( )	Number of close characteristics data points for suboption 2 and 3 control valves. Used for reading close characteristics data.
OVTIM(,)	Time schedule (sec) for open valve characteristics for suboption 2 and 3 control valves. Used in calculating transient open valve characteristics
CVTIM(,)	Time schedule (sec) for close valve characteristics for suboption 2 and 3 control valves. Used in calculating transient close valve characteristics
OVCL(,)	Transient opening valve flow coefficient history for suboption 2 and 3 control valves. Used in calculating transient open valve characteristics
CVCL(,)	Transient closing valve flow coefficient history for suboption 2 and 3 control valves. Used in calculating transient close valve characteristics
OVAR(,)	Transient opening valve flow area history (in^2) for suboption 2 and 3 control valves. Used in calculating transient open valve characteristics
CVAR(,)	Transient closing valve flow area history (in^2) for suboption 2 and 3 control valves. Used in calculating transient close valve characteristics
VAREA( )	Flow area (ft^2) calculated in the subroutine CTRLVLV for each control valve. Used for calculating flow resistance for the control valve in subroutine KFACT18
VCL( )	Flow coefficient calculated in the subroutine CTRLVLV for each control valve. Used for calculating flow resistance for the control valve in subroutine KFACT18
TIMEV(,)	Time schedule (sec) for the pressure tolerance files for each control valve. Used for determining the state for each control valve (The states are fully open, fully closed, opening, closing)
PMAXV(,)	Maximum pressure tolerance (psia) history for each control valve. Used for determining the state of each control valve (The states are fully open, fully closed, opening, closing)
PMINV(,)	Minimum pressure tolerance (psia) history for each control valve. Used for determining the state of each control valve (The states are fully open, fully closed, opening, closing)
NVDAT( )	Number of pressure tolerance file data points for each control valve. Used in reading pressure tolerance data
15. COMMON/CCONV/	**Common block for conversion factor variables**
FACTP	Conversion factor for pressure (to convert to and from psf and psi, 144)

Variable	Description
FACTV	Conversion factor for Volume (to convert to and from in^3 and ft^3, 1728)
FACTA	Conversion factor for area (to convert to and from in^2 and ft^2, 144)
FACTL	Conversion factor for length (to convert to and from in and ft, 12)
FACTT	Conversion factor for area (to convert to and from °F and °R, 459.6)
FACTTH	Conversion factor for angle (to convert to and from degree to radian, 0.01745)
FACTVS	Not used
16. COMMON/CSICONV/	**Common block for SI conversion factor variables**
SIFACTP	Conversion factor for pressure (to convert to and from psf and kpa)
SIFACTV	Conversion factor for Volume (to convert to and from lbm/ft^3 and kgm/m^3)
SIFACTA	Conversion factor for area (to convert to and from m^2 and ft^2)
SIFACTL	Conversion factor for length (to convert to and from m and ft)
SIFACTT	Conversion factor for area (to convert to and from °C and °R)
SIFACTTH	Conversion factor for angle (to convert to and from degree to radian)
SIFACTVS	Not used
17. COMMON/CLOGIC/	**Common block for logical variables. NOTE: a declaration of these variables as logical variables must follow this common block**
STEADY	Logical variable to indicate if the model is steady state, quasi-steady (a series of steady state runs with changing boundary conditions and/or geometry; requires TRANSV to be false), or fully unsteady (requires TRANSV to be true)
DENCON	Logical variable to indicate if the model will use a user defined constant density fluid. NOTE: this option is ONLY valid for steady state and the energy equation will NOT be solved
GRAVITY	Logical variable to indicate if the model will account for gravity in branches where a branch length is associated
ENERGY	Logical variable to indicate if the model will solve one of the two forms of the energy equation within the code. Required for all fluids except when DENCON is true
MIXTURE	Logical variable to indicate if the model is using more than one fluid. NOTE: not valid if DENCON is true
CHOKED	Logical variable to indicate if the model will calculate choked flow
CHOK( )	Logical variable to indicate if an individual branch will have choked flow calculated
THRUST	Logical variable to indicate if the model will calculate thrust (lb_f) using pressure and thrust area (current formulation neglects thrust from linear inertia)
RESTART	Not used
TRANSV	Logical variable to indicate if the model will operate in an unsteady mode. NOTE: requires STEADY to be false

Variable	Description
INERTIA	Logical variable to indicate if the model will include linear inertia in the calculation of the momentum equation
CONDX	Logical variable to indicate if the model will calculate thermal conduction between nodes
TWOD	Not used
PRINTI	Logical variable to indicate if the main output file will contain the initial guess at the flow field
ROTATION	Logical variable to indicate if the model contains branches where rotation will contribute a momentum source/sink
ROTATE( )	Logical variable indicating if an individual branch will have rotation included as a momentum source/sink. NOTE: requires ROTATION to be true
BUOYANCY	Logical variable to indicate if the model will consider buoyancy (density variation) effects in a gravity field. NOTE: requires GRAVITY to be true
HRATE	Logical variable to indicate if heat sources are in BTU/sec (true) or BTU/lb$_\text{m}$ (false)
INVAL	Logical variable to indicate if the model will read in previously saved data from two restart files for node and branch data
SAVER	Logical variable to indicate if the model will write data into two restart files to be used for later restarting of the model
HEX	Logical variable to indicate if the model includes heat exchangers
MSORCE	Logical variable to indicate if the model contains additional momentum sources (in addition to pumps)
MOMSOR( )	Logical variable to indicate which branches have additional momentum sources. NOTE: requires MSORCE to be true
HCOEF	An additional option for heat exchanger calculation; if true UA is alculated, otherwise it must be specified
MOVBND	Logical variable to indicate if the model contains nodes which have a moving boundary
MVBND( )	Logical variable to indicate which nodes contain a moving boundary. NOTE: requires MOVBND to be true
REACTING	Option for activating chemical reaction. NOTE: equilibrium reaction of hydrogen and oxygen is only available
REACTION( )	Logical variable indicating if an individual node will have chemical reaction
TPA	Logical variable to indicate if the model includes turbopump assemblies
TPABR( )	Logical variable indicating if an individual branch represents pump or turbine
ELASTIC	Not used
VARGEO	Logical variable to indicate if the model will consider variable geometries (time dependent geometries). NOTE: requires STEADY to be false

Variable	Description
TVM	Logical variable to indicate if the model will consider the transverse component of inertia in the momentum equation (transverse momentum)
TRNSM( )	Logical variable to indicate which branches will consider transverse momentum. NOTE: requires TVM to be true
SHEARE	Logical variable to indicate if the model will consider shear stress instead of using a friction factor on at least one branch within the model
SHER( )	Logical variable to indicate which branches will consider shear stress instead of friction. NOTE: requires SHEARE to be true
ADDPROP	Logical variable to indicate if the model will use a fluid thermodynamic/theromphysical property package other than that which is already incorporated into the code. When ADDPROP is false, GASP and WASP are used; when GASPAK is true, the commercially available code GASPak is required (along with a licence)
PRNTIN	Logical variable to indicate if the main output file will contain the input variables
PRNTADD	Logical variable to indicate if the main output file will contain additional thermodynamic output data for each internal node
PRESS	Logical variable to indicate if the model will contain pressurization of a tank. NOTE: requires STEADY to be false and TRANSV to be true
INSUC	Logical variable to activate calculation of initial guess by using a successive substitution method
VARROT	Logical variable to indicate if the model will have time dependent rotation. NOTE: requires ROTATION to be true, ROTATE to be true for at least one branch and STEADY to be false
VOPEN( )	Logical variable that stores the initial position of the valve (T=Open, F=Closed). Used in initializing valve settings
VFLOW( )	A logical variable that indicates whether or not there is flow through a control valve (T=CL & A >0, F=CL & A = 0). Used in flow resistance calculations by subroutine KFACT18
INERT( )	Logical variable to activate longitudinal inertia calculation in a given branch
NORMAL	Logical variable to activate normal stress calculation in a given branch
SIMULA	Logical variable to indicate if the user would like the model to be solved using a totally simultaneous solution scheme or a modified scheme (inner & outer loop for first five iterations of a given time step, then simultaneous)
SIMUL	Logical variable set by SIMULA at the beginning of each time step. See description of SIMULA
SECONDL	Logical variable to indicate if the model will solve the energy equation using the first law of thermodynamics (false, uses enthalpy) or the second law of thermodynamics (true, uses entropy)
FRICTBP	Logical variable used to override SIMULA and bypass the simultaneous solution scheme

