Introduction to Sinaps*Plus*®

Sinaps*Plus*® (pronounced “synapse plus”) is a complete graphical user interface and model debugging environment for the SINDA/FLUINT thermal/fluid simulation package.

SINDA/FLUINT is a comprehensive software package used by over 400 sites in the aerospace, energy, electronic, automotive, aircraft, HVAC, and petrochemical industries for design and simulation of heat transfer and fluid flow problems. It is the NASA-standard analyzer for thermal control systems. *The reader is assumed to be familiar with SINDA/FLUINT*, which is described in separately available documents. This document concentrates on the use of Sinaps*Plus* for thermal modeling. A parallel document concentrating instead on fluid modeling using Sinaps*Plus* is also available.

As described in those documents, SINDA/FLUINT is traditionally a batch-style program. The analyst supplies ASCII (card image) input files, and receives results as ASCII tabulations. Sinaps*Plus* completely replaces this method of communicating models and results via modern graphical pre- and postprocessing.

What is Sinaps*Plus*®?

Sinaps*Plus* is a circuit design tool for thermal/fluid circuits (i.e., heat transfer and fluid flow problems represented as networks). It is a point-and-click, menu-driven interface to the heat transfer and fluid flow modeling tools available in the full SINDA/FLUINT analyzer.

Users describe their SINDA/FLUINT model by sketching their networks on the computer screen and by providing data in pop-up forms. Sinaps*Plus* then automatically creates the traditional SINDA/FLUINT input file, or directly launches a SINDA/FLUINT run. Once SINDA/FLUINT execution is complete, Sinaps*Plus* may be used to postprocess results files by applying color to the original network schematic, by preparing pop-up plots, etc. Figure 1 depicts a Sinaps*Plus* network sketch that was used to create and launch a SINDA/FLUINT simulation model, and was then postprocessed to reflect the results of that execution. (Color figures in this document may be misleading or unaesthetic if viewed or printed in black and white.) Sinaps*Plus* can also be used to produce generalized tools called *prebuilts* for use by others.

Organization: Images and Desktops

All Sinaps*Plus* data and diagrams are stored within *image files* (Figure 2). Individual users typically employ several such image files, one for each project. Each image file contains a set of SINDA/FLUINT *model desktops* that are all accessible like folders or subdirectories from the *top-level model browser.* One model desktop exists for each SINDA/FLUINT model. An example of a top-level model browser containing multiple models is presented in Figure 3.

The top-level *model browser* allows the user to browse the list of models contained within the image, to create new models, delete models, and to perform model-level input/output operations. For example, the user may read and write Sinaps*Plus* desktop ‘binaries’ from the top-level model browser. The user may enter a model desktop from the top level. Once in a model desktop, the user may either return to the top level, or jump directly to another model desktop.

Upon starting Sinaps*Plus* from and existing image file, the model desktop that first appears will correspond to the one that was in use when the image file was last saved. It will appear in the same layout as the last time the user commanded a *save everything*.

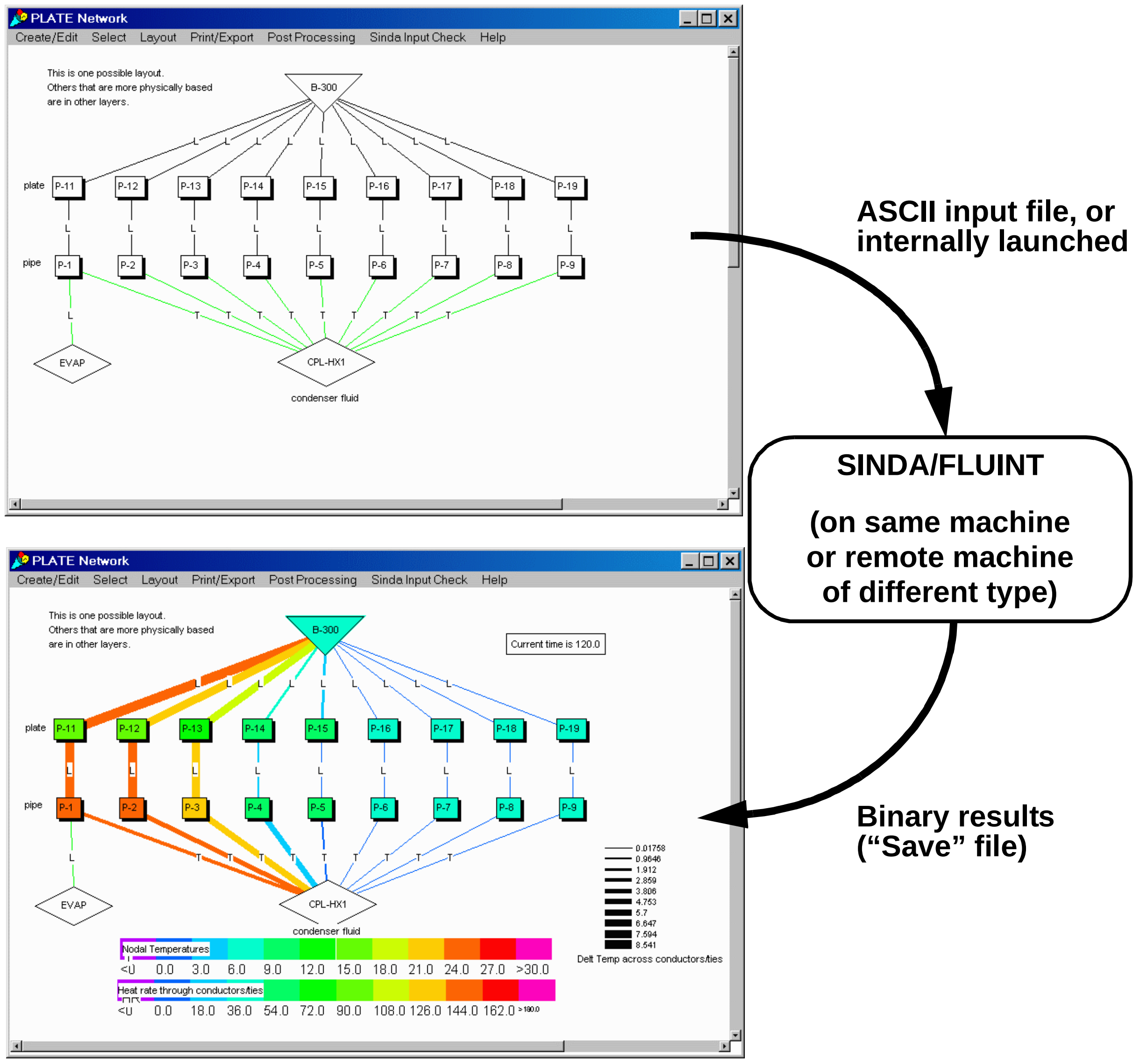


PLATE Network
Create/Edit Select Layout Print/Export Post Processing Sinda Input Check Help
This is one possible layout.
Others that are more physically based
are in other layers.
B-300
plate
P-11 P-12 P-13 P-14 P-15 P-16 P-17 P-18 P-19
pipe
P-1 P-2 P-3 P-4 P-5 P-6 P-7 P-8 P-9
EVAP
CPL-HX1
condenser fluid
ASCII input file, or
internally launched
SINDA/FLUINT
(on same machine
or remote machine
of different type)
Binary results
("Save" file)
Current time is 120.0
Nodal Temperatures
<U 0.0 3.0 6.0 9.0 12.0 15.0 18.0 21.0 24.0 27.0 >30.0
Heat rate through conductors/ties
<U 0.0 18.0 36.0 54.0 72.0 90.0 108.0 126.0 144.0 162.0 >180.0
0.01758
0.9646
1.912
2.859
3.806
4.753
5.7
6.647
7.594
8.541
Delt Temp across conductors/ties

Figure 1: Basic Data Flow

Each model desktop (e.g., Figure 4) contains all the information relating to a complete SINDA/FLUINT model: logic blocks, network diagrams, etc. When Sinaps*Plus* is entered, the user has access to all the models contained within that image, but may not access models contained within other image files. Within an image file, the user as the ability to include portions of one model in a second model, thus avoiding the need to recreate common networks. To access a model contained within another image, Sinaps*Plus* must be exited and restarted using the new image file name. Users may archive models independently of images, and may move models between images via input/output operations: a model or submodel may be *filed out* of one Sinaps*Plus* image, and then *filed into* another image via model desktop options. Such transfer of model desktops can be performed even if the two images reside on different types of host computers.

Windows and Buttons

Each Sinaps*Plus* desktop is composed of multiple windows. Some windows invoke program options, some search for and/or open new windows, and other windows store SINDA/FLUINT model information.

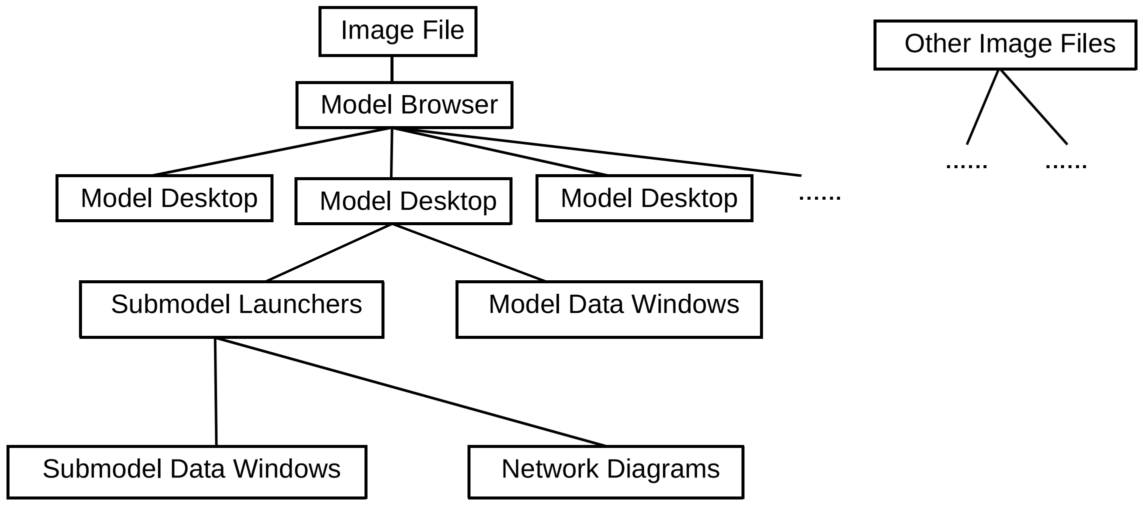
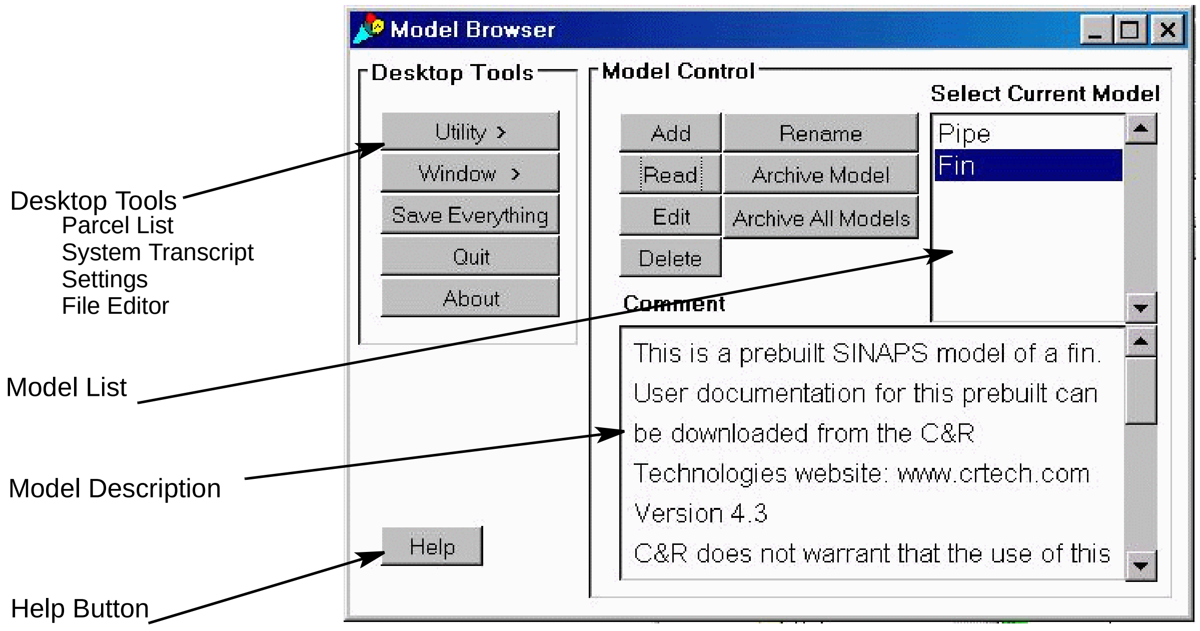


Figure 2: Image File Structure



This is a prebuilt SINAPS model of a fin.
User documentation for this prebuilt can
be downloaded from the C&R
Technologies website: www.crtech.com
Version 4.3
C&R does not warrant that the use of this

Figure 3: Sample Top-level Desktop (Model Browser)

Almost all program options are accessible via mouse button clicks, using the left-most mouse button on multiple-button mice, or the only button on single-button mice. (Unless otherwise noted, "mouse button" means "left-most mouse button" throughout this document.) Help windows are also available throughout Sinaps*Plus* to describe operations within each window.

Control Panels—Control panels are multi-function windows. Each control panel contains model or submodel browsers in addition to cascading menus which either open new windows, or initiate actions such as preprocessing, importing data, etc.

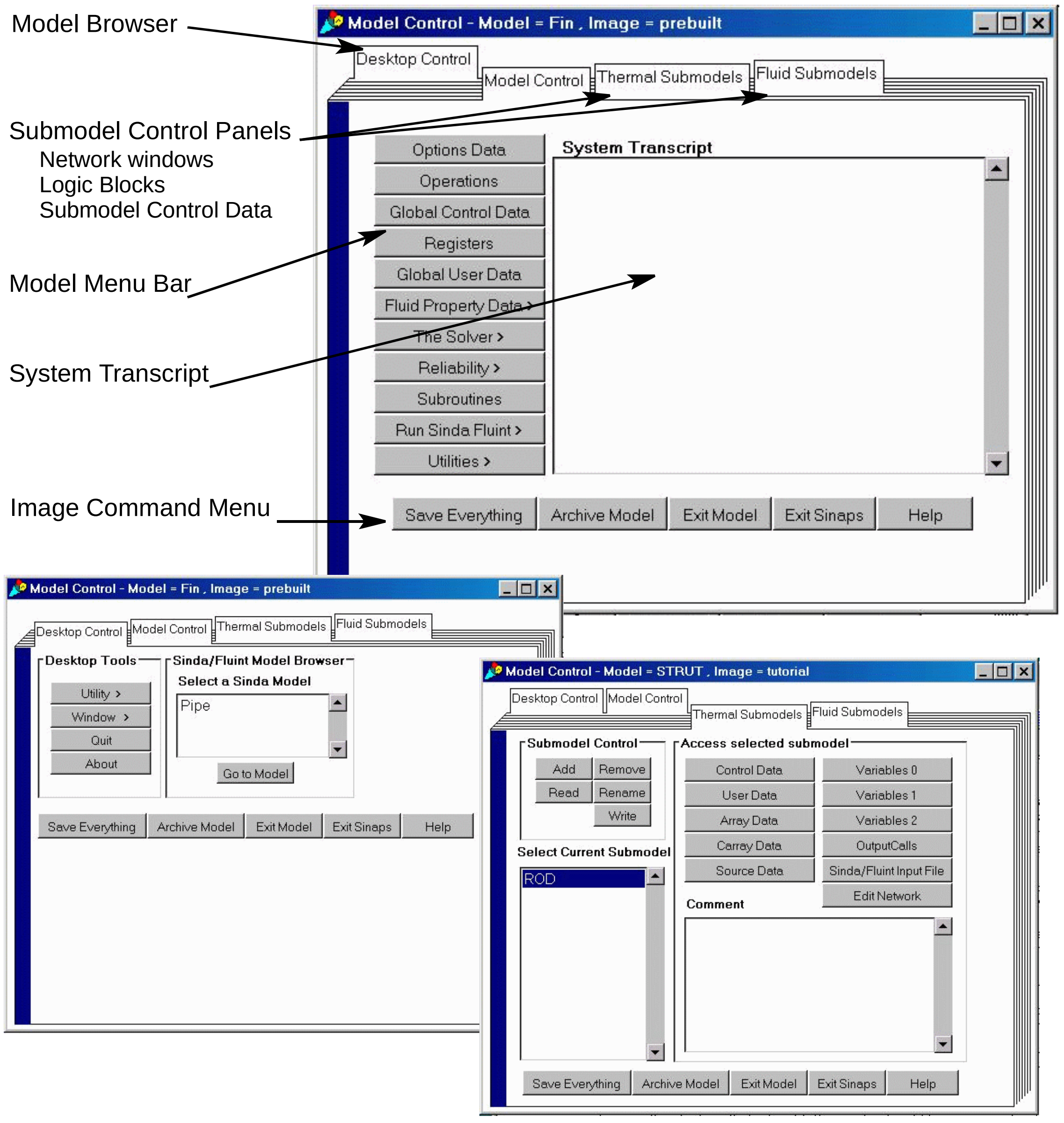


Figure 4: Sample Model-level Desktop (Control Panel)

Each image file contains a top-level *model browser* (Figure 3) which provides access to various image level options such as Save Everything (which writes all image data to the disk) and allows the user to browse the list of models contained within the image, to create new models, delete models, and to perform model-level input/output operations. The user also has access to *utilities* including the *parcel list* (used to install patches), an ASCII *file editor*, *system transcript*, and *settings* (display preferences).

