

Introduction to SinapsPlus[®]

SinapsPlus[®] (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 SinapsPlus for fluid modeling. A parallel document concentrating instead on thermal modeling using SinapsPlus 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. SinapsPlus completely replaces this method of communicating models and results via modern graphical pre- and postprocessing.

What is SinapsPlus[®]?

SinapsPlus 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. SinapsPlus then automatically creates the traditional SINDA/FLUINT input file, or directly launches a SINDA/FLUINT run. Once SINDA/FLUINT execution is complete, SinapsPlus 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 SinapsPlus 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.) SinapsPlus can also be used to produce generalized tools called *prebuilts* for use by others.

Organization: Images and Desktops

All SinapsPlus 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 SinapsPlus 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 SinapsPlus from an 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*.

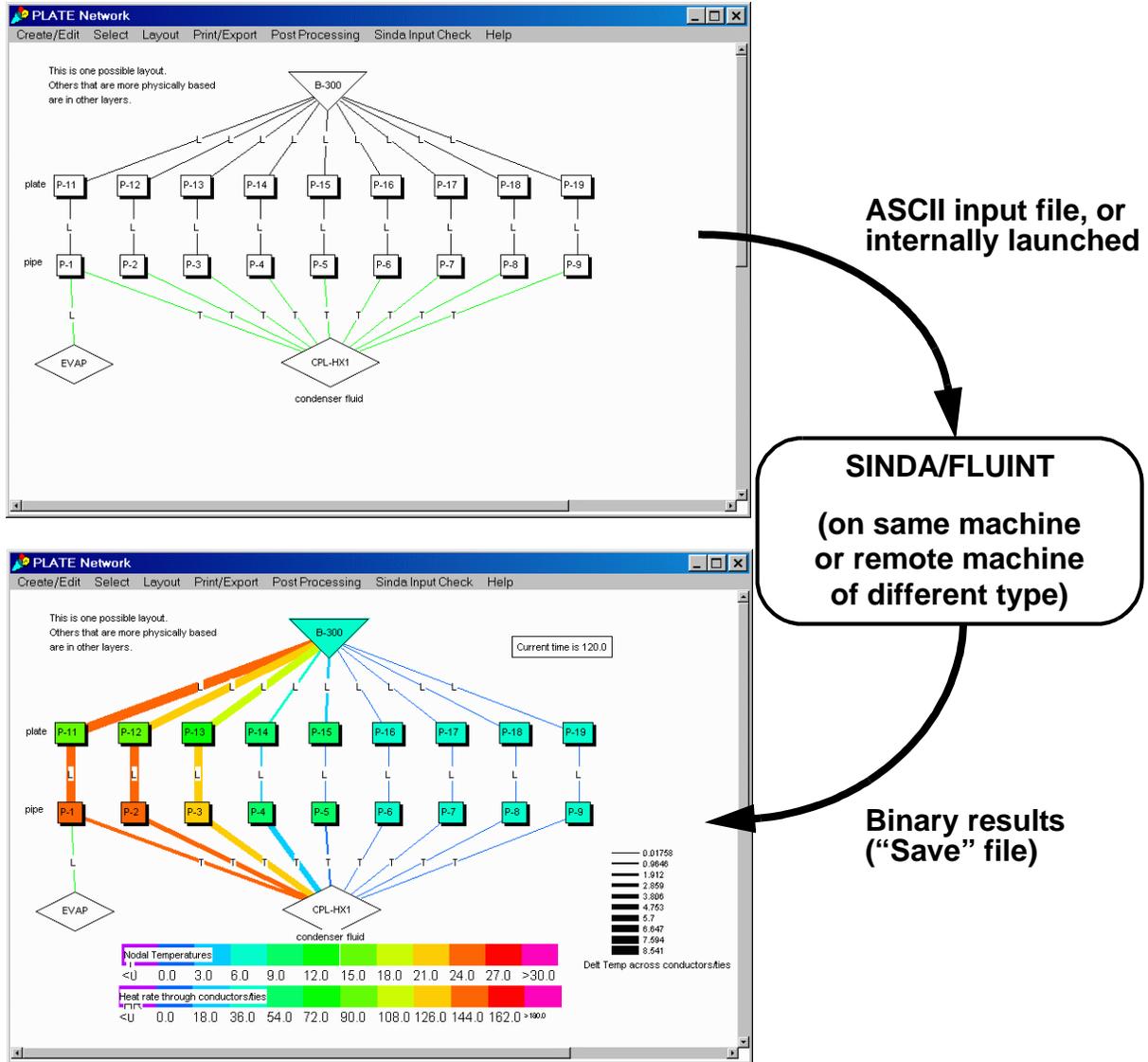


Figure 1: Basic Data Flow

Each model desktop (e.g., Figure 4) contains all the information relating to a complete SINDA/FLUINT model: logic blocks, network diagrams, etc. When SinapsPlus 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 has 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, SinapsPlus 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 SinapsPlus 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 SinapsPlus 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.