288

Variable	Description
USETUP	Logical variable to allow the user to customize the input deck. Used in association with the USRSET subroutine (User Subroutine)
LAMINAR	Logical variable used in conjunction with the SHEARE logical variable. When set to false, shear stress is calculated using a modified Prandtl mixing length model for branch to branch interaction and the log-law of the wall for branch to solid interaction. When set to true, shear stress is calculated from the derivative of the velocity and the fluid viscosity
TRANSQ	Logical variable used to identify that heat addition will vary with time using user supplied data in history file(s). Requires TRANSV to be true
TRQ( )	Logical variable used to identify which nodes will have a time variant heat added. Requires TRANSQ to be true
DFLI	Logical variable used to identify between two formulations for longitudinal inertia (requires INERTIA to be active). If set to true in a User Subroutine, the differential form of longitudinal inertia will be active, else the original formulation will be active
CONJUG	Logical variable to indicate if the model will calculate solid to fluid heat transfer
DALTON	Logical variable to indicate if the model will calculate mixture property by Dalton's law of partial pressure
HYDPOX	Logical variable to indicate if hydrogen peroxide is the working fluid
OPVALVE	Logical variable to indicate if there is any opening or closing of valve
RADIATION	Logical variable to indicate if model needs radiation heat transfer
Winplot	Logical variable to indicate if Winplot will be used to plot data
HSTAG	Logical variable to indicate if the model uses stagnation (TRUE) or static (FALSE) enthalpy
VISCWRK	Logical variable to indicate if viscous work is computed in energy equation
CYCLIC	Logical variable to indicate if cyclic boundary condition is active
REPEAT	Logical variable to indicate if the iterative calculation to be repeated to satisfy adjustable boundary condition
NRSOLVT	Logical variable to indicate if solid temperature equation will be solved by Newton-Raphson method. NRSOLVT=.FALSE. indicates that solid temperature equation is solved by successive substitution method
HCGIVEN	Not used
CHKVAL	Logical variable to indicate if calculated values need to be checked for debugging purpose

Variable	Description
WFILE	Logical variable to indicate if data dump for Winplot is in ASCII(WFILE=.TRUE.) or binary (WFILE=.FALSE.)
UPP	Universal Property Package
NOSTATS	Logical variable to print file open error
NOPLT	Logical variable to activate VTASC plotting package
PRESSREG	Not used
FLOWREG	Not used
TRANS_MOM	Logical variable to activate transient momentum term
USRVAR	Logical variable to print user variable in Winplot
ISHT	Logical variable to activate Inter Species Heat Transfer
TRANS_HMIX	Not used
NEWTON_SS	Logical variable to activate Newton-Raphson Solver
BROYDEN_SS	Logical variable to activate Broyden Solver
PSM( )	ogical variable for Phase Separation Model at node
SATURATED( )	Logical variable to indicate saturation state at node
PENERGY	Logical variable to activate potential energy
PSMG	Logical variable for global activation of PSM
RADSA( )	Logical variable for solid to ambient radiation
RADSF( )	Logical variable for solid to fluid radiation
SATMIX(,)	Logical variable to indicate saturation in mixture
PLOTADD	Logical variable to print additional variables
SIUNITS	Logical variable to activate input in SI units
TECPLOT	Logical variable to activate writing data for Tecplot
GRIDGEN	Logical variable to activate grid generation
PRINTD	Logical variable for diagnostic printout
FLUIDMIX( )	Not used
MSORIN	Global logical variable for momentum source
MSORINI( )	Local logical variable for local momentum source
SATTABL	Not used
RLFVLV	Logical variable for Relief Valve
18. COMMON/CUSER/	**Common block for user defined variables. For use in the User Subroutines**
SORCEMAS( )	User defined mass source. Usually defined in the SORCEM User Subroutine
SORCEMOM( )	User defined momentum source. Usually defined in the SORCEF User Subroutine
SORCECON(,)	User defined specie concentration source. Usually defined in the SORCEC User Subroutine
SORCEH( )	User defined heat source. Usually defined in the SORCEH user subroutine
USRVAR1( )	User defined one dimensional variable. Used to pass information between different User Subroutines
USRVAR2( )	User defined one dimensional variable. Used to pass information between different User Subroutines

Variable	Description
USRVAR3( )	User defined one dimensional variable. Used to pass information between different User Subroutines
USRVAR4( )	User defined one dimensional variable. Used to pass information between different User Subroutines
USRVAR5( )	User defined one dimensional variable. Used to pass information between different User Subroutines
USRVAR6( )	User defined one dimensional variable. Used to pass information between different User Subroutines
USRVAR7( )	User defined one dimensional variable. Used to pass information between different User Subroutines
USRVAR8( )	User defined one dimensional variable. Used to pass information between different User Subroutines
USRVAR9( )	User defined one dimensional variable. Used to pass information between different User Subroutines
USRVAR10( )	User defined one dimensional variable. Used to pass information between different User Subroutines
USRVAR11(,)	User defined two dimensional variable. Used to pass information between different User Subroutines
USRVAR12(,)	User defined two dimensional variable. Used to pass information between different User Subroutines
USRVAR13(,)	User defined two dimensional variable. Used to pass information between different User Subroutines
USRVAR14(, ,)	User defined three dimensional variable. Used to pass information between different User Subroutines
ITERADJU	Iteration counter for adjustable boundary condition iteration loop
19. COMMON/OPVALVE/	**Common block for valve opening or closing**
AREAV(,)	Variable valve area during opening or closing in the valve history file (ft^2)
NVOCBR( )	Branch names that represent variable area valve
NVOPBR	Number of valves in a circuit where valve area changes with time
NVOP( )	Number of data points in valve opening/closing history file
TIMEVOP(,)	Time in valve history file (s)
20. COMMON/CHT/	**Common block for conjugate heat transfer index variables**
NSOLIDX	Number of solid nodes
NAMB	Number of ambient nodes
NSAC	Number of solid to ambient conductors
NSFC	Number of solid to fluid conductors
NSSC	Number of solid to solid conductors
NSSR	Number of solid to solid radiation conductors

Variable	Description
21. COMMON/SNODE/	**Common block for solid node properties**
CPSLD( )	Specific heat of solid node (Btu/lb$_m$-°R)
CPSLDM( )	Specific heat of solid node (Btu/lb$_m$-°R) at previous time step
MATRL( )	Index number to indicate material of solid node
NAMESA(,)	Names of solid to ambient conductors connected to a solid node
NAMESF(,)	Names of solid to fluid conductors connected to a solid node
NAMESS(,)	Names of solid to solid conductors connected to a solid node
NAMESSR(,)	Names of solid to solid radiation conductors connected to a solid node
NODESL( )	Names of the solid node
NUMSA( )	Number of solid to ambient conductors
NUMSF( )	Number of solid to fluid conductors
NUMSS( )	Number of solid to solid conductors
NUMSSR( )	Number of solid to solid radiation conductors
SHSORC( )	External heat source to solid node (Btu/s)
SMASS( )	Mass of solid node (lb$_m$)
TS( )	Temperature of solid node (°R)
TSM( )	Temperature of solid node (°R) at previous time step
TSMM( )	Temperature of solid node (°R) at two steps prior
CPSLDMM( )	Specific heat of solid node (Btu/lb$_m$-°R) at two steps prior to current time step
22. COMMON/ANODE/	**Common block for ambient node properties**
NODEAM( )	Name of Ambient node
TAMB( )	Ambient Temperature (°R)
IHIST( )	Index variable for ambient node has history file
23. COMMON/SSCOND/	**Common block for solid to solid conductor properties**
ARCSIJ( )	Conduction area between neighboring solid nodes (ft^2)
CONDKIJ( )	Conductivity between neighboring solid nodes (Btu/ft-s-°R)
DISTSIJ( )	Distance between neighboring solid nodes (ft)
EFCSIJ( )	Effective conductance between neighboring solid nodes (Btu/s-°R)
ICNSI( )	Upstream solid node of solid to solid conductor
ICNSJ( )	Downstream solid node of solid to solid conductor
ICONSS( )	Name of solid to solid conductor
QDOTSS( )	Heat transfer through a solid conductor (Btu/s)
24. COMMON/SFCOND/	**Common block for solid to fluid conductor properties**
ARSF( )	Heat transfer area between solid and fluid nodes (ft^2)
EFCSF( )	Effective conductance between solid and fluid node (Btu/s-°R)
EMSFF( )	Emissivity of fluid

292

Variable	Description
EMSFS( )	Emissivity of solid
HCSF( )	Heat transfer coefficient between solid and fluid nodes (Btu/ft^2-s-°R)
HCSFR( )	Radiation heat transfer coefficient between solid and fluid (Btu/ft^2-s-°R)
ICF( )	Name of connecting fluid nodes
ICONSF( )	Name of solid to fluid conductors
ICS( )	Name of connecting solid nodes
MODEL( )	Index to specify which heat transfer coefficient model to use; MODEL(I)=0 – (User Specified), =1 – (Dittus-Boelter), =2 - Miropolosky
QDOTSF( )	Heat transfer between solid and fluid node (Btu/s)
25. COMMON/SACOND/	**Common block for solid to ambient conductor properties**
ARSA( )	Heat transfer area between solid and ambient nodes (ft^2)
EFCSA( )	Effective conductance between solid and ambient node (Btu/s-°R)
EMSAA( )	Emissivity of ambient
EMSAS( )	Emissivity of solid
HCSA( )	Heat transfer coefficient between solid and ambient nodes (Btu/ft^2-s-°R)
HCSAR( )	Radiation heat transfer coefficient between solid and ambient nodes (Btu/ft^2-s-°R)
ICONSA( )	Name of solid to ambient conductors
ICSAA( )	Name of connecting ambient nodes
ICSAS( )	Name of connecting solid nodes
QDOTSA( )	Heat transfer between solid and ambient node (Btu/s)
26. COMMON/SSRADC/	**Common block for solid to solid radiation conductor properties**
ARRSI( )	Area of ith solid node for radiation calculation (ft^2)
ARRSJ( )	Area of jth solid node for radiation calculation (ft^2)
EFCSSR( )	Effective radiation conductance between solid and ambient node (Btu/s-°R)
EMSSI( )	Emissivity of ith solid node
EMSSJ( )	Emissivity of jth solid node
ICONSSR( )	Name of solid to solid radiation conductors
ICNSRI( )	Name of ith solid node connecting radiation conductor
ICNSRJ( )	Name of jth solid node connecting radiation conductor
QDOTSSR( )	Heat transfer in solid to solid radiation conductor (Btu/s)
27. COMMON/CYCLIC/	**Common block for cyclic boundary condition**
NDCYCLB	Name of cyclic boundary node
NDCYCLU	Name of internal node upstream of cyclic boundary