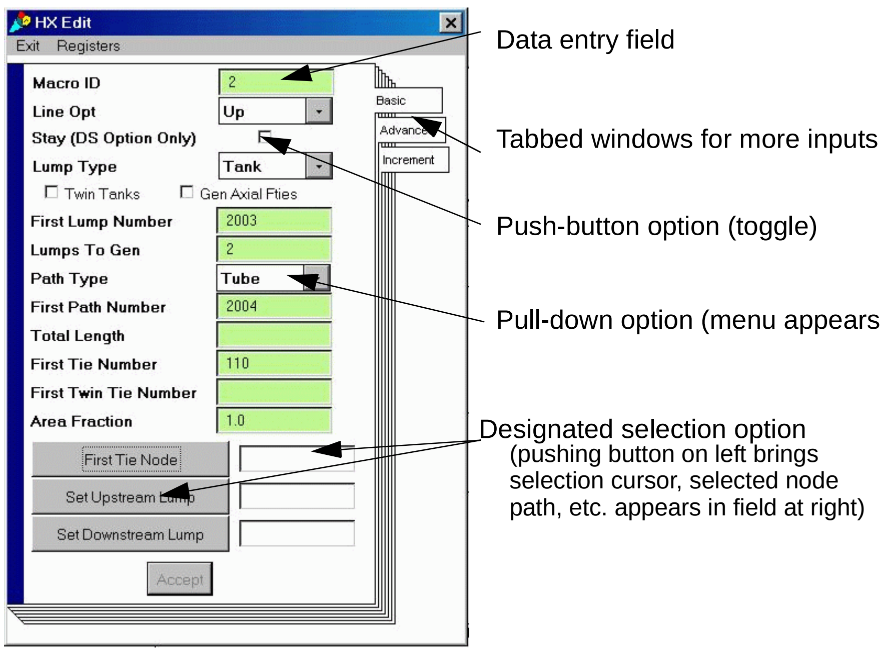
Each model has an associated *control panel* which contains multiple tabbed windows as shown in Figure 4. The model control panel gives the user access to a *desktop control* subpanel which provides functions similar to the top-level model browser; a *model control* subpanel providing access to model level logic blocks, registers, control data, fluid property data, advanced modeling options, model execution, and model level utilities; *thermal/fluid submodel* control panels providing the user with access to submodel networks, browsers, logic, control data, and utilities. Each tabbed window of the model control panel contains an image level command menu at the bottom of the window allowing the user to save the image to disk, archive/exit the model, exit and access help. When a model is exited, the user is returned to the top-level model browser window.

Other Windows—Most windows within a desktop are specialized to handle a different type of SINDA/FLUINT header block. Almost all such windows have menu bars along the tops through which pull-down menus may used to perform all functions within those windows.

Some windows, such as those used to edit Operations and other logic blocks, are windows in which the user simply enters text. These *text edit windows* function like most visual text editors with mouse-driven cut and paste operations, etc.

Other types of windows include the model diagram window (examples of which are presented in Figure 1), which have many features and options. These sketch pads, which will be described later, are the heart of Sinaps*Plus*.

Menus, Buttons, and Forms—As mentioned above, most Sinaps*Plus* options are accessible using the mouse button to navigate through pull-down menus. Many features are also accessible via double-clicks (i.e., pushing the left-most mouse button twice in rapid succession).


HX Edit
Exit
Registers
Basic
Advanced
Increment
First Tie Node
Set Upstream Lump
Set Downstream Lump
Accept
Data entry field
Tabbed windows for more inputs
Push-button option (toggle)
Pull-down option (menu appears
Designated selection option
(pushing button on left brings selection cursor, selected node path, etc. appears in field at right)

Macro ID	2
Line Opt	Up
Stay (DS Option Only)	☐
Lump Type	Tank
☐ Twin Tanks	☐ Gen Axial Fties
First Lump Number	2003
Lumps To Gen	2
Path Type	Tube
First Path Number	2004
Total Length	
First Tie Number	110
First Twin Tie Number	
Area Fraction	1.0

Figure 5: Sample Data Entry Form

Pop-up forms are also common. Such forms feature push-button toggles, fill-in fields (for numeric data and expressions), and tabbed window options. Examples of these features can be found in Figure 5.

Information in each window, form, and pop-up can be saved individually. Such "saves" store the changes in memory; *the Save Everything option must be used to store the changes to hard disk.*

For the Novice User—In addition to help buttons, Sinaps*Plus* employs several methods for preventing novice users from becoming overwhelmed by the many options available. First, menu trees have been carefully arranged such that the more common or easily used options are available in the top layers, while the more arcane options are "hidden" deeper within the tree. Also, input options have been color coded as follows:

White	Almost always a required input
Green	Novice (required options, or options that are easily or often employed)
Blue	Intermediate (options more difficult to use or more rarely employed)
Red	Advanced (options rarely used, or difficult to use correctly)

Input Development

Within a model desktop, the user may perform input, editing, data validation, and postprocessing operations in almost arbitrary order by simply moving between windows or opening new ones. This section describes some of the features available for the creation and manipulation of SINDA/FLUINT model information.

Network Diagrams—The heart of Sinaps*Plus* is the network diagrams (e.g., Figure 1). These diagrams enable a nongeometric code like SINDA/FLUINT to be used visually. The user creates arbitrarily arranged 2D schematics of their networks within such windows. These diagram windows functionally replace the NODE DATA, CONDUCTOR DATA, and FLOW DATA portions of a traditional SINDA/FLUINT input file.

One such window exists for each thermal submodel, fluid submodel, and FLUINT macro. These windows display a portion of an underlying sketchpad *sheet*, which may be arbitrarily large (subject only to the memory constraints of the host machine). The window may be used to pan the view of large sheets. Also, the entire sheet may be temporarily shrunk such that it can be viewed within the current window.

Each SINDA/FLUINT element (tank, arithmetic node, etc.) has its own distinctive icon, as depicted in Figure 6. These may be placed anywhere within the diagram sheet, and moved or aligned as needed. To select icons for operations (edit, cut, etc.), use the mouse button. A red square will appear confirming the selection. The pull-down menus can then be used to operate on the selected icons. Or, to edit a single icon, simply double-click it. To move icons, hold down the shift key and drag them to the desired location using the mouse button. Note that many pull-down menu items have key-board shortcuts, which appear on the menu (i.e. Alt-S).

Once they have been generated, these icons may be interconnected by lines representing conductors, paths, and ties. These lines may be straight, or they may be shaped as poly-line segments if needed. Lines are labeled by their type (e.g., “R” for radiation conductor, “T” for tube, etc.) and optionally by their numeric identifier.*

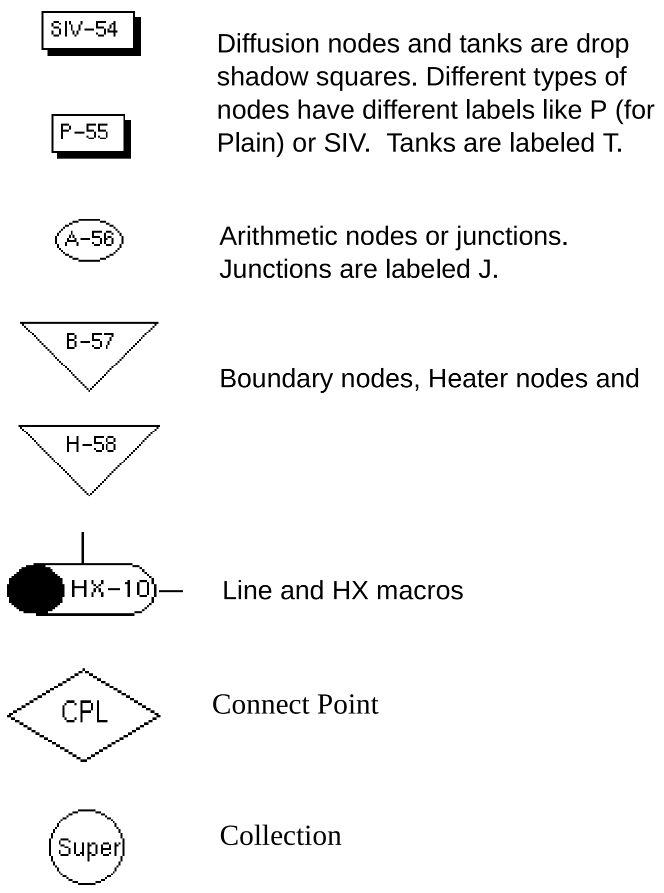

SIV-54
P-55
Diffusion nodes and tanks are drop shadow squares. Different types of nodes have different labels like P (for Plain) or SIV. Tanks are labeled T.
A-56
Arithmetic nodes or junctions. Junctions are labeled J.
B-57
H-58
Boundary nodes, Heater nodes and
HX-10
Line and HX macros
CPL
Connect Point
Super
Collection

Figure 6: Node and Lump Icon Types

Intermodel conductors and ties connect two submodels together, and they must therefore exist in two network diagram windows at once. To interconnect two diagram sheets, the user employs *connect points*, or special diamond-shaped icons. Connect points exist in

* With the advent of Sinaps*Plus*, the naming of conductors, ties, and paths has become much less important since they can be visualized. For these reasons, the user can let Sinaps*Plus* name them automatically by finding the next available integer identifier. Labels can be toggled on or off, singly or globally, as needed.

pairs, one in each window. Ties and intermodel conductors pass through these pairs, in effect disappearing through one connect point and reappearing through its twin connector in the other window. The user can create as many pairs of connect points as needed to clarify the network. As with nodes and lumps, they can be moved around the sheet arbitrarily. Connect points are visible as the diamond shaped icons in Figure 1. (Conductors and ties that pass through connect points are colored green by default if they are not owned by the current window.)

The pipe-shaped FLUINT macros icons are actually a type of connect point, since they connect into macro windows. Each Line or HX macro has its own unique window that offers options specific to those subnetworks.

The circular Collection is a tool for hiding the icons that appear on a network diagram. Once a collection is created, nodes or lumps may be moved into the collection so they no longer appear on the network diagram. Only those paths and conductors that connect to lumps and nodes inside the collection will appear. Collections may also be post-processed.

Complex Diagrams & Layout Options—Sinaps*Plus* has several powerful features intended to help the user depict large and/or complicated models within the constraints of a two-dimensional sketchpad. In addition to being able to relocate and resize icons, and to bend and shape conductors, paths, and ties, the user may employ *clones, layers* and *collections.*

An icon representing a node, lump or connect point may be split or *cloned* into multiple icons that can be independently placed on the sheet, and used equivalently to the original icon. An icon may be cloned as many times as desired. All clones of an icon are treated as a single entity for all pre- and postprocessing operations. In other words, SINDA/FLUINT solutions are produced based on the existence of a single element, and all cloned icons will have the same color during Sinaps*Plus* colorization operations.

Nodes and lumps that have been cloned may be merged back together. Any conductors, paths or ties associated with the deleted icon will be reattached to the remaining icon.

Layers are another utility for enhancing the clarity of network schematics. Each icon (node, cloned node, comment, etc.) is assigned to a particular layer. By default, only one layer named “top” exists. However, the user may create any number of user-named layers, and may position any of these icons on any such layer.

The user has control over each layer’s visibility. When the network is displayed, only those lumps/nodes that are in visible layers are shown. Further, only conductors/paths/ties between visible icons are displayed. A layer’s visibility can therefore be toggled on and off to hide portions of a network, or to show alternative depictions of the same network.

The utility of layers is greatly enhanced when used in combination with cloned icons, since clones of the same icon may reside on different layers. This allows the user to display multiple depictions of the same network, or to display “cross-sections.” An example of the combined use of clones and layers is shown in Figure 7.

Registers and Expressions—In almost all data fields in Sinaps*Plus*, expressions may be used instead of numeric constants. In other words, “0.25*pi*(1.0/12.0)^2” can be used to specify the area of a circle, as can its numeric equivalent “5.454e-3.” Common values such as π (“pi”) and conversion constants are available as “built-ins” through the pull down menus. Common functions (“sin(),” “ln()” etc.) may be used within these expressions, which follow normal Fortran and C rules of operator precedence. Expressions can also contain conditional formulae (equivalent to IF/THEN/ELSE in Fortran). The advantage of using expressions rather than numeric inputs include improved self-documentation and reduced errors.

A much more important advantage of expressions is that the model can be defined algebraically using *registers*. In addition to prestored constants such as “pi,” the user can define up to 5000 named registers (e.g.,