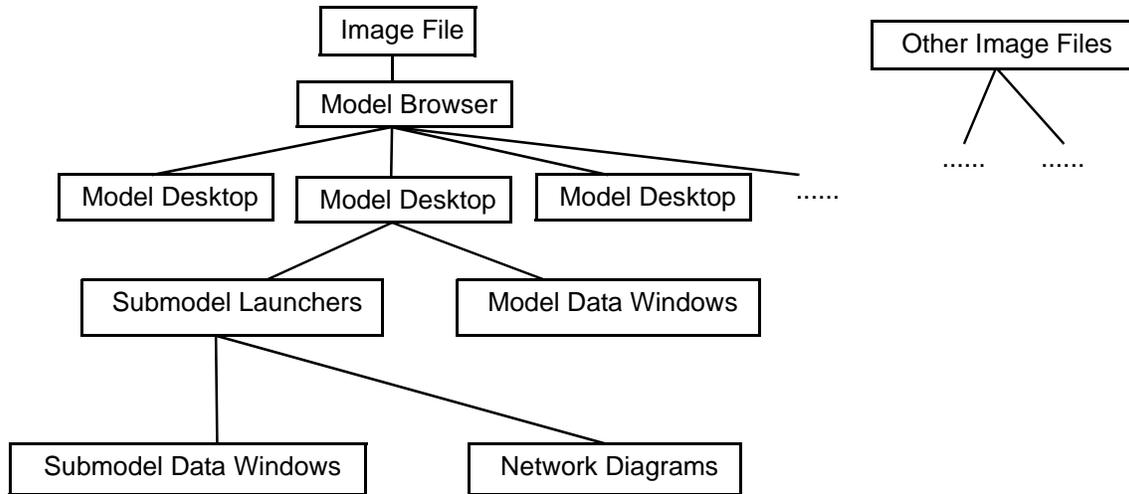


Figure 2: Image File Structure

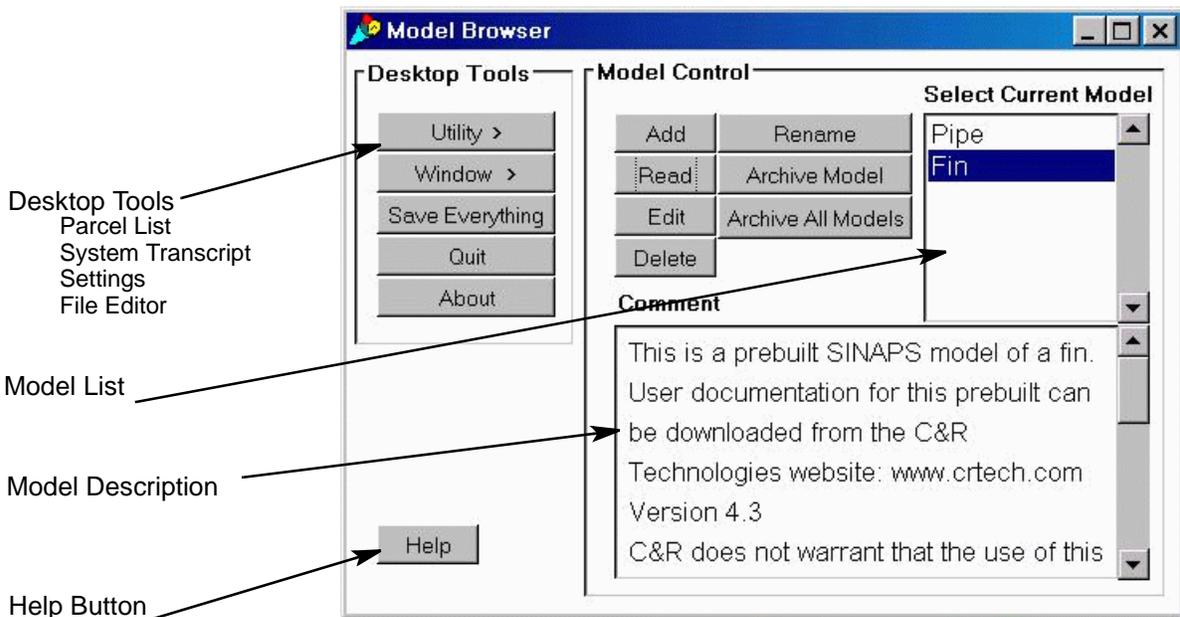


Figure 3: Sample Top-level Desktop (Model Browser)

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 *SinapsPlus* to describe operations within each window.