Variable	Description
ITERADJC	Integer count for iteration loop for cyclic boundary calculation
DIFTEM	Normalized difference in temperature between upstream and downstream node of cyclic boundary condition
28. COMMON/PRESSREG/	**Common block for pressure regulator properties**
IBRPRG( )	Branch number representing pressure regulator
IPHIST( )	Index variable to indicate if there is a history file
AREAMAXP( )	Maximum allowable area of pressure regulator
REQPRESS( )	Required pressure downstream of pressure regulator
RELAXPR( )	Relaxation parameter of pressure regulator
CCPRG( )	Convergence criterion of pressure regulator
PRGHIST( )	Name of pressure regulator history file
PREGTIME(,)	Time array in pressure regulator history file
NPRGVALS()	Number of lines in pressure regulator history file
PREGPRESS(,)	Pressure array in pressure regulator history file
IREGTYP	Not used
NUMPREGS	Number of pressure regulators in the circuit
IPRESREG	Index to define the type of pressure regulator
MAXITER( )	Maximum number of allowable iterations
AREAMINP( )	Minimum allowable area of pressure regulator
29. COMMON/FLOWREG/	**Common block for flow regulator properties**
IBRFRG( )	Branch number representing flow regulator
IFHIST( )	Index variable to indicate if there is a history file
AREAMAXF( )	Maximum allowable area of pressure regulator
REQFLOW( )	Required flow rate in flow regulator
RELAXFR( )	Relaxation parameter of flow regulator
CCFRG( )	Convergence criterion of flow regulator
FRGHIST( )	Name of flow regulator history file
FREGTIME(,)	Time array in flow regulator history file
NFRGVALS()	Number of lines in flow regulator history file
FREGFLOW(,)	Flow rate array in flow regulator history file
NUMFREGS	Number of flow regulators in the circuit
IFLOWREG	Index to define the type of flow regulator
30. COMMON/GPATH/	
GPATH( )	Maximum number of allowable iterations
APATH( )	Minimum allowable area of pressure regulator
MODELNAME( )	Name of the model input data file

294

Variable	Description
31. COMMON/MATPROPS/	**Common block for material properties**
MAXMATLS	Maximum number of materials allowed
NUMMATLS	Number of materials read from the model
MATL( )	Array of material numbers read from the model
MATLCCT( )	Number of entries (records) in the Conductivity file
MATLCPCT( )	Number of entries (records) in the Specific Heat file
TEMPC(,)	Temperature profile in Conductivity file
TCONDC(,)	Conductance profile in Conductivity file
TEMPS(,)	Temperature profile in Specific Heat file
CPSC(,)	Specific Heat profile in Specific Heat file
32. COMMON/USERPVARS/	**Common block for user specified variables**
USERPVARS	Value of the user variable to be plotted
USRVARSNUM	Number of user variables to be plotted
33. COMMON/USERNVARS/	**Common block for user specified variables**
USRPVARNAME	Name of the user variable to be plotted
USRPVARUNIT	Unit of the user variable to be plotted
34. COMMON/CMIXTURE/	**Common block for mixture properties**
ARFACT	Factor of area calculation for inter-phase heat transfer
CONDUCTX(,)	Thermal conductivity of mixture component
CPX(,)	Sp. Heat at const. pressure of mixture component
CVX(,)	Sp. Heat at const. volume of mixture component
EMX(,)	Mass of mixture component
EMUX(,)	Viscosity of mixture component
GAMAX(,)	Specific heat ratio of mixture component
HSORCEX(,)	Heat source for mixture component
HX(,)	Enthalpy of mixture component
HXM(,)	Enthalpy of mixture component at previous time
QHES(,)	Inter-phase heat transfer between mixture component
HXMM(,)	Enthalpy of mixture component at two time steps prior
PX(,)	Partial Pressure of Mixture Component
RHOX(,)	Density of mixture component
RHOXM(,)	Density of mixture component at previous time step
RHOXMM(,)	Density of mixture component at two time steps prior
SORCEHX(,)	Energy Source for mixture component
SX(,)	Entropy of mixture component
TFX(,)	Temperature of mixture component

Variable	Description
VOLX(,)	Volume of mixture component
XVX(,)	Vapor quality of mixture component
ZX(,)	Compressibility factor of mixture component
PXM(,)	Partial pressure of mixture component at previous time step
PXMM(,)	Partial pressure of mixture component at two times steps prior
EMXM(,)	Mass of mixture component at previous time step
EMXMM(,)	Mass of mixture component at two time steps prior
NAF(,)	Index variable used for Inter-phase heat transfer
CMV(,)	Not used
CML(,)	Not used
SXM(,)	Not used
SGENX(,)	Not used
SXREF(,)	Not used
TFXM(,)	Not used
RHOLMIX(,)	Not used
RHOVMIX(,)	Not used
EMULMIX(,)	Not used
EMUVMIX(,)	Not used
CPNODEVMIX(,)	Not used
CPNODELMIX(,)	Not used
CONDFVMIX(,)	Not used
CONDFLMIX(,)	Not used
ITEMPMIX	Not used
35. COMMON/CCART/	**Common block for cartesian properties**
NODEX( )	Not used
NODEY( )	Not used
NODEZ( )	Not used
XDIST( )	Not used
YDIST( )	Not used
ZDIST( )	Not used
INDNAB( )	Not used
XU( )	Not used
YV( )	Not used
ZW( )	Not used
IBRVEL( )	Not used
DXU( )	Not used
DYV( )	Not used
DZW( )	Not used

Variable	Description
NBRNX( )	Not used
NBRNY( )	Not used
NBRNA( )	Not used
NX	Number of nodes in x-direction
NY	Number of nodes in y-direction
NZ	Number of nodes in z-direction
XLEN	Length in x-direction
YLEN	Length in y-direction
ZLEN	Length in z-direction
DXP( )	Not used
DYP( )	Not used
DZP( )	Not used
ANORTH( )	Not used
ASOUTH( )	Not used
AEAST( )	Not used
AWEST( )	Not used
TNORTH( )	Not used
TSOUTH( )	Not used
TEAST( )	Not used
TWEST( )	Not used
36. COMMON/CSPARSE/	**Common block for sparse matrix properties**
MAXL	Not used
MT	Not used
IROW( )	Not used
JCOL(,)	Not used
AS( )	Not used
MTMAXS	Not used
37. COMMON/TECPLOT/	**Common block for tecplot properties**
XDISTFN( )	x-coordinate of node
YDISTFN( )	y-coordinate of node
ZDISTFN( )	z-coordinate of node
XDISTB( )	x-coordinate of branch
YDISTB( )	y-coordinate of branch
ZDISTB( )	z-coordinate of branch
XDISTSN( )	Not used
YDISTSN( )	Not used
ZDISTSN( )	Not used

Variable	Description
XDISTAN( )	Not used
YDISTAN( )	Not used
ZDISTAN( )	Not used
UVELN( )	u-velocity component of node
VVELN( )	v-velocity component of node
38. COMMON/FIXEDFLOW/	**Common block for fixed flow option**
FFHIST( )	History file name for fixed flow rate option
NFIXFLB	Number of fixed flow branch
AREAFF(,)	Not used
TIMEFF(,)	Time array in fixed flow rate history file
FFLOW(,)	Flow rate in fixed flow rate history file
NDATFF( )	Number of data in fixed flow rate history file
39. COMMON/RLFVLV/	**Common block for relief valve properties**
NRLFVLV	Number of relief valve in the flow circuit
IRLFVLVBR( )	Branch ID number of a relief valve
NRLFVLVVALS( )	Number of A(dP) or Cv(dP) pairs in the relief valve control file
RLFVLVDP(,)	Pressure differential points (dP) in the relief valve control file
RLFVLVACV(,)	Area (A) or Cv points in the relief valve control file
RLFVLVPCRACK( )	Cracking pressure for the relief valve
RLFVLVHIST( )	Filename of the relief valve control file