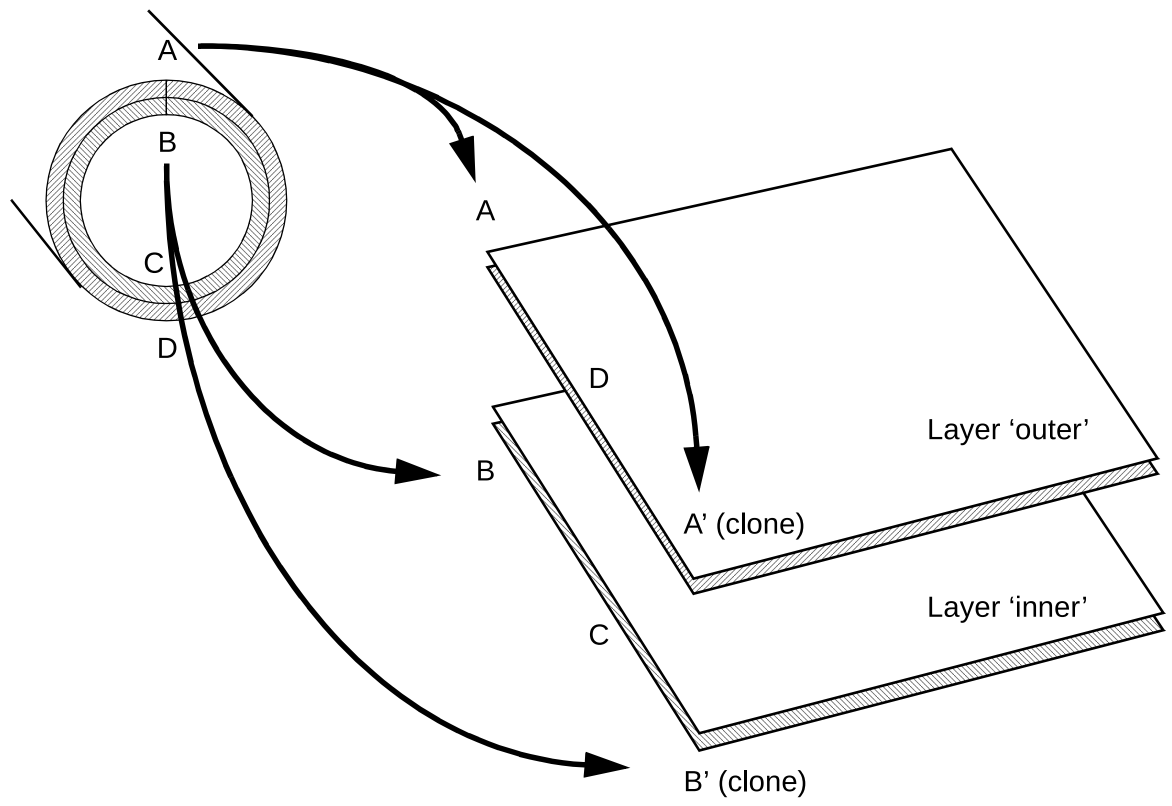


Figure 7: Example of Clones and Layers to Depict a Cylinder in 2-D

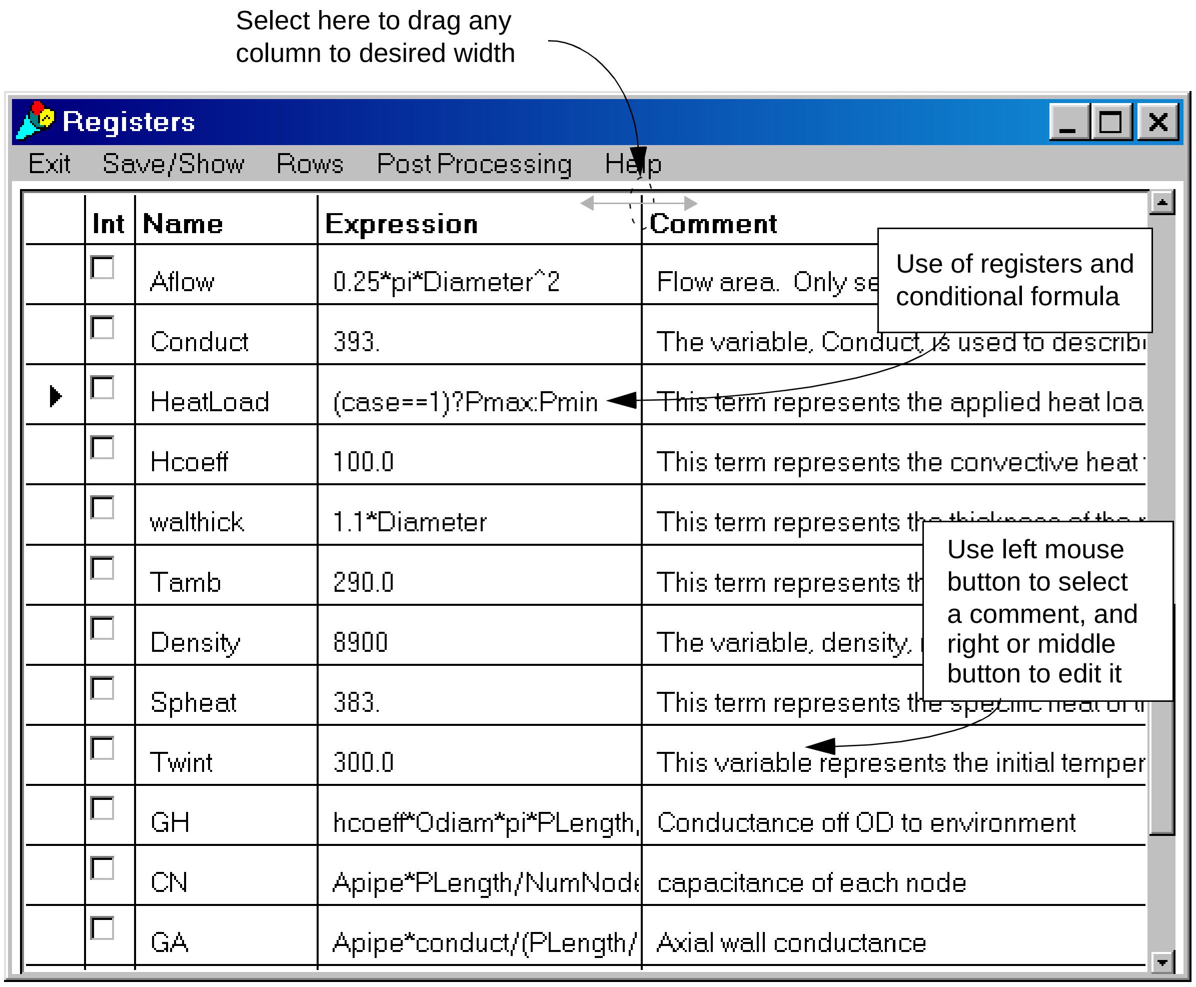
“area” or “freddy”). These names can then be used throughout the model instead of numeric data, or as part of expressions. For example, if a register named “diam” were created and were assigned the value “1.0/12.0”, the expression in the previous paragraph could have been specified as “0.25*pi*diam**2” (both “^” and “**” mean exponentiation). Figure 8 shows an example of an input form for registers.

Expressions can also refer to *processor variables* such as “mysub.T33” (the temperature of node 33 in thermal submodel mysub) and “timen” (the current problem time). Such values change while the problem is being solved, permitting complex, dynamic interrelationships between inputs and outputs to be defined. (Since processor variables have no value before entering the processor, any expression containing a processor variable is temporarily assigned a random number to allow the expression to checked for proper syntax.)

Registers, conditional formulae, and the processor variables are an enormously powerful and popular feature, since they enable the entire model to be built with variables instead of with fixed values. Once a model has been defined algebraically, parametric variations may be performed quickly, and dimensional or material property changes can be quickly and *consistently* applied throughout the model. The tutorial at the end of this document illustrates such parametric descriptions. The ability to import/export register data allow the same data base to be used for multiple models and image files.

In effect, registers make Sinaps*Plus* a cross between a spreadsheet and a thermal/fluid analyzer. Registers can also be used as a control panel for execution of prebuilts, license-free tools that can be written for third parties using Sinaps*Plus* as a development environment.

If the value of a register is changed, the new value will be used the next time expressions are evaluated. If a register was originally used to define an input but that register has been subsequently deleted, an error


Select here to drag any column to desired width
Registers
Exit
Save/Show
Rows
Post Processing
Help
Use of registers and conditional formula
Use left mouse button to select a comment, and right or middle button to edit it

	Int	Name	Expression	Comment
	☐	Aflow	0.25*pi*Diameter^2	Flow area. Only se
	☐	Conduct	393.	The variable, Conduct, is used to describ
▶	☐	HeatLoad	(case==1)?Pmax:Pmin	This term represents the applied heat loa
	☐	Hcoeff	100.0	This term represents the convective heat
	☐	walthick	1.1*Diameter	This term represents th
	☐	Tamb	290.0	This term represents th
	☐	Density	8900	The variable, density,
	☐	Spheat	383.	This term represents the
	☐	Twint	300.0	This variable represents the initial temper
	☐	GH	hcoeff*Odiam*pi*PLength,	Conductance off OD to environment
	☐	CN	Apipe*PLength/NumNode	capacitance of each node
	☐	GA	Apipe*conduct/(PLength/	Axial wall conductance

Figure 8: Example of Register Form

message will be produced the next time the input expression is evaluated. The user must then correct the defective expression before continuing.

Model Read-in—Sinaps*Plus* also possesses the ability to read in existing SINDA/FLUINT ASCII input files. Models may be read into empty model desktops. Sinaps*Plus* automatically reads and stores most of the information appropriately.

Network diagrams, however, have no analog in SINDA/FLUINT since that code has no underlying geometry. Therefore, these portions of the model are read in interactively with the user. During read-in the user has the option of manually placing each lump or node or allowing Sinaps*Plus* do place them automatically. The automatic placement will place the lumps and nodes in rows and columns across the network diagram. For the manual read-in, as each node or lump is read, the user places it in the appropriate diagram window (opened and managed automatically by Sinaps*Plus*). Once all such icons have been placed, Sinaps*Plus* automatically draws the paths, ties, and conductors as straight line connections, completing the model read-in operation.

The user can expect to spend some time cleaning up the diagrams: arranging and aligning icons, adding comments, and employing clones and layers to make the diagrams more legible or meaningful. Also, other Sinaps*Plus*-unique features, such as networks, can then be employed to make the model more maintainable. For these and other reasons, model read-in is intended to be a single event in the lifetime of a model. Once read into Sinaps*Plus*, the model should be maintained using Sinaps*Plus* and not modified externally. It is highly recommended that during the read-in of large models, the user maximizes the use of include or insert files.

Preprocessing and Input Checks

As a model nears completion and a network diagram exists, the user can use Sinaps*Plus* diagrams to check inputs. Input values may be viewed on these diagrams by requesting that some or all icons and connections be colored by a specific input value (e.g., conductors by conductances, lumps by pressures, nodes by capacitances, etc.). A scale or *color bar* will appear that depicts the colors associated with the displayed values. Like any icon, this color bar can be moved around the diagram. The selection set and color scale range, size, and orientation (vertical vs. horizontal) can be adjusted as needed to check specific inputs. A vertical and a horizontal color bar are visible in the bottom diagram of Figure 1.

To validate the model, the SINDA/FLUINT preprocessor consistency and validity checks can be performed inside Sinaps*Plus* via the Run SINDA/FLUINT→Preprocess Only option on the model control panel. This will launch the SINDA/FLUINT preprocessor only. If any errors (syntax errors, unbalanced parenthesis, etc.) are encountered, a window will pop up flagging the error to the user.

Running SINDA/FLUINT

Once a model has been generated and checked to the user's satisfaction, it may be sent to SINDA/FLUINT for execution. There are two ways of performing this hand-off.

First, the user may create a traditional SINDA/FLUINT ASCII input file based on the Sinaps*Plus* model. (Actually, these inputs may also be created at the submodel level for the purposes of checking inputs, or for exporting data to sites not employing Sinaps*Plus*.) This input file may be passed to SINDA/FLUINT, whether on the same or different machine.

If SINDA/FLUINT resides on the same machine as Sinaps*Plus*, a much faster and convenient method exists. The user can preprocess the model and launch its execution directly from Sinaps*Plus*. If SINDA/FLUINT or an appropriate compiler is not available on the current machine, or if the user chooses to send the model to a different machine, he or she should first preprocess the model (described above) in order to make sure that the foreign machine will be able to run the model the first time, without iterations.

Once SINDA/FLUINT has been launched with Run SINDA/FLUINT→Preprocess and Run→Normal Execution, it can be quickly relaunched (using Run Existing Executable) without having to recompile or relink if only minor changes to inputs or registers have been made. This feature not only saves time, it allows users to pass executable models ("prebuilts") to unlicensed users who lack a compatible compiler (see the prebuilt documents under separate cover) and/or SINDA/FLUINT.

Postprocessing Features

Once SINDA/FLUINT has executed to completion, any binary *save files* produced by that run can be used to postprocess the model using the original Sinaps*Plus* schematic. Save files created by the routines SAVE, RESAVE (restart), and CRASH routines can be used.

To postprocess a model, the user names the save file to be used within the current diagram window. In order to view data at a particular time or save file record number, the user may select from a list of times/records available on that file. The user may also plot results from multiple save files on the same plot to compare analysis results.

Colorization—Once the save file data is identified, the user may color nodes, conductors, etc. by their value at specific times or record numbers on the save file. Possible values include those available for input checks (e.g., temperatures, flowrates, etc.) as well as an extensive list of values calculated by SINDA/FLUINT or derived from those calculations, such as conductor or tie delta temperatures. As with input checks, the color bar and corresponding scale are editable. Nodes and conductors may be colored separately, as can lumps, paths, and ties. Thus, up to three color bars may exist on the window at one time.

Thickening—Simultaneous with color options, the user may choose to thicken paths, ties, and conductors by the same or different values. In other words, the color of each conductor might be set to its current conductance, while its thickness might be set to its current heat rate. Thicknesses are often more intuitive than colors. More importantly, this feature enables “visual QMAPs,” whereby all of the relevant information contained in the text output routines QMAP and FLOMAP can be viewed as colors and thicknesses.

Each thickness operation results in a thickness scale for calibration. A thickness scale can be seen in the lower right hand portion of the bottom diagram in Figure 1. Like color bars, these thickness scales can be customized.

Animating and Perusing—Once colorizations and thickness operations have been specified, the user can move through the save file manually or automatically. In other words, the user can move to a new time point (save file record), and the screen will update accordingly. The current time is displayed in a special temporary comment box.

The user can also animate the model results, moving forward or backward through the save file, watching the network change thicknesses and colors. Before performing such animations, it is recommended that the color and thickness scales be changed from the default autoscaling to fixed limits. Otherwise, since the color bars and thickness scales will reset themselves to cover the range of data present, the resulting animation is often reduced to a change in scale without a significant change in icon colors or thicknesses.

Plots—Sinaps*Plus* features an extensive plotting package. Users can request data for some or all icons (or registers, using the features in the Register form) be plotted in X-Y coordinates as functions of problem time, position,* steady state iteration count (LOOPCT), Solver iteration count (LOOPCO), or any register. The user can then customize the plot, which appears in a separate window that can be saved independently of the model (for the purposes of archiving, transmitting to other sites, or saving for future editing), or exported as a GIF (graphics interchange format) file. When the plot window is resized, the plot resizes itself automatically.

Customizations include adding axis labels, annotations, and extra lines (pointers), coloring lines, editing and moving legends, etc. The scale and units of each axis may also be changed. The user may also create plot templates to be saved and used for subsequent plots.

Advanced plotting options are available to the user through the Plot Control Panel.These options include the ability to plot from multiple networks and submodels, plot register values, and the ability to change save files or simultaneously plot from multiple save files.

In addition to X-Y plots, the user may choose to employ polar plots. Polar plots are convenient for systems that are cyclic, either intentionally (e.g., periodic systems such as orbital spacecraft) or incidentally (e.g., hydrodynamic oscillations).

Information about the model at the current time may also be represented as bar charts. A special feature of bar charts is especially important: node and lump balances. The energy and mass flows through selected nodes and lumps can be displayed via bar charts, with optional legends providing additional data. These charts are the visual equivalent of the NODMAP and LMPMAP routines in SINDA/FLUINT.

Examples of plots may be found in the sample problem description.

* Since neither Sinaps*Plus* nor SINDA/FLUINT have dimensional geometry, “position” means location of the icon on the screen, in units of pixels. Using alignment features and plot axis filter expressions, accurate plots of values versus linear dimension can be easily made, as explained in the full User’s Manual and illustrated in the sample problem.

Text Outputs—Users may elect to view tabulations of data instead of plotted or colored results. Also, plot data may be exported as ASCII files for import into third party software.

Why Sinaps*Plus*?

The goal of Sinaps*Plus* is to completely replace the traditional ASCII in, ASCII out, batch-style method of employing SINDA/FLUINT, and to make the powerful simulation capabilities of SINDA/FLUINT accessible to a wider group of analysts. Sinaps*Plus* enables analysts to:

1. **Reduce Errors.** It is much more difficult to create an erroneous or nonsensical model when using Sinaps*Plus*. While large models are notorious for hiding mistakes, it has been found that even small models can be problematic when visual representations are lacking. One such model, a SINDA/FLUINT sample problem with only about 20 nodes and 15 lumps, was found to be in error once brought into Sinaps*Plus*, despite having existed in publicly documented form for many years.
2. **Improve Productivity.** Fewer iterations are spent arriving at a model that is both legal and adequate to its purpose, and more flexibility can be built into the model. Sinaps*Plus* reduces the amount of time required to find modeling mistakes that affect the results of the analysis.
3. **Improve Results Interpretation.** Important results and trends can escape unnoticed without graphical results presentations. For example, the cause of a spike within a cyclic pump-valve oscillation was not evident using only traditional SINDA/FLUINT methods, even though the spike existed in test data. After bringing the model into Sinaps*Plus* and postprocessing it, the spike became immediately evident and was quickly explained as a laminar to turbulent transition. Earlier text-based investigations had simply failed to look in the correct locations, whereas Sinaps*Plus* enabled global trends to be immediately recognized.
4. **Make Self-documenting Models.** Inheriting another analyst's thermal/fluid model is often a cause for great concern, even assuming supporting documentation has been written, is correct, and has not been lost. Sinaps*Plus* eliminates this problem, since the diagrams are self-documenting. Opportunities for additional comments, expression-based inputs, etc. also exist to enable more complete documentation. Also, reviewers can interactively review provided model results, and can even generate new results.

   In fact, the recipient need not be a licensed user. Licensed users can pass their models on to managers, customers, etc. in Sinaps*Plus* form. The recipient will have full access to Sinaps*Plus* viewing and postprocessing systems, and will even be able to generate new results if a prebuilt model is supplied. They simply will not be able to save the results of any changes they make to the model. In effect, *licensed users can provide living documentation of their models and results in a controlled manner.*
5. **Transport Model Data, Plots, and Diagrams**. SINDA/FLUINT need not reside on the same machine, much less the same type of machine as does Sinaps*Plus*. Furthermore, images, model desktop files, and plot files are all stored in a machine-independent binary format. This means that the user can move from machine to machine without interruption, and that multiple users can cooperatively build and postprocess models even if they use different types of machines.

In addition to visualization and form-based inputs, Sinaps*Plus* also offers several capabilities that are not available in SINDA/FLUINT. These include:

1. the ability to create prebuilt models as tools, perhaps for use by others;
2. the ability to skip compile and link steps for small model changes;

3. the ability to initialize model inputs from the results of previous runs;
4. the ability to customize the individual elements within a FLUINT LINE or HX macrocommand*;
5. the ability to define and reuse subnetworks (user-defined components);
6. the ability for different models to attach to a single reference submodel (i.e. changes made to that submodel are automatically propagated to all models that employ it);

Sinaps*Plus* Sample Problem Description

This section describes a simple SINDA model that will be developed using Sinaps*Plus*. A demonstration of the same problem, using the traditional ASCII input file approach, is available in a separate document, “Introduction to SINDA” available from www.crtech.com. Modeling decisions and other engineering topics are covered in that document in more detail. In this document, the focus will be on Sinaps*Plus* usage rather than on SINDA/FLUINT usage.

Problem Description

Consider a cylindrical rod with constant thermal properties ($\rho = 8000$ kg/m^3, k = 15 W/m-K, $C_p = 500$ J/kg-K) that is coated with a paint whose infrared emissivity is $\varepsilon = 0.3$. The length of the rod is 1.0 meter, and the diameter is 1 cm.

The rod is used to suspend a 40K cryogenic vessel inside of a 300K vacuum chamber. For the purposes of this problem, the effective radiation sink temperature within the vacuum is specified to be 110K. See Figure 9.

What is the heat leak into the vessel?