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

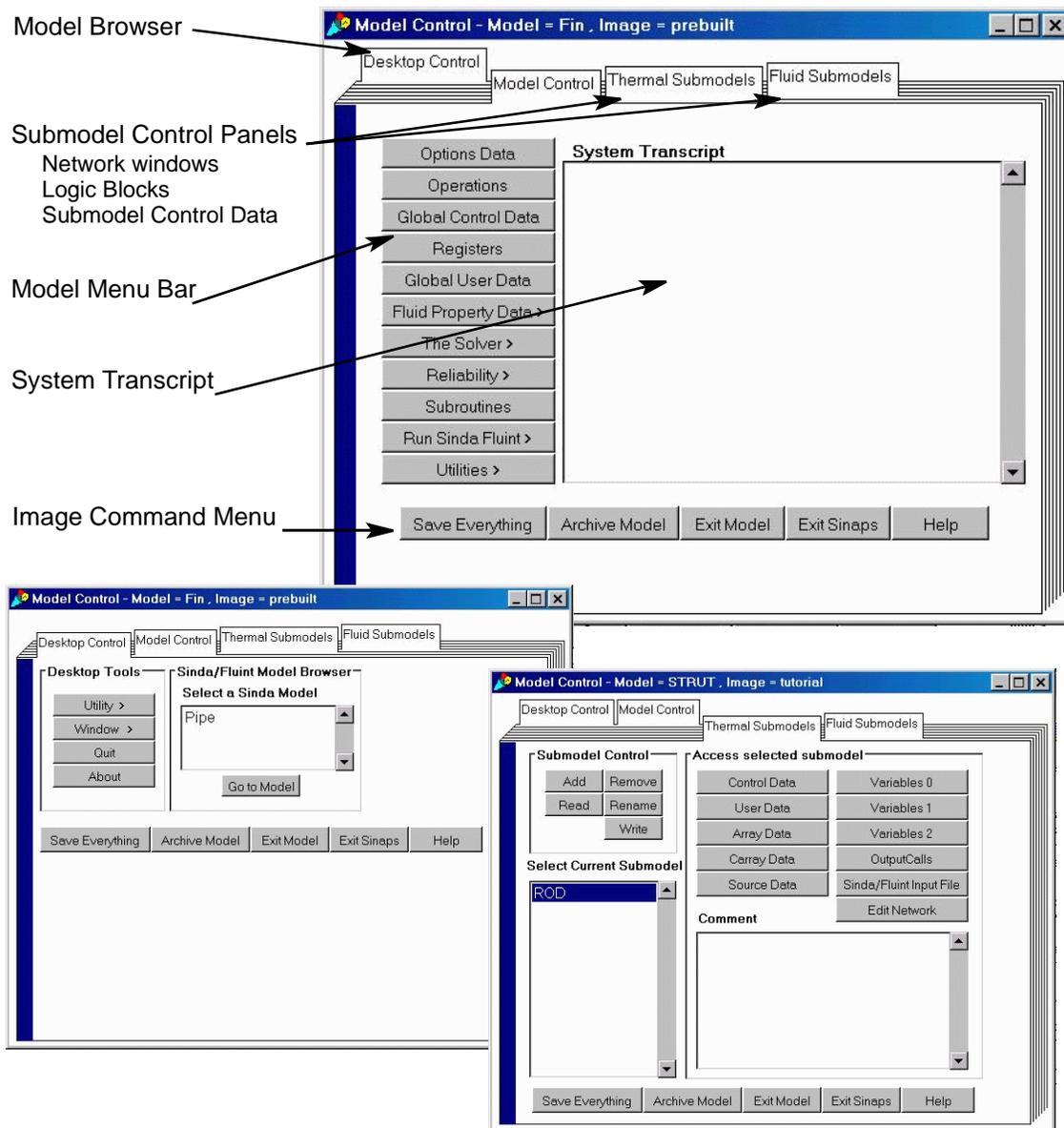


Figure 4: Sample Model-level Desktop (Control Panel)

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 con-

tains 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 be 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 *SinapsPlus*.

Menus, Buttons, and Forms—As mentioned above, most *SinapsPlus* 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).