APPENDIX E—LISTING OF BLANK USER SUBROUTINES

```
C************************************************************************
C                                    *
C   ***** GFSSP User SubroutineS *****              *
C                                    *
C************************************************************************
C  SUBROUTINE USRINT IS CALLED FROM INIT TO SPECIFY INITIAL VALUES COMPUTED
C       BY USER SPECIFIED THERMODYNAMIC PROPERTY PACKAGE
C
C  SUBROUTINE SORCEM(IPN,TERMU) IS CALLED FROM EQNS FOR MASS SOURCES.
C       IN THIS ROUTINE THE USER DEFINES ANY ADDITIONAL MASS
C       SOURCES TO THE MODEL (MASS SOURCES ARE IN LBM/SEC). USER
C       CAN MODIFY TRANSIENT TERM BY REDEFINING THE ARGUMENT TERMU.
C
C  SUBROUTINE SORCEF(I,TERM0,TERM1,TERM2,TERM3,TERM4,TERM5,TERM6,TERM7,
C       TERM8,TERM9,TERM10,TERM100) IS CALLED FROM EQNS FOR
C       MOMENTUM SOURCES. USER CAN MODIFY INDIVIDUAL TERMS OR
C       DEFINE ADDITIONAL MOMENTUM SOURCES THROUGH TERM100.
C
C  SUBROUTINE SORCEQ IS CALLED FROM EITHER THE ENERGY ROUTINE (EITHER
C       ENTHALPY OR ENTROPY). IN THIS ROUTINE THE USER DEFINES
C       ANY ADDITIONAL HEAT SOURCES TO THE MODEL (HEAT SOURCES
C       ARE IN BTU/SEC)
C
C  SUBROUTINE SORCEC IS CALLED FROM THE SPECIES CONCENTRATION ROUTINE
C       IN THIS ROUTINE THE USER DEFINES ANY ADDITIONAL SPECIES
C       CONCENTRATION SOURCES TO THE MODEL (CONCENTRATION SOURCES
C       ARE IN MASS FRACTIONS SUCH THAT THE SUM OF ALL OF THE
C       CONCENTRATIONS EQUALS 1.0)
C
C  SUBROUTINE KFUSER IS CALLED FROM THE RESIST ROUTINE. IN THIS ROUTINE
C       THE USER DEFINES ANY VARIATION OF THE K-FACTOR OF A BRANCH
C       SUCH THAT THE K-FACTOR IS DEFINED AS THE PRESSURE DROP
C       DIVIDED BY THE MASS FLOW RATE^2 (PRESSURE IS IN PSF, FLOW
C       RATE IS IN LBM/SEC; I.E. THE K-FACTOR IS IN PSF-SEC^2/
C       (LBM-FT)^2)
C
C  SUBROUTINE PRPUSER IS CALLED FROM THE DENSITY ROUTINE. IN THIS
C       ROUTINE THE USER ADDS OR MODIFIES FLUID PROPERTIES (ALLOWS
C       FOR USER SPECIFIED FLUID)
C
C  SUBROUTINE TSTEP IS CALLED FROM THE MAIN ROUTINE. IN THIS ROUTINE
C       THE USER CAN MODIFY THE TIMESTEP, DTAU, FOR AN UNSTEADY
C       MODEL (DTAU IS IN SECONDS)
C
C  SUBROUTINE BNDUSER IS CALLED FROM THE BOUND ROUTINE. IN THIS ROUTINE
C       THE USER CAN MODIFY BOUNDARY CONDITIONS AND GEOMETRY AT
C       EACH TIMESTEP FOR AN UNSTEADY MODEL (PRESSURE IS IN PSF,
C       TEMPERATURE IS IN DEG. R, LENGTH {ETC.} IS IN FT, AREA IS
C       IN FT^2, VOLUME IS IN FT^3)
C
C  SUBROUTINE PRNUSER IS CALLED FROM THE PRINT ROUTINE. IN THIS ROUTINE
C       THE USER CAN MODIFY ADD ADDITIONAL OUTPUT FILES SPECIFIC
C       TO A PARTICULAR MODEL
C
C  SUBROUTINE FILNUM IS CALLED FROM THE MAIN ROUTINE. IN THIS ROUTINE
C       ESTABLISHES THE FILE NUMBERS THAT ARE TO BE OPENED FOR ALL
C       FILES IN GFSSP, AND INCLUDES 10 USER FILE NUMBERS FOR USE
C       IN THE PRNUser Subroutine
C
```

```
C  SUBROUTINE USRSET IS CALLED FROM THE READIN ROUTINE. IN THIS ROUTINE
C       THE USER SETS UP THE MAJORITY OF THE MODEL; ONLY A DUMMY
C       SEGMENT OF AN INPUT FILE IS NECESSARY TO BE READ, WITH THE
C       REMAINDER OF THE MODEL SETUP IN THIS SUBROUTINE.
C
!  THE FOLLOWING FUNCTIONS ARE AVAILABLE FOR CONVERTING SI UNITS INTO GFSSP INTERNAL (ENGLISH) UNITS.
!
!   FUNCTION rlbmft3_kgm3(rlbmft3)
!   PURPOSE: to convert LBM/FT^3 to KG/M^3

!   FUNCTION PSF_KPA(PSF)
!   PURPOSE: to convert PSF to KPA

!   FUNCTION RLBMS_KGS(RLBMS)
!   PURPOSE: to convert LBM/SEC TO KG/SEC

!   FUNCTION rKGS_LBMS(rKG)
!   PURPOSE: to convert KG/SEC LBM/SEC

!   FUNCTION FT_M(FT)
!   PURPOSE: to convert FT to METERS

!   FUNCTION rM2_FT2(rM2)
!   PURPOSE: to convert METERS^2 to FT^2

!   FUNCTION BTULB_KJKGK(BTU)
!   PURPOSE TO CONVERT BTU/LB TO KJ/KG K

!   FUNCTION BTULBMR_KJKGK(BTU)
!   PURPOSE TO CONVERT BTU/LBM R TO KJKG K

!   FUNCTION rLBMFTS_KGMS(rLBM)
!   PURPOSE TO CONVERT LBM/FT S TO KG/M S

!   FUNCTION BTUFTSR_WMK(BTU)
!   PURPOSE TO CONVERT BTU/FT-S R TO W/M K

!   FUNCTION FTS_MS(FT)
!   PURPOSE TO CONVERT FT/SEC TO M/SEC

!   FUNCTION R_C(T,SIFACTT)
C   PURPOSE: CONVERSION OF UNIT FOR TEMPERATURE

!   FUNCTION C_R(T,SIFACTT)
C   PURPOSE: CONVERSION OF UNIT FOR TEMPERATURE

!   REAL FUNCTION KW_BTUS(KW)
C   PURPOSE: TO CONVERT kW TO BTU/sec (HEAT RATE)

!   REAL FUNCTION KJKG_BTULBM(KJKG)
C   PURPOSE: TO CONVERT kJ/kg to BTU/lbm (HEAT RATE)

!   REAL FUNCTION KGM3_LBMF3(KGM3)
C   PURPOSE: TO CONVERT kg k/m^3 to lbm/ft^3 (DENSITY)

!   REAL FUNCTION NSM2_LBMFS(NSM2)
C   PURPOSE: TO CONVERT ns/m^2 to lbm/f s (VISCOSITY)

!   FUNCTION WMK_BTUFSR(WMK)
C   PURPOSE: TO CONVERT W/m K to BTU/f s R (THERMAL CONDUCTIVITY)

!   FUNCTION WM2K_BTUF2SF(WM2K)
C   PURPOSE: TO CONVERT W/m^2 K to BTU/f^2 s F (HEAT TRANSFER COEFFICIENT)

!   REAL FUNCTION KGS_LBMS(KGS)
C   PURPOSE: TO CONVERT kg/s to lbm/s (MASS)

!   REAL FUNCTION M_F(M)
```