Model Description

The problem can be solved entirely by steady-state solutions, since no transient event exists. The rod may therefore be represented by zero capacitance (massless) arithmetic nodes. (In steady state solutions, diffu-

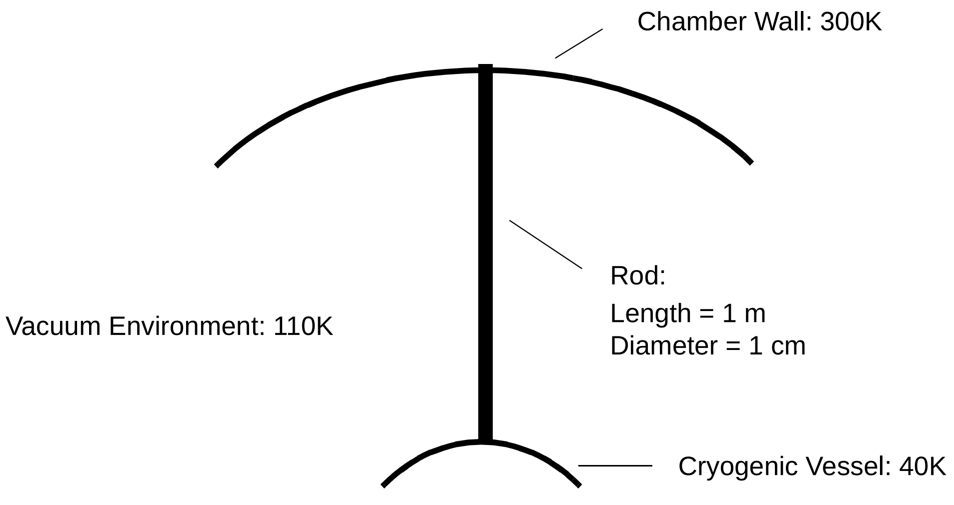


Figure 9: Sample Problem Schematic

* Due to the ability to customize macros within the network window instead of user logic, SINDA/FLUINT models containing macros which have been exported to ASCII file format cannot be read back into Sinaps*Plus*.

sion nodes are be treated as arithmetic nodes, so their capacitances will therefore be ignored in this case.) Nonetheless, diffusion nodes will be used in this document for demonstration purposes.

Nodes—The problem has three boundary conditions: the chamber wall, the effective vacuum environment, and the vessel wall. Each will be represented by boundary nodes that are labeled 1000, 2000, and 3000 respectively.

Because of the long aspect ratio of the rod, this is clearly a one-dimensional problem. The rod will be divided into 10 equal lengths, represented by nodes numbered 1 through 10. (For demonstration purposes, the discretization of the rod has been purposely underestimated.)

Conductors—There are two types of heat flow paths: axial conduction along the rod, and radiation exchange between the surface of the rod and the vacuum environment. Linear conductors 1 through 11 will be used to represent solid conduction within the rod. Note that the first and last conductors have twice the conductance of the others since they represent half the length of a typical node.

Radiation conductors 201 through 210 will be used to represent the exchange between the rod and the vacuum environment.

The above nodes and conductors will be placed in a single submodel named “ROD.”

Standard metric (SI) units will be employed in this model, with temperature in degrees Kelvin. The value of absolute zero temperature (“ABSZRO”) is therefore zero, and the Stefan-Boltzmann constant (“SIGMA”) is 5.67E-8.

Solution Sequence—The solution sequence is defined in the OPERATIONS data block of SINDA/FLUINT. The first step in a solutions sequence is to define the model. This is done through a BUILD statement which identifies an arbitrary build configuration “MYMODEL” followed by a list of submodels as shown below.

```
BUILD MYMODEL, ROD
```

Alternatively, since all defined thermal submodels (in this case it happens to be only one) are to be built, the build statement can be defined as follows.

```
BUILD ALL
```

In this particular sample problem, a single steady-state run is needed. Such solutions are requested by calling single routines such as:

```
      CALL STEADY
```

The resulting OPERATIONS block becomes:

```
BUILD ALL
      CALL STEADY
```

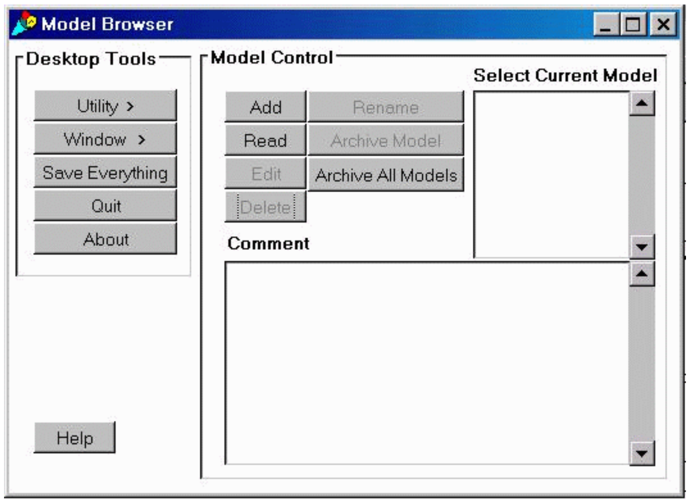
Sinaps*Plus* Tutorial

This section describes the step-by-step process for creating and postprocessing the model described in the previous section. Note that the exact size and appearance of the windows will vary slightly from machine to machine, depending on the host windowing system.

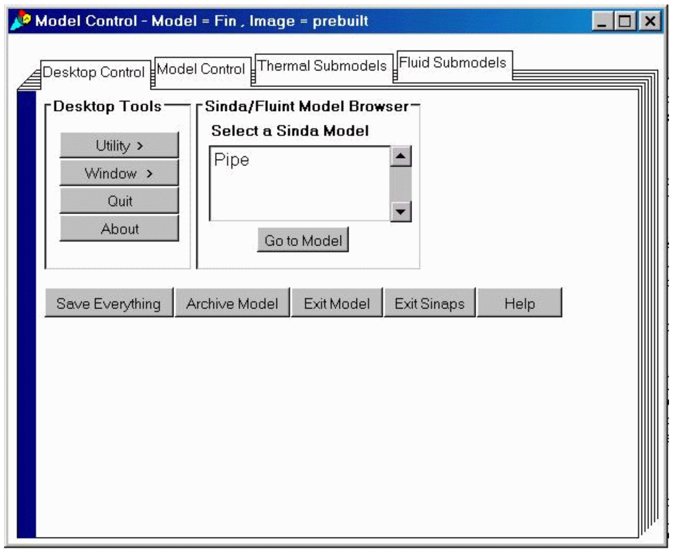
Starting Sinaps*Plus*, and Creating a new Sinaps*Plus* Model Desktop

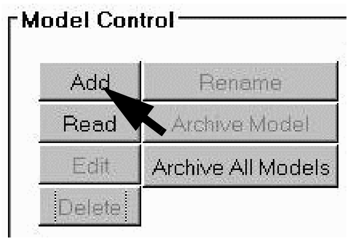
To run Sinaps*Plus*, the user must first enter the windowing system. On a PC Sinaps*Plus* may be started by finding the appropriate Sinaps*Plus* image icon (perhaps under the Start menu) and double clicking it. On Unix machines, the user should type the following sequence in a command shell:

```
sinaps myimage.im &
```

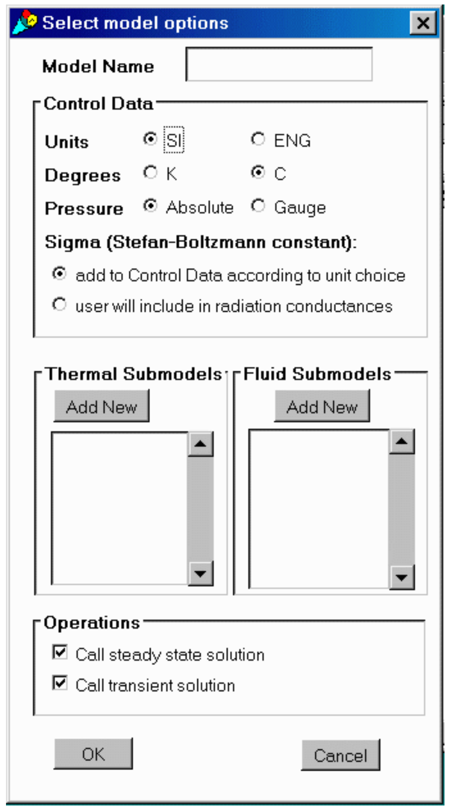


where myimage.im is the name of the desired image file. Sinaps*Plus* will then start, with windows appearing on the screen that are appropriate to the desktop that was visible in that image when it was last saved. This desktop might be the top-level desktop, or it might be a model desktop. For this tutorial it will be assumed that a new (empty) image has been opened. Upon opening Sinaps*Plus*, the top-level *model browser* window will appear as shown at the above right. Since the image is empty, no models appear in the model list in the upper right portion of the window. If the image is not empty and you see a window similar to the one shown below, you need to click Exit Model on the lower menu in the window to return to the model browser window.





Simple Model Building Template- In the top-level model browser window, locate the model control menu and click on Add as depicted on the left. You will be prompted to Create an empty model or to use the model template builder. This option allows the user to access a building template or wizard to create models within SINDA/FLUINT. The template builder will create the basic inputs for a steady state or transient model using user specified thermal and/or fluid submodels. The template builder defines the submodels to be built but will not create the networks associated with each submodel since these will be specific to each model. Click on Template when prompted.


Select model options

Model Name

Control Data

Units [✓ SI] ENG

Degrees K [✓ C]

Pressure [✓ Absolute] Gauge

Sigma (Stefan-Boltzmann constant):

[✓ add to Control Data according to unit choice]

user will include in radiation conductances

Thermal Submodels — Add New

Fluid Submodels — Add New

Operations

[✓ Call steady state solution]

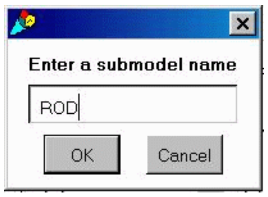
[✓ Call transient solution]

OK Cancel

The *select model options* window will appear next as shown on the left. First we need to supply a model name. In the input field at the top of the form, type “STRUT”. Next, we need to define the units for the model. Our sample problem is defined in SI units with temperatures specified in degrees Kelvin so we need to click on the radio buttons labeled “SI” and “K”. We have no fluid submodels in this problem so we can ignore the “Pressure” radio buttons. Whenever a set of radio buttons appear in a window, at least one must be checked.

SINDA/FLUINT allows you to define the Stephan-Boltzmann constant (“sigma”) to be applied to all radiation conductors. If you have a set of RADKs output from a radiation analysis package such as RadCAD®, the conductors may already have sigma included. If this is the case you need to select “user will include in radiation conductances”. For this problem we will click on the “add to Control Data according to unit choice” radio button.

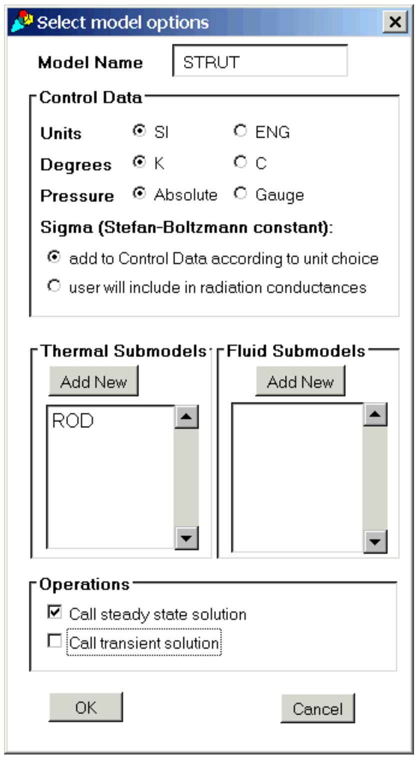
Next we need to define our submodels. In this case we will only have one thermal submodel. (SINDA/FLUINT and Sinaps*Plus* are designed to handle many more submodels. Most models will usually have at least two or three submodels.) Select Add New under the Thermal Submodels portions of the window. When the pop-up query form appears, supply the name “rod” as the submodel name. The window will now appear as shown at the right below. Click “OK” to save the data and dismiss the window.


Enter a submodel name

ROD

OK Cancel

For this problem we are only interested in obtaining a steady state solution. In the Operations portion of the select model options window, we need to uncheck Call transient solution. If both Call steady state solution and Call transient solution are checked the model will be setup to call a steady state simulation first, followed by the transient simulation. The completed *select model options* window should now appear as shown on the left below. Click OK at the bottom of the window to accept these inputs. The *model browser* window should reappear.



Select model options

Model Name: STRUT

Control Data

Units: [✓ SI] ENG

Degrees: [✓ K] C

Pressure: [✓ Absolute] Gauge

Sigma (Stefan-Boltzmann constant):

[✓ add to Control Data according to unit choice]

user will include in radiation conductances

Thermal Submodels: ROD

Fluid Submodels:

Operations

[✓ Call steady state solution]

Call transient solution

The model browser window is the same as shown at the top of Figure 4 in this tutorial. Click on the button labeled Exit Model at the bottom of the form and you will move to the top-level model control panel. You should see the model name, STRUT, in the model list portion of the window. Double click on the model name STRUT in the model browser to return to the model-level control panel.

At this point we have created a simple model structure using the building template. Basic output file names have been defined, as have simulation options etc. However no network has been created and no registers have been defined. Appendix A provides details of what has been created with the wizard.

Save Everything—Periodically, you should save your work to disk instead of just to memory. Now is a good time. Select Save Everything from the Model Control Panel. By default, the current image file name appears. If this name is “startim,” the starting image file name supplied with Sinaps*Plus*, you must create a new one, perhaps “myimage”. Hit return or press the OK button, which will commence the disk writing operation. Note: ***Save Everything*** *is disabled in the evaluation version* of Sinaps*Plus*.

To start up SinapsPlus the next time using this new image file name, use “sinaps myimage.im &” on Unix machines. On PCs, the SinapsPlus icon must be edited to reflect this new image file name, as detailed in the installation instructions or simply double-click the image file name in Windows Explorer.

Input Generation

This subsection describes the generation of the new SINDA/FLUINT model within Sinaps*Plus*. To a large extent, the exact sequence of the following operations is actually up to the user**.**

Registers—Although not a strict requirement, it is extremely convenient to define any significant variables, dimensions, properties, etc. up front as registers. While these registers can be defined or redefined at any time (as can the expressions in which they are used), forethought results in greater model versatility and in less work.

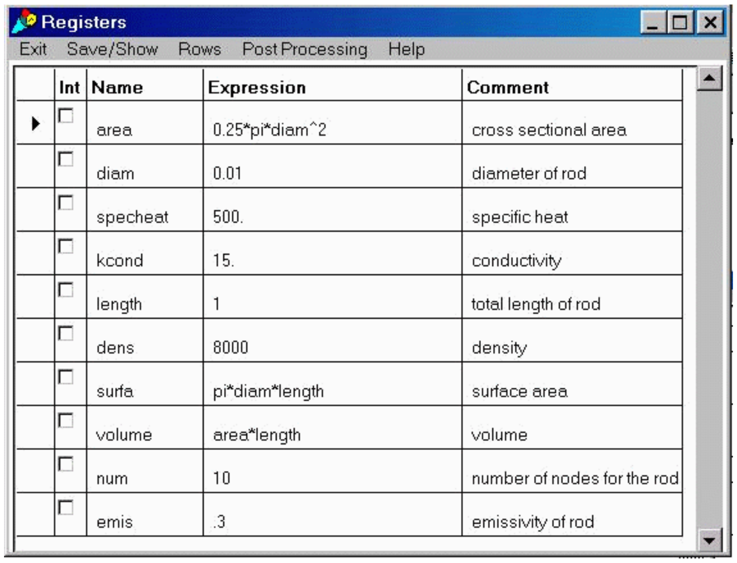
In this problem, key dimensions as well as material properties[*] will be input as registers, making them easy to change later. These include the rod dimensions (diameter ‘diam’, cross sectional area ‘area’, surface area ‘surfa’, length ‘length’, and volume ‘volume’) and properties (conductivity ‘kcond’, density ‘dens’, and specific heat ‘specheat’). Even the number of nodes will be specified as “num” since doing so enables the model resolution to be more quickly changed. The surface emissivity will be specified as “emis.”

The register form may be accessed via the Model Control tab of the *Model Control Panel* under Registers. The form is depicted in the completed state below on the left. Note that the expressions use prestored values[†] such as “pi,” and that they include the values of other registers in their definitions.

Once the expressions are input, save the data using Save/Show→Save Values from the pull down menu bar. If an expression has been input illegally, a message will advise as such. Correct the defective expression and try saving again.

* Actually, temperature-dependent properties are often more convenient to define using processor variables (sub.T22) or in ARRAY DATA such that SINDA options can be exploited. Even within ARRAY DATA, registers can be employed.