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, *SinapsPlus* 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)

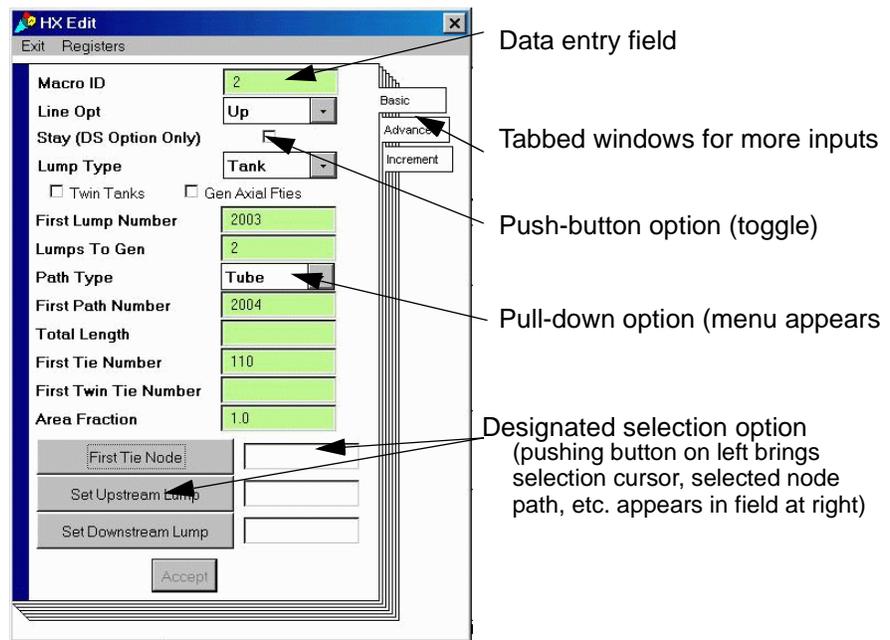


Figure 5: Sample Data Entry Form

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 SinapsPlus 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.*

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

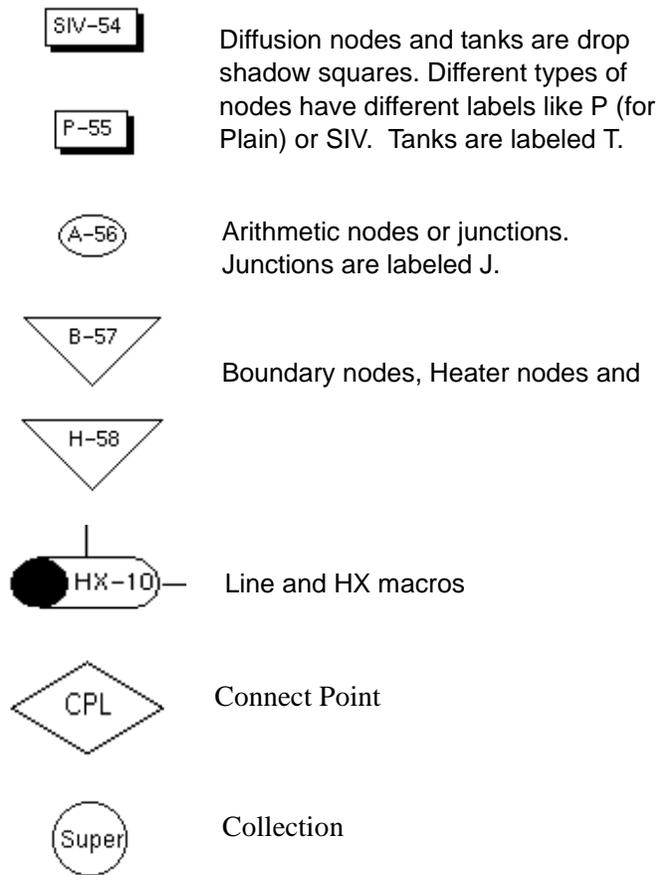


Figure 6: Node and Lump Icon Types

* With the advent of SinapsPlus, the naming of conductors, ties, and paths has become much less important since they can be visualized. For these reasons, the user can let SinapsPlus 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—SinapsPlus 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 SinapsPlus 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 SinapsPlus, 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.,