300

```
C    PURPOSE: TO CONVERT meters to feet (VELOCITY)


!    FUNCTION W_BTUS(W)
C    PURPOSE: TO CONVERT watts to btu/S ( HEAT RATE)


!    REAL FUNCTION KJ_BTU(KJ)
C    PURPOSE: TO CONVERT kJ to BTU (ENERGY)


!    REAL FUNCTION KJKGK_BTULBMR(KJKGK)
C    PURPOSE: TO CONVERT kJ/kg-K to BTU/lbm-R (SPECIFIC HEAT)


C*************************************************************************
     SUBROUTINE FILENUM
C    PURPOSE: ESTABLISH THE FORTRAN FILE NUMBERS FOR READING &
C        WRITING OF INFORMATION
C*************************************************************************
     INCLUDE 'comblk.for'
C*************************************************************************
C
C     FILES ALREADY WITHIN GFSSP
C
C     NWRTE = FILE # CORRESPONDING TO THE WRITEIN SUBROUTINE
C        (WRITING INPUT DECK FROM COMMAND LINE PREPROCESSOR)
C     NPRNT = FILE # CORRESPONDING TO THE PRINT SUBROUTINE
C        (WRITING THE MAIN OUTPUT FILE)
C     NREAD = FILE # CORRESPONDING TO THE READIN SUBROUTINE
C        (READING IN THE INPUT DECK)
C     NGSPK = FILE # CORRESPONDING TO A NonGASP PROPERTY PACKAGE
C     NFNOD = FILE # CORRESPONDING TO THE FNODE RESTART FILE
C     NGFSOUT = FILE # CORRESPONDING TO THE GFSSP.OUT FILE
C        (DEBUGGING FILE)
C     NFBR = FILE # CORRESPONDING TO THE FBRANCH RESTART FILE
C     NGASP = FILE # CORRESPONDING TO THE GASP.OUT FILE
C        (DEBUGGING FILE)
C     NHSTN = FILE # CORRESPONDING TO THE HISTN.XLS FILE
C     NHSTB = FILE # CORRESPONDING TO THE HISTBR.XLS FILE
C     NHSTF = FILE # CORRESPONDING TO B.C. & VARGEO HISTORY FILES
C     NCVHST = FILE # CORRESPONDING TO THE CONTROL VALVE HISTORY FILE
C     NCVCHR1 = FILE # CORRESPONDING TO THE FIRST OF TWO CONTROL
C        VALVE FILES
C     NCVCHR2 = FILE # CORRESPONDING TO THE SECOND OF TWO CONTROL
C        VALVE FILES
C     NHSTROT = FILE # CORRESPONDING TO THE VARIABLE ROTATION
C        HISTORY FILE
C     NERROR = FILE # CORRESPONDING TO THE ERROR.XLS FILE
C     NRP1DAT = FILE # CORRESPONDING TO THE RP1 PROPERTY DATA FILES
C     NDPLT = FILE # CORRESPONDING TO NODE RESULTS FOR VTASC POST-PROCESSING
C     NBRPLT = FILE # CORRESPONDING TO NODE RESULTS FOR VTASC POST-PROCESSING
C     NDWINP = FILE # CORRESPONDING TO NODE RESULTS FOR Winplot POST-PROCESSING
C     NBRWINP = FILE # CORRESPONDING TO BRANCH RESULTS FOR Winplot POST-PROCESSING
C     NCOND = FILE # CORRESPONDING TO THERMAL CONDUCTIVITY PROPERTY DATA
C     NCP = FILE # CORRESPONDING TO SPECIFIC HEAT PROPERTY DATA
C     NSLDPLT = FILE # CORRESPONDING TO SOLID NODE RESULTS FOR PLOTTING
C     NSSCPLT = FILE # CORRESPONDING TO SOLID TO SOLID CONDUCTOR RESULTS FOR PLOTTING
C     NSFCPLT = FILE # CORRESPONDING TO SOLID TO FLUID CONDUCTOR RESULTS FOR PLOTTING
C     NSACPLT = FILE # CORRESPONDING TO SOLID TO AMBIENT CONDUCTOR RESULTS FOR PLOTTING
C
c THESE ASSIGNMENTS DUPLICATE THOSE FOUND IN GFSSP505.FOR
C        NGSPK=1
C        NPRNT=10
C        NFNOD=11
C        NGFSOUT=12
C        NFBR=13
C        NREAD=15
C        NGASP=17
C        NHSTN=18
C        NHSTB=19
```

```
C       NWRTE=20
C       NHSTF=21
C       NCVHST=28
C       NCVCHR1=29
C       NCVCHR2=30
C       NHSTROT=35
C                NVOPFILE=36
C                NPRGFILE=37
C                NFRGFILE=38
C       NERROR=55
C       NRP1DAT=51
C    NDPLT=52
C    NBRPLT=53
C    NDWINP=54
C    NBRWINP=56
C    NCOND = 57
C    NCP = 58
C    NSLDPLT = 59
C    NSSCPLT = 60
C    NSFCPLT = 61
C    NSACPLT = 62
C    NSSRCPLT = 67
C        NSSCRWIN = 68
C    NBWPLOT = 70
C        NUVWINP = 71
C    NTECPLT = 72
C    NFIXFHST = 73
C  FILE NUMBERS FOR USER DEFINED FILES (THESE FILES CAN BE USED
C  IN ANY OF THE User SubroutineS; HOWEVER, MOST LIKELY USE IS
C  IN THE PRNUser Subroutine). COMMENT OUT FILE NUMBERS NOT IN USE.
C
C    NUSR1=
C    NUSR2=
C    NUSR3=
C    NUSR4=
C    NUSR5=
C    NUSR6=
C    NUSR7=
C    NUSR8=
C    NUSR9=
C    NUSR10=

   RETURN
   END
C************************************************************************
   SUBROUTINE USRINT
C  PURPOSE: PROVIDE INITIAL CONDITIONS WHEN ALTERNATE THERMODYNAMIC
C       PROPERTY PACKAGE IS USED
C************************************************************************
   INCLUDE 'comblk.for'
C************************************************************************
C************************************************************************
C  ADD CODE HERE
   RETURN
   END
C************************************************************************
   SUBROUTINE SORCEM(IPN,TERMU)
C  PURPOSE: ADD MASS SOURCES
C  IPN - GFSSP INDEX NUMBER FOR NODE
C  TERMU - UNSTEADY TERM IN MASS CONSERVATION EQUATION
C************************************************************************
   INCLUDE 'comblk.for'
C************************************************************************
C  ADD CODE HERE
   RETURN
   END
C************************************************************************
```

302

```fortran
      SUBROUTINE SORCEF(I,TERM0,TERM1,TERM2,TERM3,TERM4,TERM5,TERM6,
     &          TERM7,TERM8,TERM9,TERM10,TERM100)
C   PURPOSE: ADD MOMENTUM SOURCES (LBF)
C   I - GFSSP INDEX NUMBER FOR BRANCH
C   TERM0 - UNSTEADY TERM IN MOMENTUM CONSERVATION EQUATION
C   TERM1 - LONGITUDINAL INERTIA
C   TERM2 - PRESSURE GRADIENT
C   TERM3 - GRAVITY FORCE
C   TERM4 - FRICTION FORCE
C   TERM5 - CENTRIFUGAL FORCE
C   TERM6 - EXTERNAL MOMETUM SOURCE DUE TO PUMP
C   TERM7 - MOMENTUM SOURCE DUE TO TRANSVERSE FLOW(MULTI-DIMENSIONAL MODEL)
C   TERM8 - MOMENTUM SOURCE DUE TO SHEAR(MULTI-DIMENSIONAL MODEL)
C   TERM9 - VARIABLE GEOMETRY UNSTEADY TERM
C   TERM10 - NORMAL STRESS
C   TERM100 - USER SUPPLIED MOMENTUM SOURCE
C*************************************************************************
      INCLUDE 'comblk.for'
C*************************************************************************
C   ADD CODE HERE

      RETURN
      END
C*************************************************************************
      SUBROUTINE SORCEQ(IPN,TERMD)
C   PURPOSE: ADD HEAT SOURCES
C   IPN - GFSSP INDEX NUMBER FOR NODE
C   TERMD - COMPONENT OF LINEARIZED SOURCE TERM APPEARING IN THE
C       DENOMINATOR OF THE ENTHALPY OR ENTROPY EQUATION
C*************************************************************************
      INCLUDE 'comblk.for'
C*************************************************************************
C*************************************************************************
C   ADD CODE HERE
      RETURN
      END
C*************************************************************************
      SUBROUTINE SORCEHXQ(IPN,TERMD,K)
C   PURPOSE: ADD HEAT SOURCES
C   IPN - GFSSP INDEX NUMBER FOR NODE
C   TERMD - COMPONENT OF LINEARIZED SOURCE TERM APPEARING IN THE
C       DENOMINATOR OF THE ENTHALPY OR ENTROPY EQUATION
C*************************************************************************
      INCLUDE 'comblk.for'
C*************************************************************************
C*************************************************************************
C   ADD CODE HERE
      RETURN
      END
C*************************************************************************
      SUBROUTINE SORCEC
C   PURPOSE: ADD CONCENTRATION SOURCES
C*************************************************************************
      INCLUDE 'comblk.for'
C*************************************************************************
C   ADD CODE HERE
      RETURN
      END
C*************************************************************************
      SUBROUTINE SORCETS(IPSN,TERMD)
C   PURPOSE: ADD SOURCE TERM IN SOLID TEMPERATURE EQUATION
C*************************************************************************
      INCLUDE 'comblk.for'
C*************************************************************************
C   ADD CODE HERE
      RETURN
      END
```

```
C************************************************************************
      SUBROUTINE KFUSER(I,RHOU,EMUU,XVU,RHOUL,EMUUL,AKNEW)
C    PURPOSE: ADD A NEW RESISTANCE OPTION
C************************************************************************
      INCLUDE 'comblk.for'
C************************************************************************
C    ADD CODE HERE

      RETURN
      END
C************************************************************************
      SUBROUTINE KFADJUST(I,RHOU,EMUU,RHOUL,EMUUL,RHOUV,EMUUV,ISATU,
     &           AKNEW)
C    PURPOSE: ADD A NEW RESISTANCE OPTION
C************************************************************************
      INCLUDE 'comblk.for'
C************************************************************************
C    ADD CODE HERE
      RETURN
      END
C************************************************************************
      SUBROUTINE PRPUSER
C    PURPOSE: ADD NEW FLUID PROPERTY
C************************************************************************
      INCLUDE 'comblk.for'
C************************************************************************
C    ADD CODE HERE
      RETURN
      END


C************************************************************************
      SUBROUTINE TSTEP
C    PURPOSE: MODIFY TIME STEP
C************************************************************************
      INCLUDE 'comblk.for'
C************************************************************************
C    ADD CODE HERE
      RETURN
      END


C************************************************************************
      SUBROUTINE BNDUSER
C    PURPOSE: MODIFY BOUNDARY CONDITIONS
C************************************************************************
      INCLUDE 'comblk.for'
C************************************************************************
C    ADD CODE HERE
      RETURN
      END
C************************************************************************
      SUBROUTINE PRNUSER
C    PURPOSE: ADD NEW OUTPUT
C************************************************************************
      INCLUDE 'comblk.for'
C************************************************************************
C    ADD CODE HERE
C    write(nprnt,*) '*********** MIXTURE OPTION ************'
C       write (nprnt,*) 'IFRMIX = ', IFRMIX
C       write(nprnt,*) '*********** MIXTURE OPTION ************'
      RETURN
      END
C************************************************************************
      SUBROUTINE USRSET(FILEIN,TITLE,HISTORY,FNODE,FBRANCH,PCURVE,
     &       HISTGEO,HISTQ,HISTVLV,OVALV,CVALV,ANALYST,FILEOUT)
C    PURPOSE: USER SETS UP THE MAJORITY OF THE MODEL
C************************************************************************
      INCLUDE 'comblk.for'
```