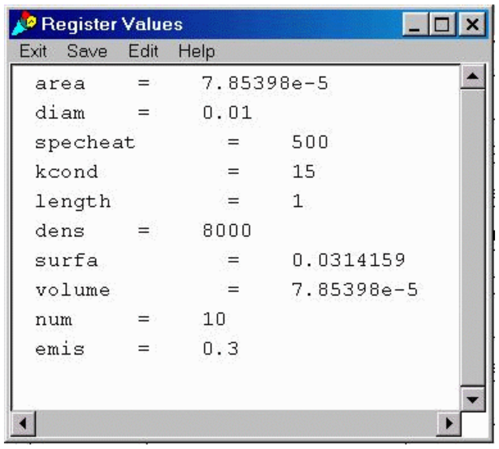
† A full list of built-in constants and functions may be found in the Sinaps*Plus* User’s Manual.



Registers

Exit Save/Show Rows Post Processing Help

Int	Name	Expression	Comment
☐	area	0.25*pi*diam^2	cross sectional area
☐	diam	0.01	diameter of rod
☐	specheat	500.	specific heat
☐	kcond	15.	conductivity
☐	length	1	total length of rod
☐	dens	8000	density
☐	surfa	pi*diam*length	surface area
☐	volume	area*length	volume
☐	num	10	number of nodes for the rod
☐	emis	.3	emissivity of rod



Register Values

Exit Save Edit Help

```
area     =     7.85398e-5
diam     =     0.01
specheat       =     500
kcond          =     15
length         =     1
dens     =     8000
surfa          =     0.0314159
volume         =     7.85398e-5
num      =     10
emis     =     0.3
```

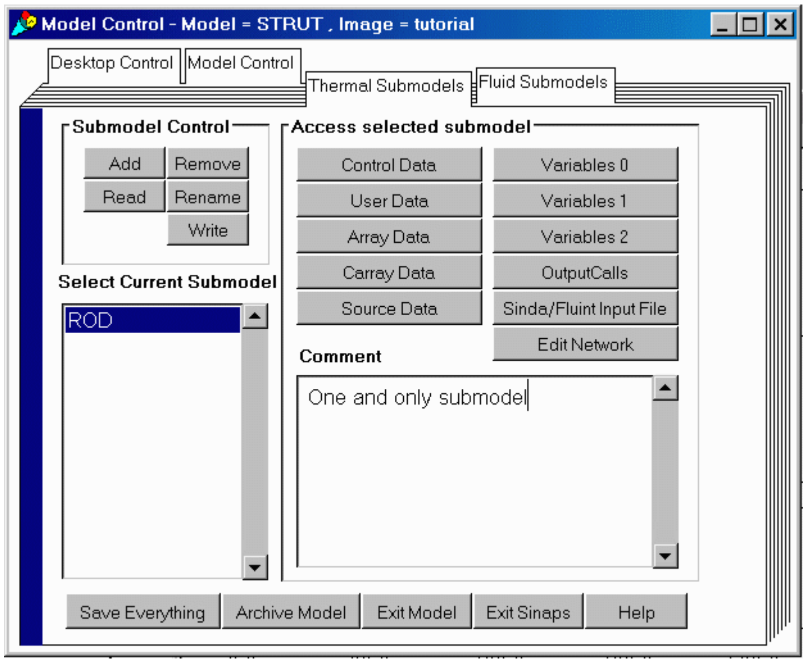
To validate your inputs, select the Save/Show→Show Values. This will create a pop-up window that displays the current values of each register. The window is shown above on the right. Before proceeding, make sure your registers are producing the correct values. If the values are correct, click on Exit→Quit Window in the Register Value window and Exit→Quit Window in the Register Window (we previously saved the values).

System Transcript—A useful tool for the novice SinapsPlus user is the *System Transcript* window. Click on the Desktop Control tab of the *Model Control Panel* and then click on Utilities→System Transcript in the *Desktop Tools* menu. A blank window will then appear and you should place it in a location in which it remains visible (perhaps the extreme upper right of your screen). If Sinaps*Plus* is expecting a response from the user, such as during network development, the user will be prompted for the expected action in the system transcript window.

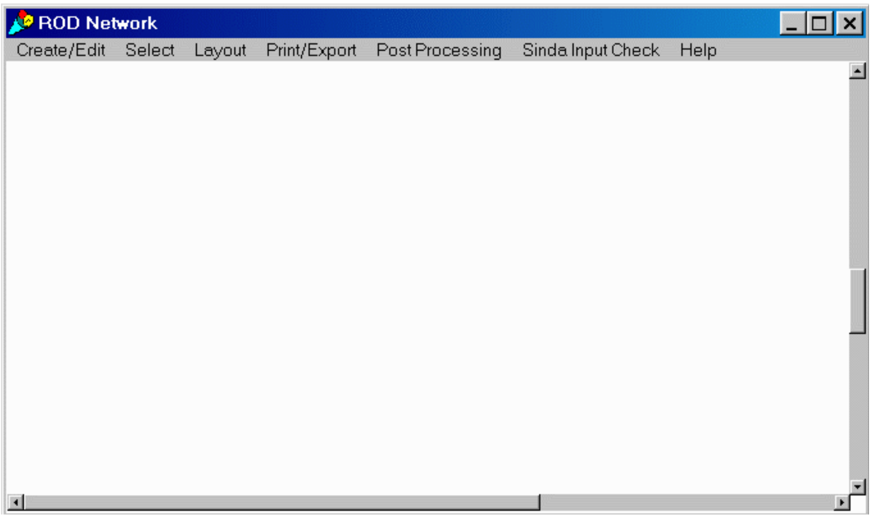
The *model control panel* should appear similar to that shown in the lower left of Figure 4. Note the title of the window should read as “Model Control - Model = STRUT, Image = myimage.”

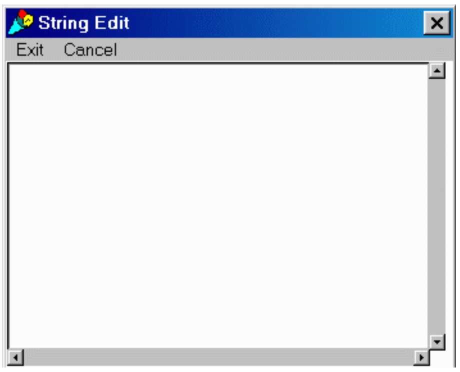
This identifies that you are in the “model control” window of the “STRUT” model contained within the “myimage” image.

Submodel Control Panel—Each submodel in SINDA/FLUINT represents a network of nodes and conductors for a thermal submodel or lumps, paths, etc. for a fluid submodel. Since each submodel has a unique network, access to the network is provided through this submodel control panel, not the model control panel. In the *model control panel* click on the tabbed form labeled Thermal Submodels and the *thermal submodel control panel* will appear as shown at right. (Note that a comment has been added to provide some description of the submodel. To add a similar comment, highlight the submodel name in the list and then click and type in the comment field.) This single control panel provides access to all thermal submodels defined within the model. The *submodel control panel* is similar to the *top-level model browser* in that it contains a submodel browser window (in the lower left of the panel) and the ability to add, rename, remove, import/export submodels. The functionality of this window is also similar to the *model control panel* in that it gives the user access to all submodel specific data such as array data, heat sources, logic blocks, and the submodel network. In our case we only have one submodel so only one item appears in the submodel browser. If multiple thermal submodels were present they would all be listed in the browser portion of the control panel. In such a case whichever submodel is currently highlighted is considered to be the active submodel. This feature provides fast, convenient access to all submodels from a single control panel.



Thermal Submodel Creation—Next we will generate the network portion of our submodel ROD. With the submodel ROD highlighted in the browser, click on Edit Network in the control panel and a blank drawing field will open as shown at the right. The network diagram contains the equivalent information of both the traditional HEADER NODE DATA and the HEADER CONDUCTOR DATA blocks. Most importantly, it also contains a diagram of the network. The top-level menu bar contains many options and leads to many more suboptions, only a few of which will be covered here.

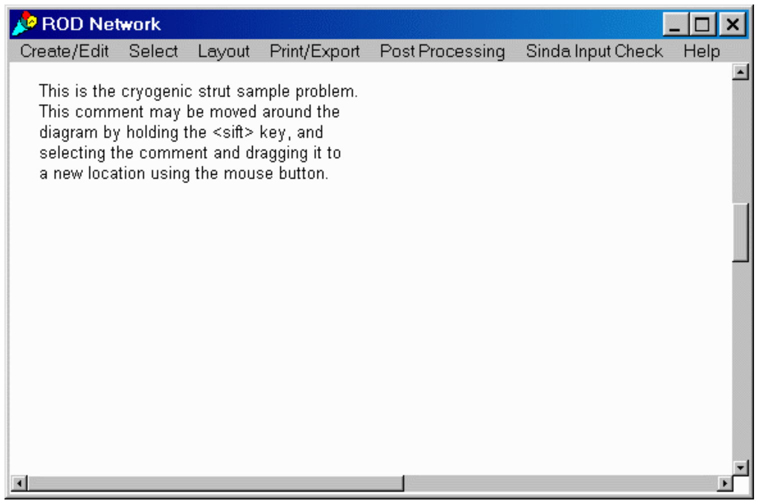




To get a feeling for how to manipulate items in this window, first create a comment by selecting Create/Edit→Comments→Add from the menu bar. The String Edit window will appear as shown at the right.

Type some text into the comment form. Simple editing may be performed, and multiple lines may be added.

Once you are satisfied with the comment, click Exit→Save and Exit from the pull down menu. The comment will appear as you define it when the mouse cursor is within the network diagram window. It will follow the mouse until you press the mouse button, at which location the comment will be “dropped” onto the sketch pad. A comment added to the top left hand corner of the window might look as shown below at the left.

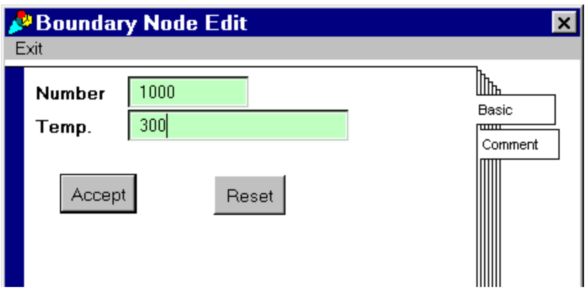

ROD Network
Create/Edit Select Layout Print/Export Post Processing Sinda Input Check Help

This is the cryogenic strut sample problem.
This comment may be moved around the
diagram by holding the <sift> key, and
selecting the comment and dragging it to
a new location using the mouse button.

If you aren’t satisfied with the comment, and wish to edit it, select it with the mouse button. A red square will appear in the upper left hand corner to confirm the selection. Then, choose Edit from underneath the Create/Edit→Comment pull-down. The original comment edit window will appear to accept any changes. Or, as a shortcut, simple double-click the mouse over the comment to pop-up its edit form. In other words, a single click selects the comment, whereas a double click edits it.

To move the selected comment, the Move option under the Layout pulldown can be used. However, this option is intended to move many items at once.

To quickly move any single item (such as this comment) within the diagram, hold down the shift key, located the cursor over the item and depress the mouse button. As long as the mouse button is depressed, the item will follow the mouse motions. (The use of the shift key is required to distinguish this action from a select action).

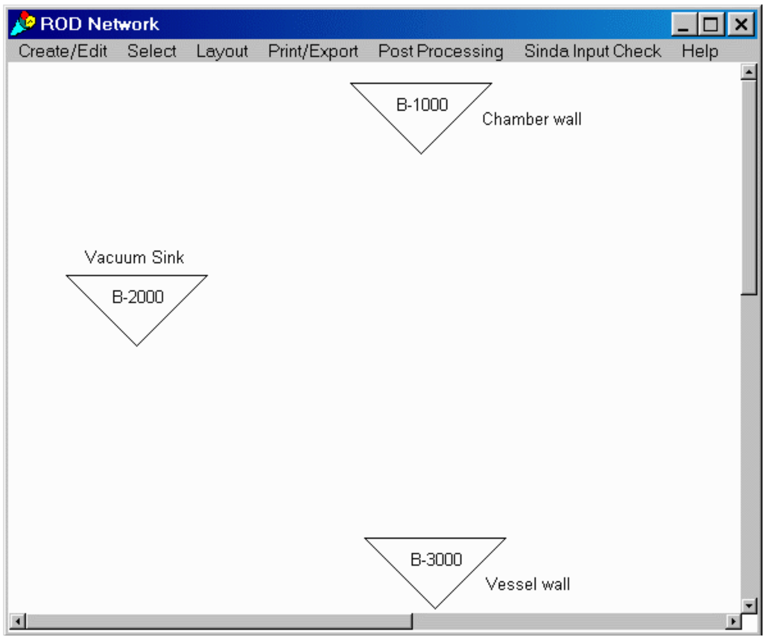


Number 1000
Temp. 300

Adding Nodes—We are now ready to add nodes to the diagram, staring with the boundary nodes. To add individual nodes, choose the option Create/Edit→Nodes→Add Single→Boundary from the pull down menu. The form at the right will appear. Make the node number 1000, and the temperature 300.0 (for the chamber wall). The user must click on Accept to apply the changes. The tabbed window allows access to a text window for adding user defined comments. These comments do not appear in the network diagram but are useful for model documentation.

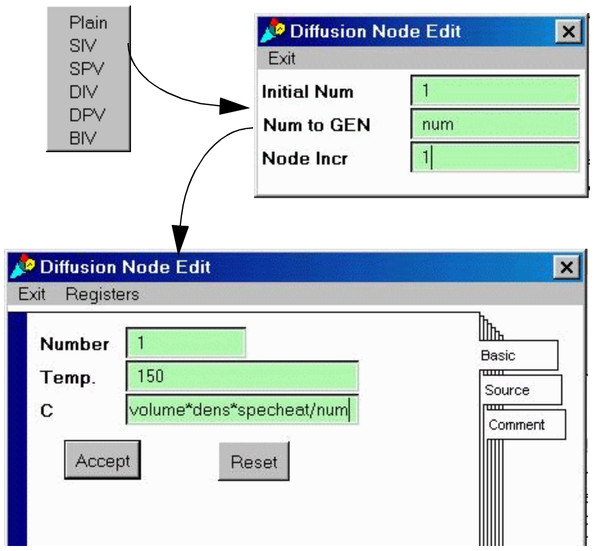
When Exit→Quit Window and Create is chosen from the pull-down menu, the form will disappear and a triangular icon with a “B-1000” in the middle will appear. Drop this icon near the top of the screen.

Repeat this process adding boundary nodes number 2000 (at 110K, representing the vacuum sink) and 3000 (at 40K, representing the vessel wall). Adding comments to help identify each node, the diagram might look the one presented at the right.


ROD Network
Create/Edit Select Layout Print/Export Post Processing Sinda Input Check Help
B-1000
Chamber wall
Vacuum Sink
B-2000
B-3000
Vessel wall

We are now ready to add the diffusion nodes that represent the rod. Instead of generating them individually (Add Single) we might use the Add Multiple feature, which continues to add more nodes of the same type until the user selects “Exit.”

Since the rod nodes are identical, an even more convenient method is the Add GEN feature, which is equivalent to the GEN command (as well as SIM, SPM, etc.) in SINDA. Selecting this option (Create/Edit→Nodes→Add GEN→Diffusion) leads to the pop-up form shown below at the left. Selecting Plain (meaning that temperature-dependent capacitance options are not needed) leads to the form at the upper right. Specify 1 as the initial node number, NUM (set to 10 in the registers) as the number to generate, and 1 as the node number increment, as shown below at the right. Saving and exiting this form results in the next form (lower left). The initial temperature (which will be discarded by the SINDA STEADY solution) is specified as 150.0, and the capacitance for each node is defined using the expression “volume*dens*specheat/num” (equating to volume times density times specific heat divided by the number of nodes) using predefined registers. The final form should appear as shown on the below at the left before saving. Accept the changes and then Exit→Quit Window and Create (or use the short cut Alt+X). Note that many menu options also have keyboard shortcuts. The shortcuts are listed on the menus to the right of the option.



Initial Num	1
Num to GEN	num
Node Incr	1

Number	1
Temp.	150
C	volume*dens*specheat/num

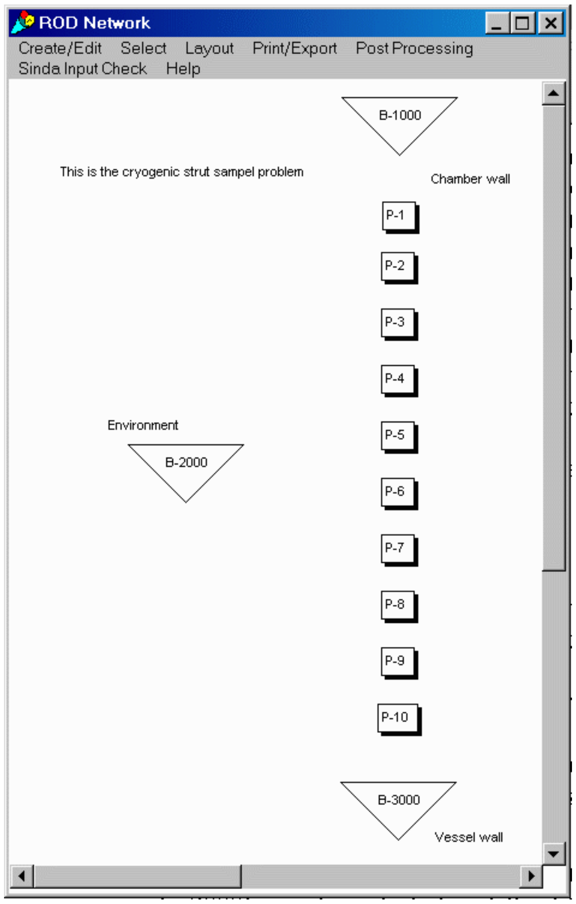
A drop-shadow rectangle (the symbol for tanks and diffusion nodes per Figure 6) with the label “P-1” should appear under the mouse, waiting to be dropped with a mouse click (using the same method by which the boundary nodes and comment were placed). Place node 1 near the top of the window. Nodes 2 through 10 will appear sequentially for placement. Place them below the previously placed nodes in a vertical stack, similar to the diagram below at the right.