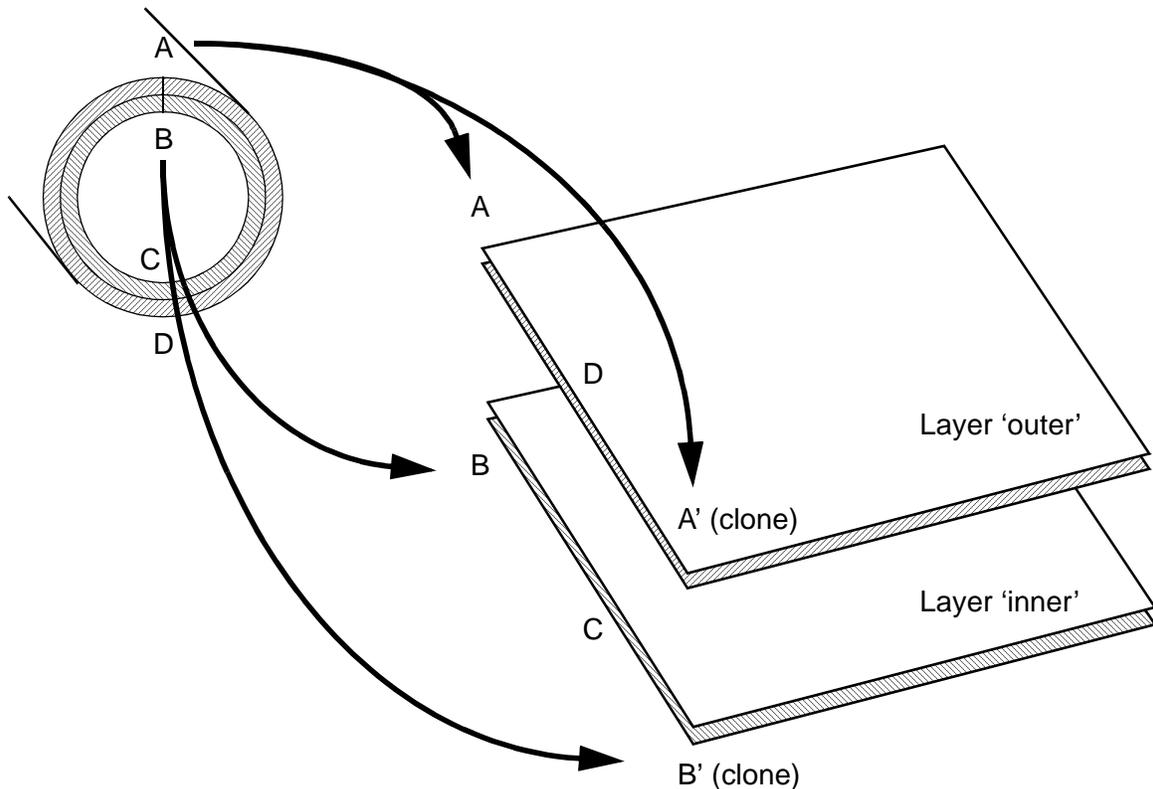


Figure 7: Example of Clones and Layers to Depict a Cylinder in 2-D

“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 be 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 SinapsPlus 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 SinapsPlus 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