304

```
C***********************************************************************
      CHARACTER*256, FILEIN,FILEOUT,ANALYST
      CHARACTER*80, TITLE
      CHARACTER*20, HISTQ(100),PCURVE(10),HISTGEO,HISTROT
    CHARACTER*256, HISTORY(100)
    CHARACTER*20, HISTVLV(10),OVALV(10),CVALV(10)
    CHARACTER*20, FNODE,FBRANCH
C   ADD CODE HERE
C
C THIS IS THE DEFAULT CODE FOR THIS BLOCK, COMMENT THIS OUT WHEN
C CREATING A MODEL WITHIN THIS SUBROUTINE
C
    WRITE(*,*) ' '
    WRITE(*,*) ' USER ROUTINE USRSET DOES NOT HAVE A MODEL DEVELOPED'
    WRITE(*,*) ' '
    WRITE(*,*) ' OPEN THE User Subroutine FILE AND MODIFY SUBROUTINE'
    WRITE(*,*) ' USRSET TO DEVELOP MODEL OR CHANGE LOGICAL VARIABLE'
    WRITE(*,*) ' USETUP TO FALSE AND DEVELOP MODEL IN INPUT FILE'
    WRITE(*,*) ' '
C    STOP
C
C END OF DEFAULT CODE
C
    RETURN
    END
C***********************************************************************
    SUBROUTINE USRHCF(NUMBER,HCF)
C   PURPOSE: PROVIDE HEAT TRANSFER COEFFICIENT
C***********************************************************************
    INCLUDE 'comblk.for'
C***********************************************************************
C   ADD CODE HERE
    RETURN
    END
C***********************************************************************
    SUBROUTINE USRADJUST
C   PURPOSE: ADJUST BOUNDARY CONDITION OR GEOMETRY FOR Steady state MODEL
C***********************************************************************
    INCLUDE 'comblk.for'
C***********************************************************************
C   ADD CODE HERE
    RETURN
    END
C***********************************************************************
    SUBROUTINE PRPADJUST
C   PURPOSE: ADJUST THERMODYNAMIC OR THERMOPHYSICAL PROPERTY
C***********************************************************************
    INCLUDE 'comblk.for'
C***********************************************************************
C   ADD CODE HERE
    RETURN
    END
C***********************************************************************
    SUBROUTINE TADJUST
C   PURPOSE: ADJUST TEMPERATURE IF NECESSARY
C***********************************************************************
    INCLUDE 'comblk.for'
C***********************************************************************
C   ADD CODE HERE
    RETURN
    END


C***********************************************************************
    SUBROUTINE PADJUST
C   PURPOSE: ADJUST PRESSURE IF NECESSARY
C***********************************************************************
    INCLUDE 'comblk.for'
C***********************************************************************
```

```
      RETURN
      END
C***********************************************************************
      SUBROUTINE FLADJUST
C    PURPOSE: ADJUST Flow rate IF NECESSARY
C***********************************************************************
      INCLUDE 'comblk.for'
C***********************************************************************
C    ADD CODE HERE
      RETURN
      END
C***********************************************************************
      SUBROUTINE HADJUST
C    PURPOSE: ADJUST ENTHALPY IF NECESSARY
C***********************************************************************
      INCLUDE 'comblk.for'
C***********************************************************************
C    ADD CODE HERE

      RETURN
      END
C***********************************************************************
      SUBROUTINE USRMDG
C    PURPOSE: ADJUST INPUT PARAMETERS FOR MULTI-D FLOW, IF NECESSARY
C***********************************************************************
      INCLUDE 'comblk.for'
C***********************************************************************
C    ADD CODE HERE
      RETURN
      END
C***********************************************************************
C***********************************************************************
C                                    *
C         ***** END OF User SubroutineS *****         *
C                                    *
C***********************************************************************
```

APPENDIX F—LIST OF PUBLICATIONS WHERE GFSSP HAS BEEN USED

	Title	Author(s)	Conference/Journal
1	A General Fluid System Simulation Program to Model Secondary Flows in Turbomachinery	Alok Majumdar Katherine Van Hooser	31st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, San Diego, CA, July 10–12, 1995, AIAA 95-2969
2	Mathematical Modeling of Free Convective Flows for Evaluating Propellant Conditioning Concepts	Alok Majumdar John Bailey Kimberly Holt Susan Turner	32nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Lake Buena Vista, FL, July 1–3, 1996, AIAA 96-3117
3	A Generalized Fluid System Simulation Program to Model Flow Distribution in Fluid Networks	Alok Majumdar John Bailey Biplab Sarkar	33rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Seattle, WA, July 6–9, 1997, AIAA 97-3225
4	Numerical Prediction of Pressure Distribution Along the Front and Back Face of a Rotating Disc With and Without Blades	Paul Schallhorn Alok Majumdar	33rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Seattle, WA, July 6–9, 1997, AIAA 97-3098
5	Flow Network Analyses of Cryogenic Hydrogen Propellant Storage and Feed Systems	Douglas Richards Daniel Vonderwell	33rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Seattle, WA, July 6–9, 1997, AIAA 97-3223
6	A Generalized Fluid System Simulation Program to Model Flow Distribution in Fluid Networks	Alok Majumdar John Bailey Paul Schallhorn Todd Steadman	34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Cleveland, OH, July 13–15, 1998, AIAA 98-3682
7	Flow Simulation in Secondary Flow Passages of a Rocket Engine Turbopump	Paul Schallhorn Alok Majumdar Katherine Van Hooser Matthew Marsh	34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Cleveland, OH, July 13–15, 1998, AIAA 98-3684
8	A Novel Approach for Modeling Long Bearing Squeeze Film Damper Performance	Paul Schallhorn David Elrod David Goggin Alok Majumdar	34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Cleveland, OH, July 13–15, 1998, AIAA 98-3684

	Title	Author(s)	Conference/Journal
9	Unstructured Finite Volume Computational Thermofluid Dynamics Method for Multi-Disciplinary Analysis and Design Optimization	Alok Majumdar Paul Schallhorn	7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St. Louis, MO, September 2–4, 1998, AIAA 98-4810
10	***Numerical Modeling of Pressurization of a Propellant Tank***	***Alok Majumdar*** ***Todd Steadman***	***Journal of Propulsion and Power, Vol. 17, No. 2, pp. 385–390, 2001; 37th AIAA Aerospace Sciences Meeting Conference and Exhibit, Reno, NV, January 11–14, 1999, AIAA 99-0879***
11	A Second Law Based Unstructured Finite Volume Procedure for Generalized Flow Simulation	Alok Majumdar	37th AIAA Aerospace Sciences Meeting Conference and Exhibit, Reno, NV, January 11–14, 1999, AIAA 99-0934
12	Numerical Prediction of Transient Axial Thrust and Internal Flows in a Rocket Engine Turbopump	Katherine Van Hooser John Bailey Alok Majumdar	35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Los Angeles, CA, June 21, 1999, AIAA 99-2189
13	Numerical Modeling of Helium Pressurization System of Propulsion Test Article (PTA)	Todd Steadman Alok Majumdar Kimberly Holt	Thermal & Fluids Analysis Workshop, September 13–17, 1999, Huntsville, AL
14	A Steady State and Quasi-Steady Interface Between the Generalized Fluid System Simulation Program and the SINDA/G Thermal Analysis Program	Paul Schallhorn Alok Majumdar Bruce Tiller	Thermal & Fluids Analysis Workshop, Huntsville, AL, September 13–17, 1999
15	Interfacing a General Purpose Fluid Network Flow Program with the Sinda/G Thermal Analysis Program	Paul Schallhorn Dan Popok	SAE Paper No. 1999-01-2162
16	An Unsteady Long Bearing Squeeze Film Damper Model – Part I: Circular Centered Orbits	Paul Schallhorn David Elrod David Goggin Alok Majumdar	38th AIAA Aerospace Sciences Meeting Conference and Exhibit, Reno, NV, January 11–14, 1999, AIAA 2000-0352

	Title	Author(s)	Conference/Journal
17	An Unsteady Long Bearing Squeeze Film Damper Model – Part II: Statically Eccentric Operation	Paul Schallhorn David Elrod David Goggin Alok Majumdar	38th AIAA Aerospace Sciences Meeting Conference and Exhibit, Reno, NV, January 11–14, 1999, AIAA 2000-0353
18	*A Fluid Circuit Model for Long Bearing Squeeze Film Damper Rotordynamics*	*Paul Schallhorn David Elrod David Goggin Alok Majumdar*	*AIAA Journal of Propulsion and Power, Vol. 16, No. 5, pp. 777–780, Sept–Oct 2000*
19	Unsteady Analysis of the Fluid Film Forces in a Long Bearing Squeeze Film Damper	Paul Schallhorn	Ph.D. Dissertation, The University of Alabama in Huntsville, 1998
20	Numerical Modeling and Test Data Comparison of Propulsion Test Article Helium Pressurization System	Kimberly Holt Alok Majumdar Todd Steadman Ali Hedayat	36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Huntsville, AL, July 16–19, 2000, AIAA 2000-3719
21	Numerical Modeling of Drying Residual RP-1 in Rocket Engines	Alok Majumdar Robert Polsgrove Bruce Tiller	Thermal & Fluids Analysis Workshop, Cleveland, OH, August 21–25, 2000
22	Incorporation of Condensation Heat transfer Model into a Flow Network Code	Miranda Anthony Alok Majumdar	Thermal & Fluids Analysis Workshop, Huntsville, AL, September 10–14, 2001
23	Discharge Characteristics of the International Space Station (Ise) Portable Fire Extinguisher (Pfe) and the effect on Closed Volumes	Charles E Martin Paul Schallhorn Paul Wieland	SAE Paper No. 2001-01-2316
24	*Modeling of Chill Down in Cryogenic Transfer Lines*	*M. Cross A.K. Majumdar J. C. Bennett Jr. R. B. Malla*	*Journal of Spacecraft and Rockets, Vol. 39, No. 2, pp. 284–289, 2002*
25	Design optimization of a nuclear reactor for earth-to-orbit applications	T. Moton D. B Landrum, P. Schallhorn	40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 14–17, 2002, AIAA-2002-363
26	Numerical modeling of cavitating venturi—a flow control element of propulsion system	Alok Majumdar	Thermal & Fluids Analysis Workshop, Houston, TX, August 12–16, 2002