Before generating conductors, let’s clean up this representation. First, the icons are rather large, so the pull down menu option Layout→Icon Size→Smaller is performed.*

Next, resize the window to make it taller and thinner. This operation will vary depending on your host windowing system, but is usually performed by selecting a corner of the window and dragging.

* Sizes will vary from machine to machine, so don’t perform this next step if you think the nodes are already too small.

The next step is to arrange the icons in approximate agreement with the physical layout presented in Figure 9. First, select the rectangular diffusion nodes, either by clicking on them individually with the mouse, or via an areal drag select. To use the latter method, hold down the mouse button, and drag it across the screen until the desired nodes have been enclosed in the resulting rectangle. Either way, all of these nodes should have a red square next to them. If some do not, click on them with the mouse button. If any other icons were inadvertently selected, click on them to toggle their red square off, thereby unselecting them.


ROD Network
Create/Edit Select Layout Print/Export Post Processing Sinda Input Check Help
B-1000
This is the cryogenic strut sampel problem
Chamber wall
P-1
P-2
P-3
P-4
P-5
P-6
P-7
P-8
P-9
P-10
Environment
B-2000
B-3000
Vessel wall

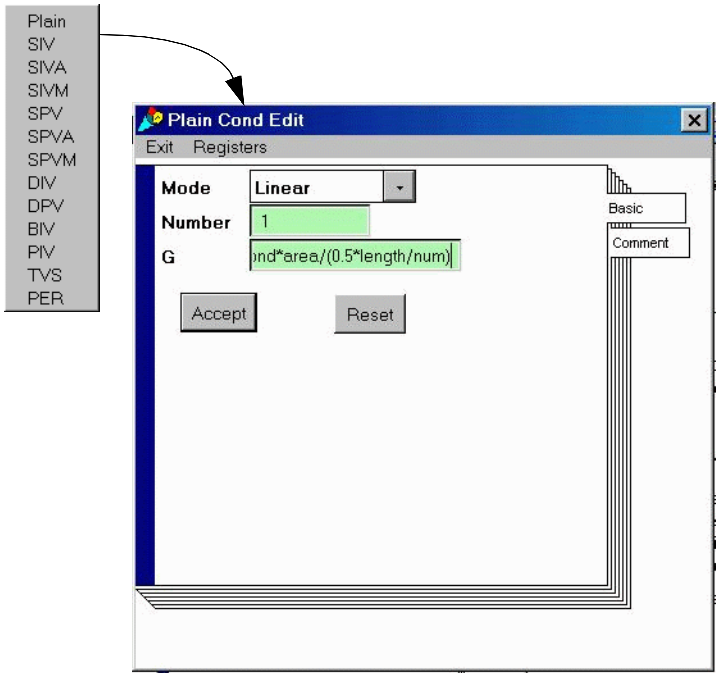
Next, select Layout→Move from the pull-down bar. A temporary picture of the selected nodes will appear under the mouse, waiting for the user to drop them in the desired location (the center of the window. When dropped, the original set of icons is erased, and the move operation is complete. Moving the boundary nodes, too, results in a diagram as shown at the left.

The vertical string of 12 nodes (excluding boundary node 2000), can be selected and the Layout→Align→Vertical even spacing option chosen to distribute them evenly, tidying up the picture. The comment has been edited and abbreviated.

What if many more nodes were needed, making the diagram hard to understand? As can be seen in the scroll bars, the window is actually a view of a much larger sheet upon which larger models may be placed.

The user can also resize this underlying sheet if needed (under Layout→Sheet size), and employ layers and clones and collections to clean up the depiction. These important options will not be covered in this tutorial, but the user should know they exist.

Adding Conductors—We are now ready to generate conductors. First the linear conductors will be generated. While the middle 9 conductors may be generated with an "Add GEN" command, the first and last will be generated individually. The conductance of these two linear conductors will be equal to "kcond*area/(0.5*length/num)," whereas the conductance of the middle 9 conductors is "kcond*area/(length/num)." (Once again, time and temperature variations, one-way conductors, and other such options will not be needed.)

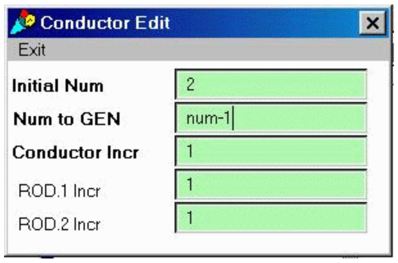


Mode	Linear
Number	1
G	›nd*area/(0.5*length/num)

To generate the first conductor between node 1000 and node 1, select Create/Edit→Conductors→Add Single. A form similar to the node pop-up form will appear (shown far left). Select Plain and a Conductor Edit form will appear (left). This form features a pull-down Mode bar, which by default contains Linear, which is the type we need. Leaving that bar alone, fill in the other two fields such that they appear as shown at left. The full expression input should be "kcond*area/(0.5*length/num)."

After accepting the changes and exiting this form, a cross-hair selection cursor will appear. This cursor is waiting for you to select the two nodes to which this first conductor will connect (as prompted by the message in the transcript window). Move the cursor over node 1000 and push the mouse button. A "splotch" will appear on the node, confirming the selection. When node 1 is similarly selected,[*] the splotches and the selection cursor will disappear, and a conductor will also appear between the two nodes. This process can be repeated to generate conductor 11 between nodes 10 and 3000. (The more adventuresome user can instead use the conductor copy and paste operations.)

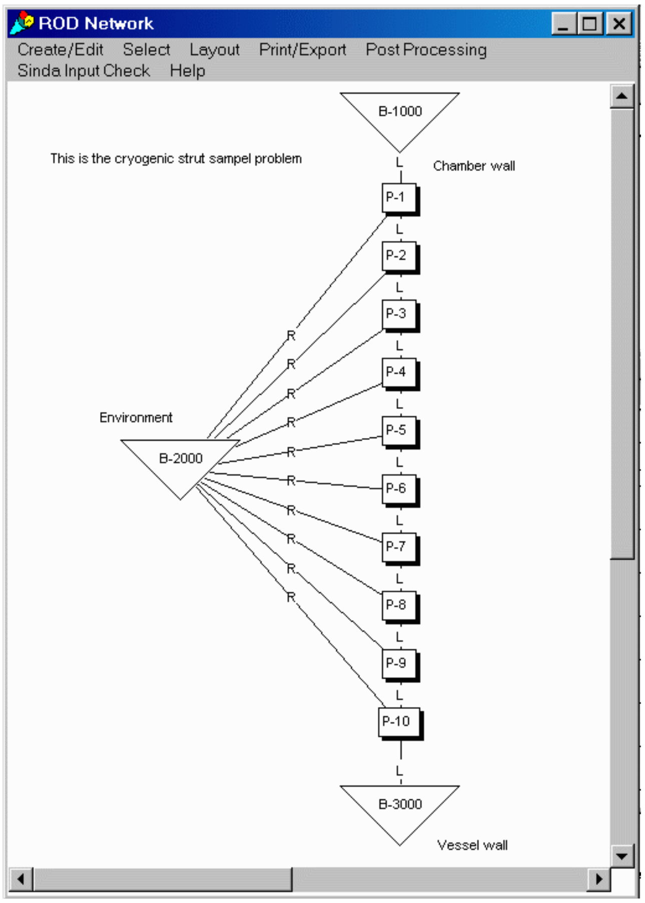
The easiest way to generate the 9 middle linear conductors is via the Create/Edit→Conductor→Add GEN option. After selecting Plain, the selection cursor appears, prompting the user to pick the first pair of nodes to which the GEN will apply. Select first node 1 and then node 2. A form (shown at right) will then appear asking for the initial conductor number (use "2"), number of conductors to generate (use "num-1"), and the conductor and node increments (all should be "1").



Initial Num	2
Num to GEN	num-1
Conductor Incr	1
ROD.1 Incr	1
ROD.2 Incr	1

Note that the selected nodes (prefixed by their submodel name in SINDA-style notation) appear within this form (shown ready to save). Upon saving this form, the traditional plain conductor edit form will appear (as shown on the previous page). The conductance should be set to "kcond*area/(length/num)."

* Note that the first splotch is black, while the second splotch is grey. In some options (such as paths and one-way conductors), it is might important to be able to distinguish between the first selected icon and the second.



The radiation conductors between the vacuum boundary (node 2000) and the rod nodes can be generated with a single Create/Edit→Conductor→Add GEN sequence similar to that demonstrated above. The first node should be node 2000, and the second node should be node 1. In the Conductor Edit form (above), the conductor number should be 201, with "num" (10 conductors) generated at increments of 1. The increment on rod.2000 should be 0 (zero), and the increment on rod.1 should be 1. When the final form appears, the Mode pull-down should be changed to "radiation," and the expression for G should be provided as "emis*surfa/num" (surface area times emissivity). The resulting diagram appears the right.

Conductor labels can be turned on by selecting Layout→Labels→All on. The labels should then be toggled off for clarity.

A **Save Everything** should be performed at this point.

Checking the Model

Input Checks—The basic inputs to the thermal model are conductances, initial temperatures, and capacitances. To make sure that the input values and expressions are correct, we will color the nodes and conductors successively by these values using the options under SINDA Input Check.

First, because of the shape of the network diagram, the color bars that will be generated are preset to be vertical using the SINDA Input Check→Set→Color Bar→Vertical option. To color nodes, choose the option Select→Everything, then select SINDA Input Check→Color nodes by T. A vertical color bar will appear under the mouse cursor, waiting to be dropped onto the screen as were nodes and comments. Like any other icon, color bars may be moved within the diagram window. They may also be resized. (Color bars and their options are described more fully described later under the postprocessing section.)

Once the color bar is placed, the nodes will be colored according to their initial temperature. Analogous operations can be performed to color nodes by capacitance (this operation is not shown in the figures that follow). All node should have a capacitance of about 31.42.

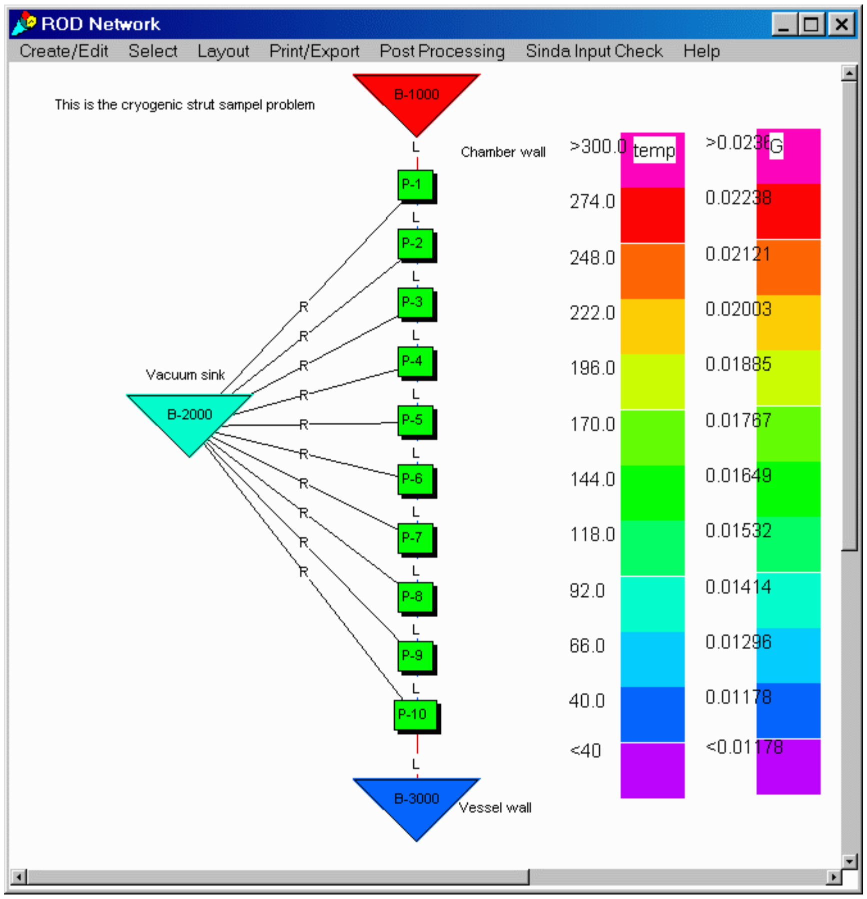
Next, color conductors by conductance using Select→Everything, then SINDA Input Check→Color conductors by G. You must then choose whether to color some or all of the conductors. Radiation and linear conductances should be checked independently since they have different units. All radiation conductances should be 9.425e-4.

The diagram below at the right is a typical result of such input checks, showing nodes shaded by their temperatures, and linear conductors shaded by their conductances. ("Color" can always be reset to grey scale for black and white display devices and printers. Color is used in the pictures in this document, although the appearances may be misleading or unaesthetic if printed or viewed in black and white.)

Before proceeding, the input checks should be cleared using SINDA Input Check→Reset. This will return the diagram to its white state, and remove the color bars.

Preprocessing—A more complete check of the model may be made by clicking on the Model Control tabbed form of the Model Control Panel. Use the Run Sinda/Fluint→Preprocess Only option to initiate the SINDA/FLUINT preprocessor. If everything has been input correctly, this option will return without reporting errors.

(Note that the preprocessor assures that the model is legal and consistent, but it cannot determine whether it is accurate or even useful from an engineering perspective.)



Execution: Running SINDA/FLUINT

We are now ready to run SINDA/FLUINT. There are two methods available, as described next.

Exporting an Input File—If SINDA/FLUINT resides on a different machine than the one currently being used to run Sinaps*Plus*, then the user should create a traditional ASCII card-image input file. This is performed via the Utilities→Sinda/Fluint Input File →To a File option on the Model Control Panel.* A form will appear asking for the name of the input file to create. It is recommended that you suffix ASCII input files names with .sin for clarity. (Unless directory/folder information is provided, this file will be generated in the same location where Sinaps*Plus* was started.) Once generated, this text file may be passed to SINDA/FLUINT in the traditional batch method (refer to the introductions to SINDA or FLUINT). After reviewing the preprocessor and processor output files to make sure that the run was completed successfully, the save file that was created (strut.sav) should then be returned to this (the Sinaps*Plus*) machine via a method that preserves its binary integrity (e.g., disk, binary ftp).

Launching SINDA/FLUINT—If SINDA/FLUINT resides on the same machine as does Sinaps*Plus* along with an appropriate compiler, then SINDA/FLUINT may be launched directly from within Sinaps*Plus* without creating an ASCII input file. Simply select Run Sinda/Fluint→Preprocess and Run→Normal Operation from the Model Control Panel. The SINDA/FLUINT job is now running as a separate background process. (On a PC, a status window will appear while the processor is running. On a Unix machine, the run can be monitored from the shell from which Sinaps*Plus* was launched using stan-

* WARNING: A similar option appears on the submodel control panel, but that option does not create a complete input file. Rather, it creates a subset consisting only of data and logic blocks relevant to the current submodel. This submodel file is intended for review, or for inclusion in full models at sites not employing Sinaps*Plus*.

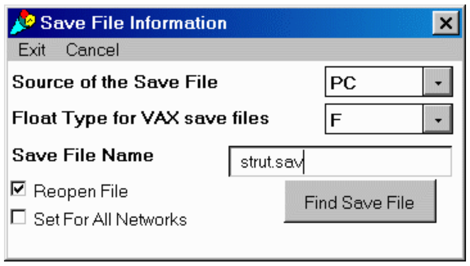
dard Unix commands.) At the completion of the run, a SINDA/FLUINT Run Status window will appear stating the “Successful Completion of the Processor”. If preprocessor or compilations errors are encountered a message will appear on the screen. The processor output file (strut.out) should be reviewed to make sure that the run was successful (error messages, if any, will pop to the screen), the save file (strut.sav) that was created can be used for postprocessing as described next.

Note: it is a common mistake to have omitted the tab before “CALL” in the call to FWDBCK in OPERATIONS, or the call to SAVE in OUTPUT CALLS. If the “C” in CALL is in column 1, then by Fortran rules this line is a comment and ignored.