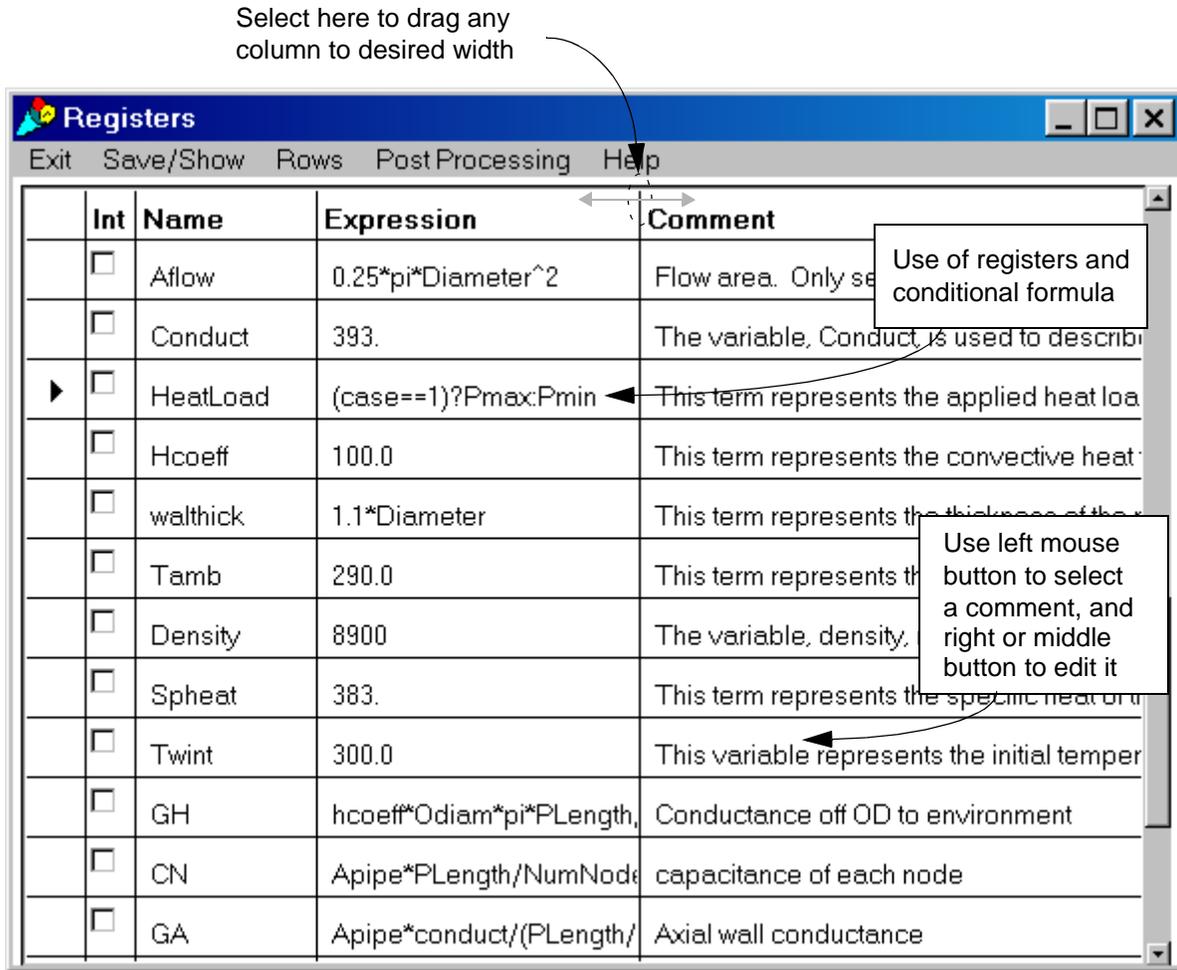


Figure 8: Example of Register Form

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—*SinapsPlus* also possesses the ability to read in existing SINDA/FLUINT ASCII input files. Models may be read into empty model desktops. *SinapsPlus* 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 *SinapsPlus* 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 *SinapsPlus*). Once all such icons have been placed, *SinapsPlus* 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 *SinapsPlus*-unique features, such as networks, can then be employed to make the model more maintain-

able. For these and other reasons, model read-in is intended to be a single event in the lifetime of a model. Once read into *SinapsPlus*, the model should be maintained using *SinapsPlus* 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 *SinapsPlus* 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 *SinapsPlus* 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 *SinapsPlus* 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 *SinapsPlus*.) 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 *SinapsPlus*, a much faster and convenient method exists. The user can preprocess the model and launch its execution directly from *SinapsPlus*. 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 *SinapsPlus* 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, flow rates, 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—SinapsPlus 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 SinapsPlus 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 SinapsPlus?

The goal of SinapsPlus 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. SinapsPlus enables analysts to:

1. **Reduce Errors.** It is much more difficult to create an erroneous or nonsensical model when using SinapsPlus. 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 SinapsPlus, 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. SinapsPlus 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 SinapsPlus 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 SinapsPlus 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. SinapsPlus 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 SinapsPlus form. The recipient will have full access to SinapsPlus 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 SinapsPlus. 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, SinapsPlus 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);

SinapsPlus Sample Problem Description

This section describes a simple FLUINT model that will be developed using *SinapsPlus*. Modeling decisions and other engineering topics are only briefly covered in this document since the focus will be on *SinapsPlus* usage rather than on SINDA/FLUINT usage.

Problem Description

Consider the filling (pressurization) of two cylindrical bottles from a source of air at 22 bar (22.0×10^5 Pascals, or just less than 22 atmospheres) gauge pressure and 20°C . The cylinders are 10cm diameter and 60cm long, are initially full of atmospheric air, and are submerged in a large water bath also at 20°C . The heat transfer coefficient between the air and water (through the bottle walls) is $10 \text{ W/m}^2\text{-K}$.

Both bottles have two fill ports which are connected to 1 meter long by 1 cm inner diameter lines. The ends of these lines can either be connected to the pressure source or to each other. Shut-off valves exist to turn on or off the flow from the pressure source. When open, the valves have a flow area of $1.0 \times 10^{-5} \text{ m}^2$ and a K factor of 25 corresponding to that flow area. Refer to Figure 9 for a schematic of this configuration. The pressure source can be applied at either point A, B, or C in that diagram.