	Title	Author(s)	Conference/Journal
27	Numerical Modeling of Fluid Transient by a Finite Volume Procedure for Rocket Propulsion Systems	Alok Majumdar Robin Flachbart	Proceedings of ASME FEDSM'03, 4th ASME/JSME Joint Fluids Engineering Conference, Paper No. FEDSM2003-45275, Honolulu, HI, July 6–10, 2003
28	Numerical Modeling of Thermofluid Transients During Chilldown of Cryogenic Transfer Lines	Alok Majumdar Todd Steadman	33rd International Conference on Environmental Systems (ICES), Paper No. 2003-01-2662, Vancouver, Canada, July 6–10, 2003
29	Numerical Modeling of Unsteady Thermofluid Dynamics in Cryogenic Systems	Alok Majumdar	Thermal & Fluids Analysis Workshop, Hampton, VA, August 18–22, 2003
30	A Novel Approach for Modeling Chemical Reaction in Generalized Fluid System Simulation Program	Mehmet Sozen Alok Majumdar	39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Huntsville, AL, July 20–23, 2003, AIAA 2003-4467
31	Forward Looking Pressure Regulator Algorithm for Improved Modeling Performance Within the Generalized Fluid System Simulation Program http://pdf.aiaa.org/getfile.cfm?urlX=5%3A7I%276D%26X%5B%22%3F%28S0KUWT%5B%5EP%2B%3B%3C4JT%22%0A&urla=%%2ARH%%23PH%0A&urlb=%21%2A%0A&urlc=%21%2A0%0A&urld=%28%2A%22T%22%21P%22ATA4%0A&urle=%28%2A%22H%%21%40%2E%40T14%0A	Paul Schallhorn Neal Hass	40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Fort Lauderdale, FL, July 11–14, 2004, AIAA-2004-3667
32	Numerical Modeling of Conjugate Heat Transfer in Fluid Network	Alok Majumdar	Thermal & Fluids Analysis Workshop, Jet Propulsion Laboratory, Pasadena, CA, August 30–September 3, 2004,
33	Numerical Modeling of Flow Distribution in Microfluidics Systems	Alok Majumdar Helen Cole C. P. Chen	Proceedings of Forum on Microfluidics Devices and Systems, ASME Fluids Engineering Conference, Paper No. FEDSM 2005-77378, Houston, TX, June 19–23, 2005

	Title	Author(s)	Conference/Journal
34	Development and Implementation of NonNewtonian Rheology into the Generalized Fluid System Simulation Program (GFSSP)	Roberto Di Salvo Stelu Deaconn Alok Majumdar	42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Sacramento, CA, July 9–12, 2006,
35	Microfluidic System Simulation including the Electro-Viscous Effect	Eileen Rojas C. P. Chen Alok Majumdar	Integration and Commercialization of Macro and Nano Systems, ASME International Conference, Paper No. MNC2007-21295, Sanya, Hainan China, Jan. 10–13, 2007
36	***Numerical Modeling of Propellant Boil-off in a Cryogenic Storage Tank***	***A K Majumdar*** ***T E Steadman*** ***J L Maroney*** ***J P Sass*** ***J E Fesmire***	***Advances in Cryogenic Engineering, Volume 53 A, American Institute of Physics, pp. 1507–1516, 2008***
37	Computational Model of the Chilldown and Propellant Loading of the Space Shuttle External Tank	André C. LeClair Alok K. Majumdar	46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Nashville, TN, July 25–28, 2010
38	***Computational Modeling of Fluid and Thermal Transients for Rocket Propulsion Systems by Fast Nonlinear Network Solver***	*Alok Majumdar* **S.S. Ravindran**	***International Journal of Numerical Methods for Heat & Fluid Flow, Vol. 20, No. 6, pp. 617–637, 2010***
39	*Numerical Modeling of Conjugate Heat Transfer in Fluid Network*	***Alok Majumdar*** ***S.S. Ravindran***	***AIAA Journal of Propulsion and Power, Vol. 27(3), pp. 620–630, 2011***
40	Network Flow Simulation of Fluid Transients in Rocket Propulsion Systems	Alak Bandyopadhyay Brian Hamill Narayanan Ramachandran Alok Majumdar	JANNAF 8th MSS, 6th LPS. 5th SPS Joint Meeting, Huntsville, AL, December 5–8, 2011
41	Generalized Fluid System Simulation Program (GFSSP) Version 6 – General Purpose Thermofluid Network Analysis Software	Alok Majumdar Andre Leclair Ric Moore Paul Schallhorn	Thermal Fluids Analysis Workshop (TFAWS), Newport News, VA , August 15–19, 2011
42	Implementation of Finite Volume based Navier Stokes Algorithm within General Purpose Flow Network Code	Paul Schallhorn Alok Majumdar	50th AIAA Aerospace Sciences Meeting, Nashville, TN, January 9–12, 2012

REFERENCES

1. Majumdar; A.K.; Bailey, J.W.; Schallhorn, P.A.; and Steadman, T.E.: Generalized Fluid System Simulation Program, U.S. Patent No. 6,748,349, June 8, 2004.

2. Streeter, V.L.: *Fluid Mechanics*, 3rd ed., McGraw-Hill, New York, 1962.

3. Stoecker, W.F.: *Design of Thermal Systems*, 3rd ed., McGraw-Hill, New York, 1989.

4. Owen, J.W.: "Thermal Analysis Workbook," NASA TM–103568, NASA Marshall Space Flight Center, AL, January 1992.

5. SINDA/G Thermal Analyzer, Network Analysis Inc., <http://www.sinda.com>, accessed January 2001.

6. SINDA/FLUINT, Heat Transfer and Fluid Flow Design and Analysis Software, C&R Technologies, <http://www.crtech.com/sinda.html>, accessed January 2001.

7. Seymore, D.C.: ROCETS Users Manual, Internal Report, NASA Marshall Space Flight Center, AL, October 1999.

8. Broyden, C.G.: "A Class of Methods for Solving Nonlinear Simultaneous Equations," *Mathematics of Computation*, Vol. 19, No. 92, pp. 577–593, October 1965.

9. Patankar, S.V.: *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corp., Washington, DC, 197 pp., 1980.

10. Hendricks, R.C.; Baron, A.K.; and Peller, I.C.: "GASP — A Computer Code for Calculating the Thermodynamic and Transport Properties for Ten Fluids: Parahydrogen, Helium, Neon, Methane, Nitrogen, Carbon Monoxide, Oxygen, Fluorine, Argon, and Carbon Dioxide," NASA TN D-7808, NASA Lewis Research Center, Cleveland, OH, February 1975.

11. Hendricks, R.C.; Peller, I.C.; and Baron, A.K., "WASP - A Flexible Fortran IV Computer Code for Calculating Water and Steam Properties," NASA TN D-7391, NASA Lewis Research Center, Cleveland, OH, November 1973.

12. *User's Guide to GASPAK*, Version 3.20, Cryodata, Inc., November 1994.

13. Winplot Version 4.54 User's Manual, NASA Marshall Space Flight Center, AL, July 16, 2008.

14. Holman, J.P.: *Heat Transfer*, 8th ed., McGraw Hill, New York, 1997.

15. Miropolskii, Z.L.: "Heat Transfer in Film Boiling of a Steam-Water Mixture in Steam Generating Tubes," *Teploenergetika*, Vol. 10, No. 5, pp. 49–52, 1963 (in Russian; translation Atomic Energy Commission, AEC-Tr-6252, 1964).

16. Colebrook, C.F.: "Turbulent Flow in Pipes, with Particular Reference to the Transition Between the Smooth and Rough Pipe Laws," *J. Inst. Civil Eng.,* Vol. 11, pp. 133–156, 1938–1939.

17. White, F.W.: *Viscous Fluid Flow*, McGraw Hill, 1991.

18. Hooper, W.B.: "Calculate Head Loss Caused by Change in Pipe Size," *Chem. Eng.,* Vol. 95, No. 16, pp. 89–92, November 7, 1988.

19. Yamada, Y.: "Resistance to a Flow through an Annulus with an Inner Rotating Cylinder," *Bull. Jpn. Soc. Mech. Eng.*, Vol. 5, pp. 302–310, 1962.

20. Ito, H.; and Nanbu, K.: "Flow in Rotating Straight Pipes of Circular Cross Section," *Trans. ASME*, Vol. 93, Series D, No. 3, pp. 383–394, 1971.

21. Howell, G.W.; and Weather, T.M.: "Aerospace Fluid Component Designers Handbook," Vol. 1, Rev. D, RPL-TDR-64-25, TRW Systems Group, February 1970.