Postprocessing

This section briefly covers some of the options available for postprocessing. The user should once again have started Sinaps*Plus* (if it was exited) and should be in the STRUT model desktop, with the ROD thermal submodel network diagram window open.

Setting Save File and Record Number—Before postprocessing operations can be initiated in any diagram window, the user must first define the save file to be used. Using the Postprocessing→Save File Info option yields the form shown at the right.



Source of the Save File: PC
Float Type for VAX save files: F
Save File Name: strut.sav
☑ Reopen File
☐ Set For All Networks

If the save file was generated on a different machine, and has not been placed in the same location as Sinaps*Plus*, do so now.* Also, set the Source of the Save File button to the type of machine on which SINDA/FLUINT was *executed*. Select Exit→Save and Exit. If Sinaps*Plus* does not find the save file, it will advise of that failure.

Next, select Postprocessing→Set Starting Time/Record.† A list of times (with corresponding integer record numbers) stored on the save file will appear. In this particular sample problem, two such rows should appear, both with time equal to zero. The first saved snapshot represents the initial condition, and the second represents the results of STEADY (per the rules governing the calls of OUTPUT CALLS for STEADY). Select the second (bottom) row. You will then be prompted if you want this starting time to apply to all the network diagrams. Choose “yes.”

Coloring and Thickening Network—To create a colored representation of the network, the default color bar style is first set via the Postprocessing→Color Scaling→Data for New option. This sets the default style (horizontal vs. vertical, autoscaling vs. specified range, number of colors, and absolute value of results) for future color bars. Because of the long, thin aspect ratio of the diagram, vertical is chosen rather than the default horizontal (shown at right). Actually, color bar sizes and styles can also be edited after their creation as well simply by double-clicking the color bar.

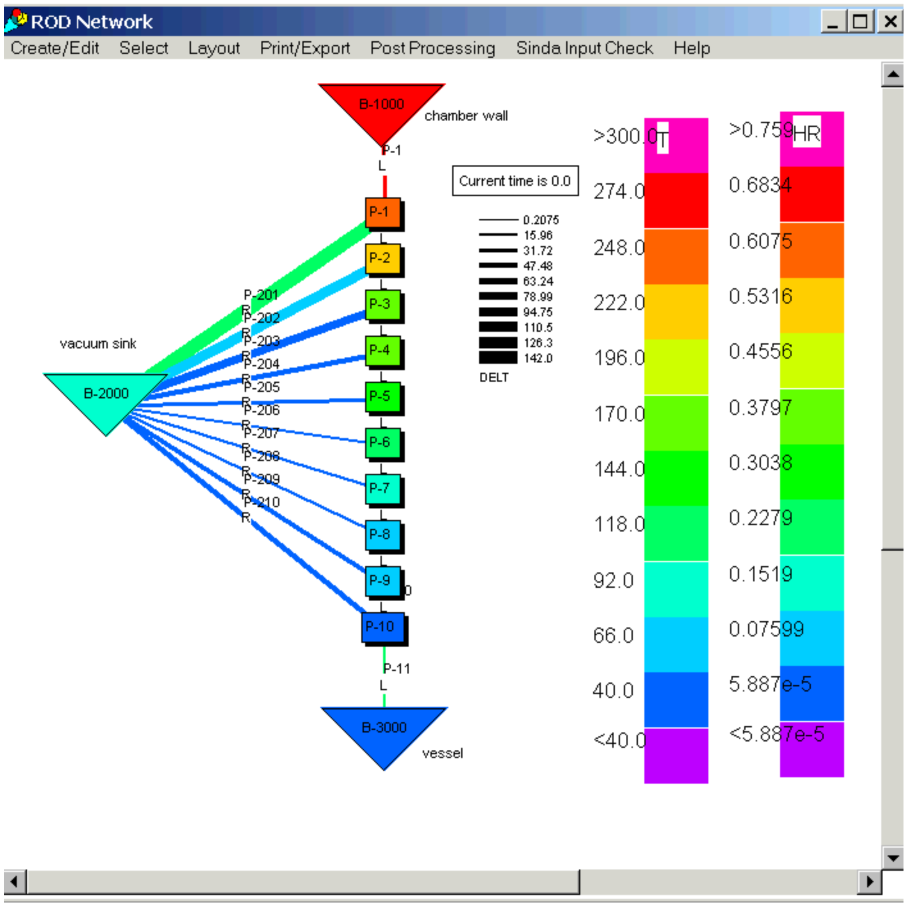
Select all nodes (or Select→Everything), then choose Postprocessing→Color→Nodes.

A form will appear to acquire the value to postprocess: temperatures, capacitances, or heat rates. Choose temperatures. A box will then appear containing the comment “Current time is 0.0.” Place it within the diagram window. The corresponding color bar will appear for placement, after which the nodes will be colored. This process can be repeated for the heat rate through conductors.

Finally, the thickness of conductors can also be simultaneously postprocessed by conductance, heat rate, or delta temperature: Postprocessing→Conductor Thickness →Thicken. Choosing delta temperature,

* The “Find Save File” option, which would otherwise be used to locate the save file, will not be covered in this tutorial.
† This step is actually unnecessary for X-Y and polar plots versus time, which access all records in the save file.

and placing the resulting thickness scale results in a diagram that resembles the one below. Thickness scales may also be customized by double-clicking them.

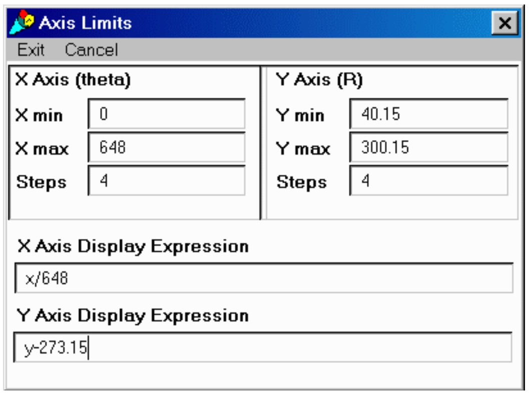


The figure at the right shows that the highest heat flow rate is due to conduction within the rod near the chamber wall. The greatest temperature difference between nodes occurs between the upper part of the rod and the vacuum sink condition.

Plotting by Location—In order to view the gradient along the rod, select all of the diffusion nodes plus the boundary nodes 1000 and 3000. Perform Postprocessing→Plotting→X-Y Plots→by position→nodes, then select "temperature." A new window will appear containing an X-Y plot.

The gradient appears skewed near the boundary nodes (end points) since they were spaced evenly with the diffusion nodes on the schematic, whereas their actual location (from a physical perspective) should be half the displayed distance to the diffusion nodes.

Using the option Axis Control→Set Axis Limits (on the plot window pull-down menu bar), a window will appear that allows the axes to be customized. The range of the axes can be reset, along with the step size and even the units. Unit changes and other such manipulations are performed using the axis display expressions. The expressions shown in the window at the right convert pixels to meters in the x direction (using the information from the legend) and converts temperature units from Kelvin to Centigrade. (Since your diagram will vary, this conversion won't be exactly correct. Select Legend→On/Off/Refresh to find out what the locations of your icons are.)



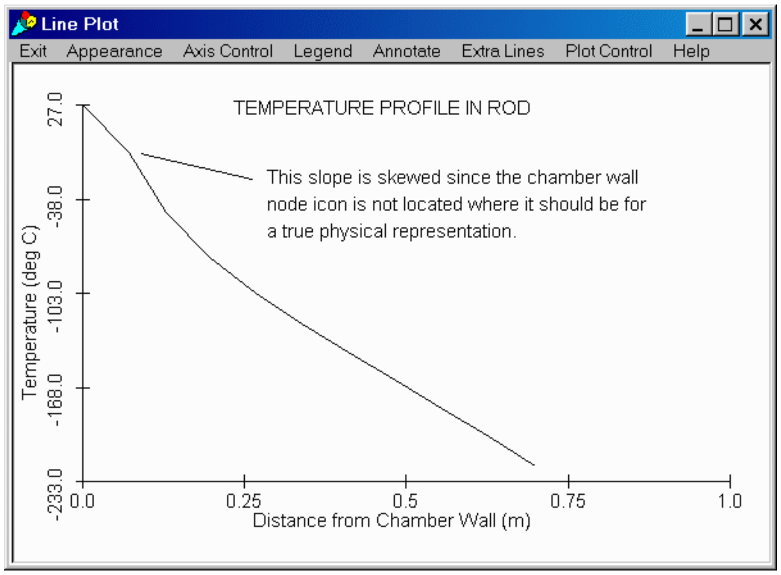
X Axis (theta)		Y Axis (R)	
X min	0	Y min	40.15
X max	648	Y max	300.15
Steps	4	Steps	4

X Axis Display Expression: x/648

Y Axis Display Expression: y-273.15

Together with the addition of some annotations and axis labels, the final plot is shown below at the right. The legend could be edited to reflect the new units, or the axial location instead of a node number, etc.

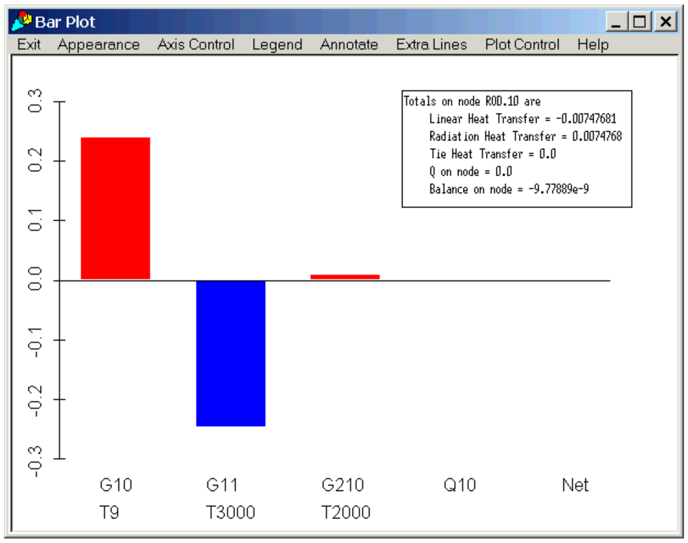
Finally, if a transient or parametric analysis had been performed, and if the user had then plotted many variables within the same X-Y plot (as functions of time or some other value), then the lines can be optionally colored to help make each line distinctive.



This slope is skewed since the chamber wall node icon is not located where it should be for a true physical representation.

Bar Plot: Node Balance—The desired output of the model, the heat leak into the tank, has not yet been found directly although its approximate value can be estimated from the previous color charts. The value of the heat leak through the last conductor (number 11, between nodes 10 and 3000) can be found by X-Y or bar plots made of that specific value. A special type of bar plot is noteworthy: the nodal balance.

Select node 10 and then choose menu option Postprocessing→Plotting→ Bar Plots→Node Balance. A bar chart will appear that will contain detailed information about the heat flowing in and out of node 10 (see right, note the legend is turned on). The imposed heat rate (“Q10”) is zero, and the net accumulation (“Net”) is also zero since this result is for a steady state. The radiation heat gain from the vacuum environment is negligible at this location; most heat flows through the node conductively. The amount of heat flowing into the tank appears to be about 0.24W.



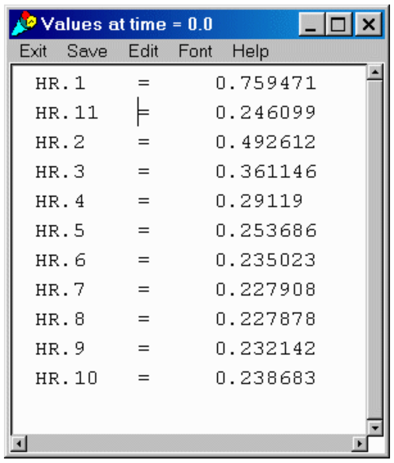
```
Totals on node ROD.10 are
    Linear Heat Transfer = -0.00747681
    Radiation Heat Transfer = 0.0074768
    Tie Heat Transfer = 0.0
    Q on node = 0.0
    Balance on node = -9.77889e-9
```

Text Output—To find the exact value of the heat flowing into the tank, the actual value can be printed by selecting all the linear conductors and by using Postprocessing→Text→Conductors. When heat rate and the current time (vs. all stored points on the save file) are then selected, the resulting text appears as shown at the top left on the next page. (Actually, Sinaps*Plus* does not necessarily print the conductors in the order shown. If not, the data can be rearranged as needed since this window is a fully operational text-edit window.) The value of the heat leak through conductor 11 is 0.246W. A bar plot of this same data (also shown below at the right) shows an important trend: the top part of the rod is cooled by radiation, whereas the bottom part next to the vessel is warmed by radiation from the vacuum environment.

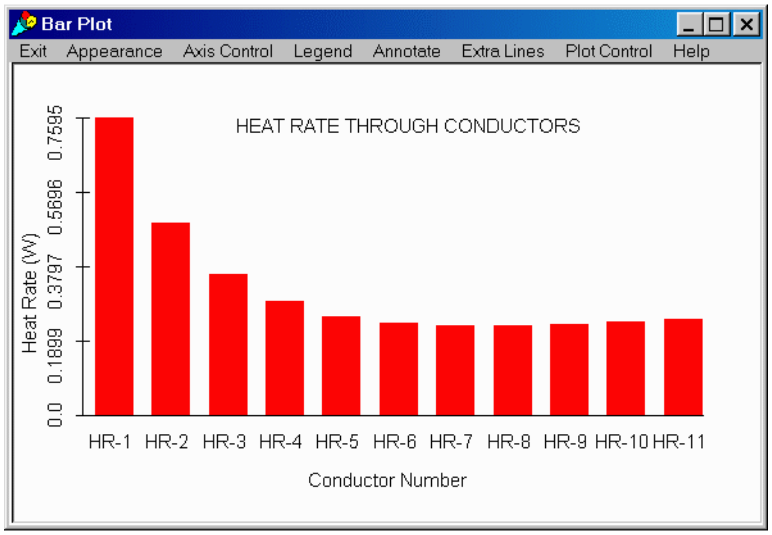
Transient Variation

As a prelude to demonstrating other features, we will first make the above steady-state sample problem into a transient problem.* Extending the previous sample, assume that a 100 watt heater, located on the rod near the vessel, is turned on at time zero. It is desired to know how long such a load can be applied before

* Users should refer to the document Introduction to SINDA for details on this model.

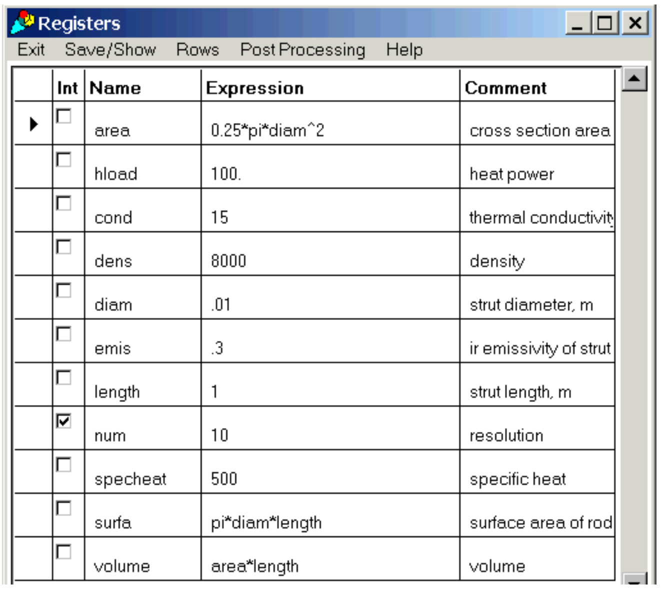


```
HR.1   =   0.759471
HR.11  =   0.246099
HR.2   =   0.492612
HR.3   =   0.361146
HR.4   =   0.29119
HR.5   =   0.253686
HR.6   =   0.235023
HR.7   =   0.227908
HR.8   =   0.227878
HR.9   =   0.232142
HR.10  =   0.238683
```



the heat leak into the vessel exceeds 1W. A constant heat source could be applied to node 9 using Header Source Data selected from the submodel (rod) control panel. However, because we only want the heat load to be applied during the transient event (and not during the steady state initialization that precedes it), we will control the heater by applying the source with an expression as demonstrated later.

We must first define one new register. Click on Registers located in the Model Control Panel. Select Rows→Add Rows and input “1” to add one new row. The new register will be called “hload” and will represent the load placed on the heater, and is therefore set to 100.



Int	Name	Expression	Comment
☐	area	0.25*pi*diam^2	cross section area
☐	hload	100.	heat power
☐	cond	15	thermal conductivity
☐	dens	8000	density
☐	diam	.01	strut diameter, m
☐	emis	.3	ir emissivity of strut
☐	length	1	strut length, m
☑	num	10	resolution
☐	specheat	500	specific heat
☐	surfa	pi*diam*length	surface area of rod
☐	volume	area*length	volume

Next we will declare the heat source into node 9. In the model control panel, click on the tab labeled Thermal Submodels. When the thermal submodel panel appears “Rod” should be highlighted identifying it as the current (and only in this case) submodel. Click on the Source Data button and a blank window will appear labeled “Rod Source Data.” Input an expression as shown at the upper left on the next page. “NSOL” is the current solution routine identifier within SINDA. NSOL is internally set to 0 (FASTIC or “STEADY”) or 1 (STDSTL) for steady state solutions, and 2 (FORWRD) or 3 (FWDBCK or “TRANSIENT”) for transient solutions. Thus, the formula “(NSOL > 1)? hload : 0.0” means that “hload” should be used as the source on this node during transients, otherwise for steady states the source should be zero. An expression “(timen > 0.)? hload : 0.0” could be used equivalently, where “TIMEN” is the current problem time. (Expressions are case insensitive.) Select Exit→Save and Exit to save the data and close the source data window.