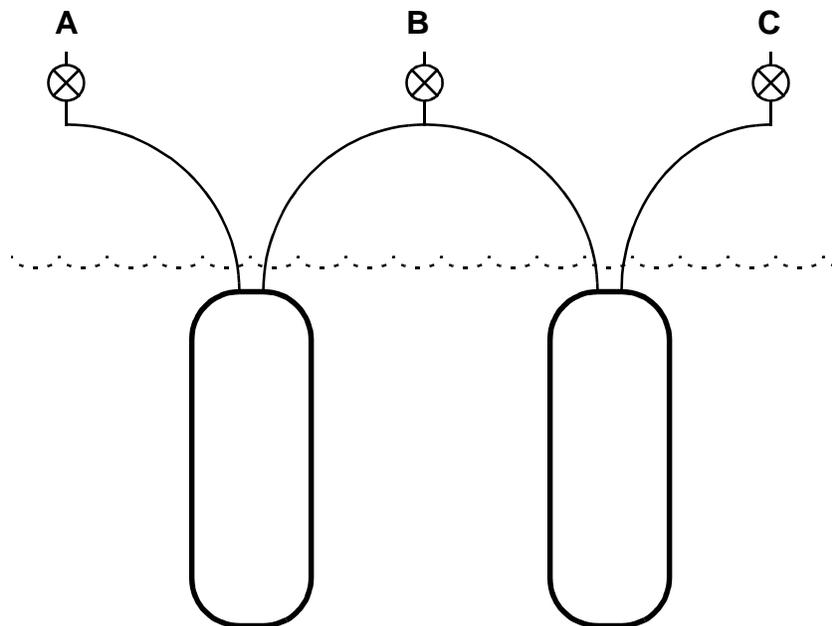


Figure 9: Sample Problem Configuration

* 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 *SinapsPlus*.

Normally, working fluid descriptions will not be developed by the user, since they will more often simply use an existing description. However, in this sample problem air properties must be input. We will use a molecular weight of 28.96, a dynamic viscosity of 19.0e-6 kg/m-s, a conductivity of 0.026 W/m-K, and a specific heat of 1005 J/kg-K.

Assuming that the objective is to pressurize the bottles in a short amount of time, which location (A, B, or C) would result in the most mass being injected into the bottles by the time pressures equilibrate?

Note: This problem will be worked in SI (metric) units. US Customary (“English”) units are also available in SINDA/FLUINT.

Model Description

The problem can be solved entirely by a transient solution, starting from stagnant initial conditions: zero flow rates, and pressures zero (gauge) everywhere.*

Air is the working fluid, designated fluid “8729.” The first “8” signifies a perfect gas, whereas the last three digits are arbitrary, other than 729 happens to be the ASHRAE designation for air. This working fluid will be input in fluid data (FPROP DATA) and then named as the working fluid of the sole fluid submodel, “air.”

Lumps and Paths—The pressure source will be treated as a plenum: an infinite source or sink of mass and energy at a constant pressure and temperature. By assuming that the bottles are well-mixed, a single FLUINT tank will be used to represent each of the bottles. CTLVLV connectors will be used to represent the valves, and STUBE connectors the lines. Junctions are then needed to represent dead-ends in unused valves (say, B and C if A is connected to the pressure source) and to join series connectors. By using junctions for these points, the volume of the lines and valves is itself neglected as small compared to that of the bottles.

Choking might initially occur within the valves. Choking is not checked automatically by FLUINT, and although it is rather easy to include such checks and to model the effects of choked flow at any point in the network, for the sake of simplicity that topic will not be covered in this introductory tutorial.

Nodes and Ties—The water bath is a thermal boundary condition: it will be treated as a SINDA boundary node in submodel “water.” It will be connected to the bottles using HTU (user) ties, whose UA is the given heat transfer coefficient times the surface area of each bottle.

Solution Sequence—In this particular sample problem, a single transient run is needed. Such solutions are requested by calling single routines such as:

```
CALL TRANSIENT
```

But both submodels, the thermal and the fluid, must be activated first using a BUILD and BUILDF command, respectfully. The resulting OPERATIONS block becomes:

```
BUILD SCUBA, WATER  
BUILDF SCUBA, AIR  
CALL TRANSIENT
```

where “SCUBA” is an arbitrary name for the current build configuration.

Since the run time is unknown, the TIMEND (end time) parameter will be set to stop the run when the pressure of a bottle reaches 99.9% of that of the source. An output interval of 1 second is chosen.

* This sample problem is a rare exception: in general it is a bad idea to start a FLUINT transient solution without first performing a steady state (FASTIC) run to start from realistic initial conditions.

SinapsPlus 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.