22. Hooper, W.B.: "The Two-K Method Predicts Head Losses in Pipe Fittings," *Chem. Engr.*, Vol. 24, pp. 97–100, August 24, 1981.

23. Lebar, J.F.; and Cady, E.C.: "Viscojet Testing at NASA Lewis Research Center CCL-7 Test Facility," Report No. A3-Y953-JFL-9G136, July 1990.

24. Kays, W.M.; and London, A.L.: *Compact Heat Exchangers*, 3rd ed., McGraw-Hill, New York, 1984.

25. Egli, A.: "The Leakage of Steam through Labyrinth Seals," *ASME Trans*, Vol. 27, pp. 115–122, April 1935.

26. Patankar, S.V.; and Spalding, D.B.: "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three Dimensional Parabolic Flows," *Int. J. Heat Mass Transfer*, Vol. 15, pp. 1787–1806, 1972.

27. Bandyopadhyay, A.; Hamill, B.; Ramachandran, N.; and Majumdar, A.: "Network Flow Simulation of Fluid Transients in Rocket Propulsion Systems," JANNAF 8th Modeling and Simulation Subcommittee Meeting, 6th LPS, 5th SPS Joint Meeting, Huntsville, Alabama, December 5–8, 2011.

28. Majumdar, A.K.; and Van Hooser, K.: "A General Fluid System Simulation Program to Model Secondary Flows in Turbomachinery," Paper No. AIAA 95-2969, 31st AIAA/ASME/ SAE/ASEE Joint Propulsion Conference and Exhibit, San Diego, CA, July 10–12, 1995.

29. Schallhorn, P.; Majumdar, A.; Van Hooser, K.; and Marsh, M.: "Flow Simulation in Secondary Flow Passages of a Rocket Engine Turbopump," Paper No. AIAA 98-3684, 34th AIAA/ASME/SAE/ASEE, Joint Propulsion Conference and Exhibit, Cleveland, OH, July 13–15, 1998.

30. Van Hooser, K.; Majumdar, A.; and Bailey, J.: "Numerical Prediction of Transient Axial Thrust and Internal Flows in a Rocket Engine Turbopump," Paper No. AIAA 99-2189, 35th AIAA/ASME/SAE/ASEE, Joint Propulsion Conference and Exhibit, Los Angeles, CA, June 21, 1999.

31. Majumdar, A.K.; and Steadman, T.: "Numerical Modeling of Pressurization of a Propellant Tank," Paper No. AIAA 99-0879, 37th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 11–14, 1999.

32. Steadman, T.; Majumdar, A.K.; and Holt, K.: "Numerical Modeling of Helium Pressurization System of Propulsion Test Article (PTA)," 10th Thermal Fluid Analysis Workshop, Huntsville, AL, September 13–17, 1999.

33. Cross, M.; Majumdar, A.; Bennett, J.; and Malla, R.: "Modeling of Chill Down in Cryogenic Transfer Lines," *J. Spacecraft Rockets*, Vol. 39, No. 2, pp. 284–289, 2002.

34. Majumdar, A.; and Ravindran, S.S.: "Numerical Modeling of Conjugate Heat Transfer in Fluid Network," *J. Prop. Power*, Vol. 27, No. 3, pp. 620–630, 2011.

35. LeClair, A.C.; and Majumdar, A.K.: "Computational Model of the Chilldown and Propellant Loading of the Space Shuttle External Tank,' Paper No. AIAA 2010-6561, 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Nashville, TN, July 25–28, 2010.

36. Majumdar, A.K.; Steadman, T.; Maroney, J.L.; et al.: "Numerical Modeling of Propellant Boil-off in a Cryogenic Storage Tank," *Adv. in Cryogenic Eng.*, Vol. 53A, pp. 1507–1516, 2008.

37. Majumdar, A.K.; and Flachbart, R.H.: "Numerical Modeling of Fluid Transients by a Finite Volume Procedure for Rocket Propulsion Systems," Paper No. FEDSM2003-45275, *Proc. ASME FEDSM'03*, 4th ASME/JSME Joint Fluids Engineering Conference, Honolulu, HI, July 6–10, 2003.

38. Schallhorn, P.; and Hass, N.: "Forward Looking Pressure Regulator Algorithm for Improved Modeling Performance Within the Generalized Fluid System Simulation Program," AIAA 2004-3667, 40th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Fort Lauderdale, FL, July 11–14, 2004.

39. Majumdar, A.; Cole, H.; and Chen, C.P.: "Numerical Modeling of Flow Distribution in Microfluidics Systems," Paper No. FEDSM 2005-77378, Proceedings of Forum on Microfluidics Devices and Systems, ASME Fluids Engineering Conference, Houston, TX, June 19–23, 2005.

40. Hodge, B.K.: *Analysis and Design of Energy Systems*, 2nd ed., Prentice Hall, 1990.

41. Schallhorn, P.A.; and Majumdar, A.K.: "Numerical Prediction of Pressure Distribution Along the Front and Back Face of a Rotating Disc With and Without Blades," Paper No. AIAA 97-3098, Presented at the 33rd Joint Propulsion Conference, Seattle, WA, July 6–9, 1997.

42. Schallhorn, P.A.; Elrod, D.A.; Goggin, D.G.; and Majumdar, A.K.: "A Novel Approach for Modeling Long Bearing Squeeze Film Damper Performance," Paper No. AIAA 98-3683, Presented at the 34th Joint Propulsion Conference, Cleveland, OH, July 13–15, 1998.

43. Moody, F.J.: *Introduction to Unsteady Thermofluid Mechanics*, Wiley-Interscience, New York, 667 pp., 1990.

44. Reid, R.C.; Prausnitz, J.M.; and Poling, B.E.: *The Properties of Gases and Liquids*, 4th ed., McGraw Hill, New York, 1987.

45. Epstein, M.; and Anderson, R.E.: "An Equation for the Prediction of Cryogenic Pressurant Requirements for Axisymmetric Propellant Tanks," *Adv. in Cryogenic Eng.*, Vol. 13, pp. 207–214, 1968.

46. Van Dresar, N.T.: "Prediction of Pressurant Mass Requirements for Axisymmetric Liquid Hydrogen Tanks," *J. Propul. Power*, Vol. 13, No. 6, pp. 796–799, 1997.

47. McRight, P.: "Pressurization Analysis - A Visual Basic Computer Program to Calculate Mass Flowrate of Compressible Sonic Flow" unpublished work at NASA Marshall Space Flight Center, AL, November 14, 1997.

48. Incropera, F.P.; Dewitt, D.P.; Bergman, T.L.; and Lavine, A.S.: *Introduction to Heat Transfer,* 5th Edition, John Wiley, 2007.

49. Majumdar, A.; and Ravindran, S.S.: "Computational Modeling of Fluid and Thermal Transients for Rocket Propulsion Systems by Fast Nonlinear Network Solver," *Int. J. Numer. Method H.*, Vol. 20, No. 6, pp. 617–637, 2010.

50. Burggraf, O.R.: "Analytical and Numerical Studies of the Structure of Steady Separated Flows," *J. Fluid Mech.*, Vol. 24, Part 1, pp. 113–151, 1966.

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, 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 Department of Defense, Washington Headquarters Services, Directorate for Information Operation and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 01–10–2013	2. REPORT TYPE Technical Memorandum	3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE Generalized Fluid System Simulation Program, Version 6.0	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) A.K. Majumdar, A.C. LeClair, R. Moore, and P.A. Schallhorn*	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) George C. Marshall Space Flight Center Huntsville, AL 35812	8. PERFORMING ORGANIZATION REPORT NUMBER M–1368

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546–0001	10. SPONSORING/MONITOR'S ACRONYM(S) NASA
	11. SPONSORING/MONITORING REPORT NUMBER NASA/TM—2013–217492

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 34
Availability: NASA CASI (443–757–5802)

13. SUPPLEMENTARY NOTES
*Kennedy Space Center, FL
Prepared by the Propulsion Systems Department, Engineering Directorate
Supplemental information is available on CD through CASI

14. ABSTRACT
The Generalized Fluid System Simulation Program (GFSSP) is a finite-volume based general-purpose computer program for analyzing steady state and time-dependant flow rates, pressures, temperatures, and concentrations in a complex flow network. The program is capable of modeling real fluids with phase changes, compressibility, mixture thermodynamics, conjugate heat transfer between solid and fluid, fluid transients, pumps, compressors and external body forces such as gravity and centrifugal. The thermo-fluid system to be analyzed is discretized into nodes, branches, and conductors. The scalar properties such as pressure, temperature, and concentrations are calculated at nodes. Mass flow rates and heat transfer rates are computed in branches and conductors. The graphical user interface allows users to build their models using the 'point, drag, and click' method; the users can also run their models and post-process the results in the same environment. The integrated fluid library supplies thermodynamic and thermo-physical properties of 36 fluids, and 24 different resistance/source options are provided for modeling momentum sources or sinks in the branches. This Technical Memorandum illustrates the application and verification of the code through 25 demonstrated example problems.

15. SUBJECT TERMS
network flow, finite volume method, thermofluid system, two-phase, conjugate heat transfer

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON STI Help Desk at email: help@sti.nasa.gov
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	747	
U	U	U			19b. TELEPHONE NUMBER *(Include area code)* STI Help Desk at: 443–757–5802

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

National Aeronautics and
Space Administration
IS20
George C. Marshall Space Flight Center
Huntsville, Alabama 35812