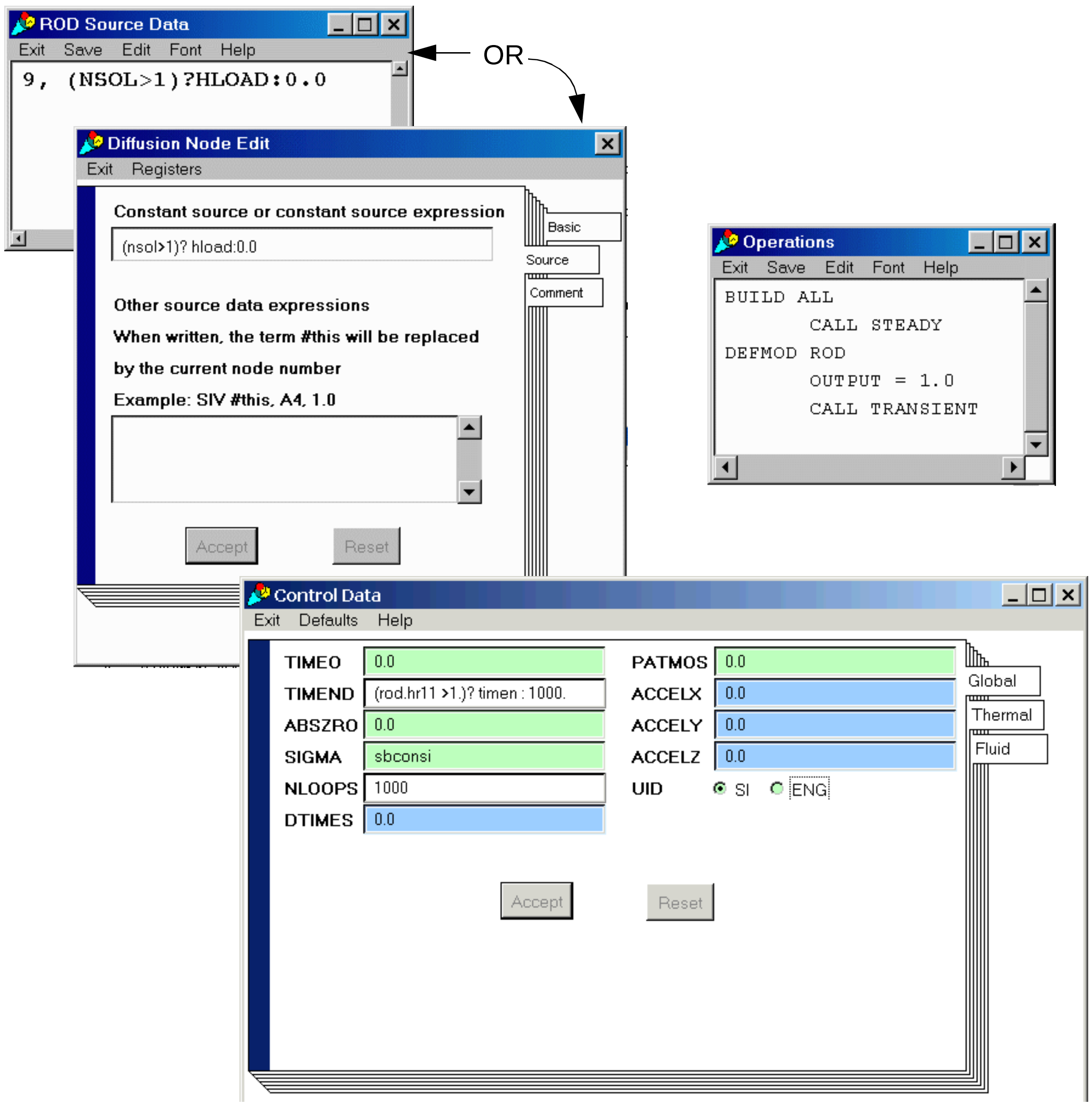
Alternatively, the source data for node 9 may be set in the node edit window using the source tab. From the network window, double click on the icon for node 9 to open the node edit window. Along the right margin of the window will appear a tab labeled “Source.” Click on this tab and the source input window will appear as shown. In the upper box, type in the expression as shown on the left below.

Now we must modify the Operations block to add a call for the transient solution, TRANSIENT. The modified Operations window should appear as shown in the window below. The output interval, OUTPUT, is set to 1.0 before calling the transient. The value of OUTPUT could alternatively be specified in the Global Control Data window under the Thermal tab.

The problem time is the unknown, and determining it is in fact the purpose of the analysis. To stop the run at the desired time, we will modify TIMEND to be the expression (rod.hr11 >1.)? timen : 1000. as shown in the Global Control Data below. This expression states that if heat rate through conductor 11 is not greater than 1 watt, then TIMEND is set to an artificially high value of 1000. Once the heat rate exceeds 1 watt, the value of TIMEND is set to timen (the current time) and the simulation stops.

The logic in the Rod Output Calls block will be invoked every OUTPUT seconds, and therefore the answer will be accurate to the nearest second. Greater accuracy could be obtained by placing the same logic in the VARIABLES 2 block, which is invoked after every time step.

Now is a good time to **Save Everything**.



```
9,  (NSOL>1)?HLOAD:0.0
```

Constant source or constant source expression

(nsol>1)? hload:0.0

Other source data expressions

When written, the term #this will be replaced by the current node number

Example: SIV #this, A4, 1.0

```
BUILD ALL
        CALL STEADY
DEFMOD ROD
        OUTPUT = 1.0
        CALL TRANSIENT
```

TIMEO	0.0	PATMOS	0.0
TIMEND	(rod.hr11 >1.)? timen : 1000.	ACCELX	0.0
ABSZRO	0.0	ACCELY	0.0
SIGMA	sbconsi	ACCELZ	0.0
NLOOPS	1000	UID	[✓ SI] ENG
DTIMES	0.0		

After saving, select Run SINDA/FLUINT→Preprocess and Run→Normal operation from the Model Control Panel to generate the transient results. The program predicts that it will take 253 seconds for the heat leak into the vessel to exceed 1 Watt.

The heat leak (heat rate through conductor 11) can be plotted as a function of time using post processing options in the rod network window.

Final Notes

This sample problem tutorial has introduced the basic operation of Sinaps*Plus*. It did not cover many layout and diagramming options, multiple submodels, advanced postprocessing features, prebuilt models, nor fluid submodels. Many more options and features exist that should be explored for future reference once you feel comfortable with the basic operation.

While all options are fully documented in the User's Manual, experimentation is encouraged. Most of the problems described in the SINDA/FLUINT User's Manual Sample Problem Volume are also available as demonstrations and starting points for experimentation, as are generalized prebuilt models of fins and pipe flow.

More Information

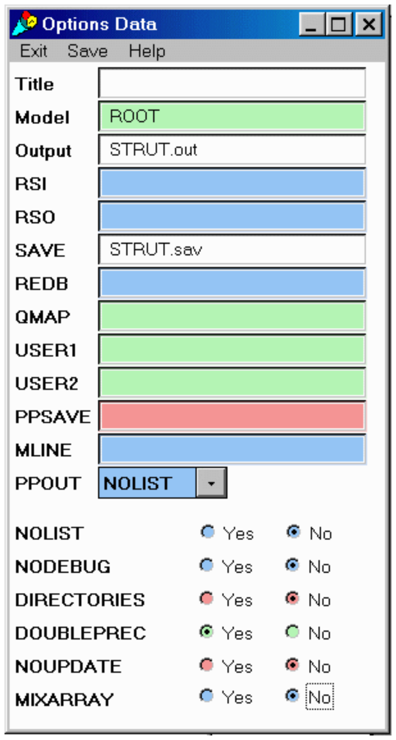
If you have questions about the use or availability of SINDA/FLUINT and Sinaps*Plus*, please contact:

C&R Technologies, Inc.
9 Red Fox Lane
Littleton, Colorado 80127-5701
Phone: 303.971.0292
FAX: 303.971.0035
E-mail: info@crtech.com
Web site: www.crtech.com

This web site contains demonstration versions, on-line hypertext users manuals, training materials, fluid properties, and other announcements.

Appendix A

This section briefly reviews the what was created by using the model building template.



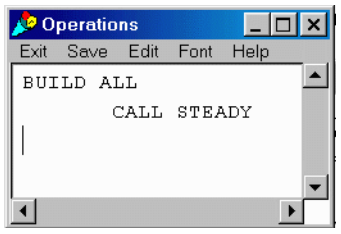
Title	
Model	ROOT
Output	STRUT.out
RSI	
RSO	
SAVE	STRUT.sav
REDB	
QMAP	
USER1	
USER2	
PPSAVE	
MLINE	
PPOUT	NOLIST
NOLIST	Yes [✓ No]
NODEBUG	Yes [✓ No]
DIRECTORIES	Yes [✓ No]
DOUBLEPREC	[✓ Yes] No
NOUPDATE	Yes [✓ No]
MIXARRAY	Yes [✓ No]

Options Data—In the Model Control tab of the model control window a menu appears on the left. This menu provides access to model level input data. Click on the Options Data button to open the Options Data window shown at the left. This form is equivalent to HEADER OPTIONS, naming files to be used by SINDA/FLUINT, along with other preprocessor options. The model name field ("ROOT" by default) is an anachronism with the advent of Sinaps*Plus*. It is used to name the model on ASCII output pages.

The simple model building wizard defined an *output* file and a *save* file. The output file is an ascii text file which will be created during execution. The user specifies what information is to be written to this file. The save file is a binary format file available for postprocessing within Sinaps*Plus*.

You may provide a title ("Cryo Strut Tutorial Problem" or similar) by selecting the field to the right of the word "Title" and typing. Notice that if a large title had been chosen, the field would simple scroll to accept it. Provide also a name for the output file ("strut.out") and a save file ("strut.sav"), until the form resembles the one shown at below. Move the mouse button over to the Exit/Save zone on the pull-down menu bar, and select "Save and Exit."

Operations — OPERATIONS is a model-level logic block. All logic blocks in Sinaps*Plus* are accessed via individual text edit windows. From the Model Control Panel, select Operations. The window at the right will appear.



```
BUILD ALL
            CALL STEADY
```

The simple model building wizard has defined the basic pseudo-fortran logic required for this sample problem. In this type of text window tabs are set for the Fortran convention of column 7 when using that language. Notice that *the HEADER statement itself is not required and should not be provided in such blocks*.

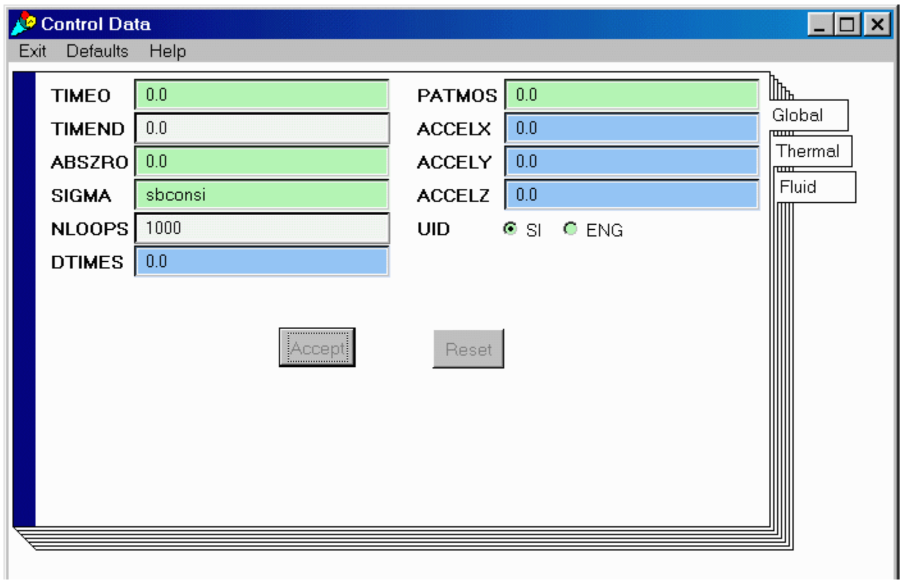
For this sample problem, the only logic is the build statement and the call to STEADY. The wizard created the submodels specified by the user and assumes all submodels are to be built. If necessary, the user can change the BUILD ALL statement to represent a specific build configuration as shown below. Such a modification would be required if more than one submodel had been defined but not all submodels were part of the same configuration. If fluid submodels were defined, a BUILDF statement would also be created by the wizard.

```
BUILD MYMODEL, STRUT
```

If you have time, you should experiment with the options available under "Edit" for future reference. Note that the user also has the ability to print the window contents and to modify font size in the pull down windows of the Operations window. If the font size is modified and the window saved, the new font size will apply to all such text windows.

To close this pop-up window, use your normal window operating system method.

Global Control Data—Open this form via the menu on the *model control panel*. This window should appear as shown below.



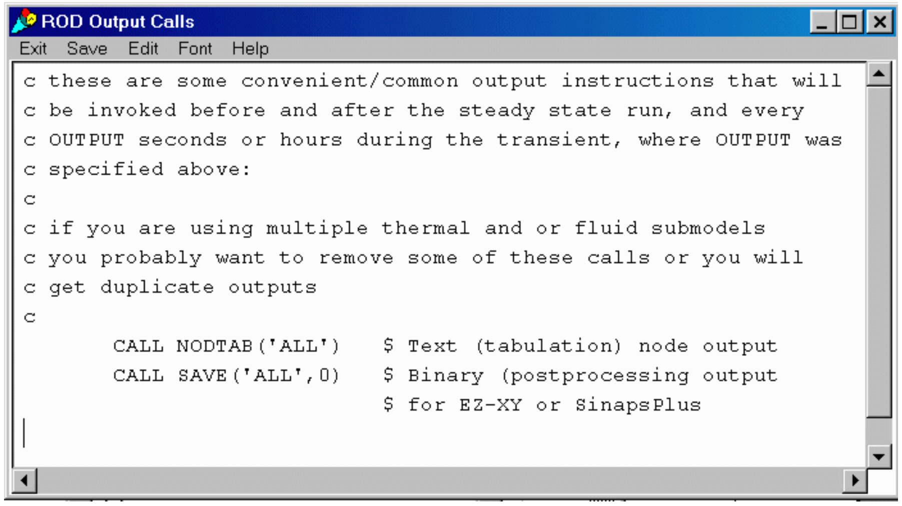
TIMEO	0.0	PATMOS	0.0
TIMEND	0.0	ACCELX	0.0
ABSZRO	0.0	ACCELY	0.0
SIGMA	sbconsi	ACCELZ	0.0
NLOOPS	1000	UID	[✓ SI] ENG
DTIMES	0.0		

The white fields denote options that are almost always used, green fields represent commonly used options, blue represents more arcane options, and red represents items that are rarely used or should be used with caution. Pages are available via the tabs that access the global, thermal, and fluid variables.

The model building template set the values necessary for this sample problem which include ABSZRO (set to 0.0 meaning degrees Kelvin) and SIGMA (set to the built in constant “sbconsi”) and NLOOPS (set to 1000). NLOOPS, the maximum number of iterations that each steady-state call may attempt before either convergence is achieved or the program gives up, is set to 1000. (The push-buttons for UID near the bottom of the global form are only needed if fluid submodels are active.) If you now click on the right tab labeled Thermal, you will notice that the model building wizard has set MATMET = 1 to elect simultaneous methods for the solution routine.

Output Calls—The building wizard creates calls in Output Calls for all submodels created (in this case only one). The Thermal Submodel tab of the *Model Control Panel* provides access to all the header blocks specific to the submodel including Output Calls. Click on the Output Calls button and another text edit window, like the one used to define Operations, will appear (right). Note that calls to NODTAB and SAVE have been created automatically by the wizard.

The NODTAB will provide a tabulation of all node output data. The SAVE routine will create a binary results file named “strut.sav” (as specified in Options Data). This file may then be used to postprocess the model. Note that if multiple submodels are defined within the building wizard, the call to SAVE will be place in the output data block associated with the first submodel defined.



```
c these are some convenient/common output instructions that will
c be invoked before and after the steady state run, and every
c OUTPUT seconds or hours during the transient, where OUTPUT was
c specified above:
c
c if you are using multiple thermal and or fluid submodels
c you probably want to remove some of these calls or you will
c get duplicate outputs
c
        CALL NODTAB('ALL')    $ Text (tabulation) node output
        CALL SAVE('ALL',0)    $ Binary (postprocessing output
                              $ for EZ-XY or SinapsPlus
```