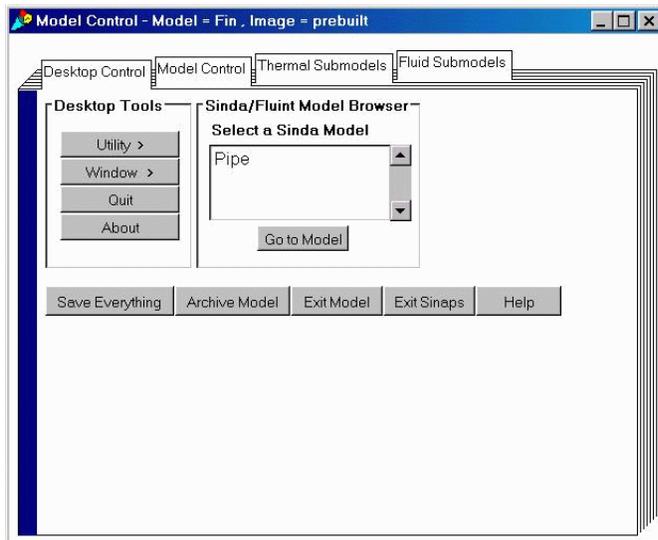
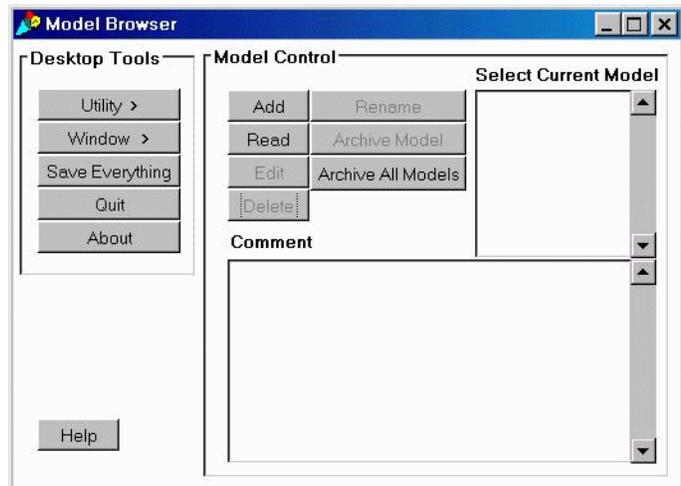
Starting SinapsPlus, and Creating a new SinapsPlus Model Desktop

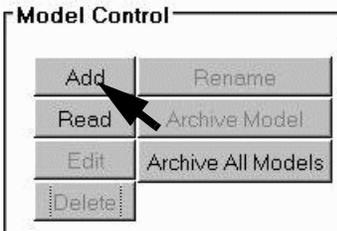
To run SinapsPlus, the user must first enter the windowing system. On a PC SinapsPlus may be started by finding the appropriate SinapsPlus image icon (perhaps under the Start menu) and double clicking it. On Unix machines, the user should type the following sequence in a command shell:

```
sinaps myimage.im &
```

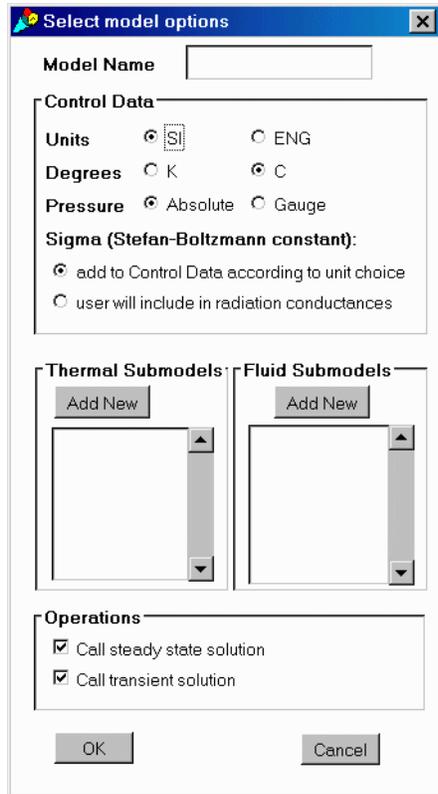
where myimage.im is the name of the desired image file. SinapsPlus will then start, with windows appearing on the screen that are appropriate to the desktop that was visible in that image when it was last saved. This desktop might be the top-level desktop,

or it might be a model desktop. For this tutorial it will be assumed that a new (empty) image has been opened. Upon opening SinapsPlus, 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 a simple model? 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.

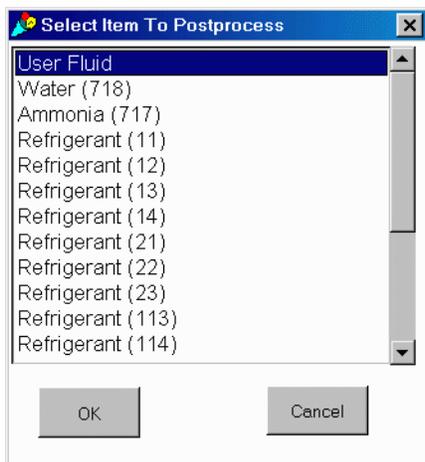


The *select model options* window will appear as shown on the left. First we need to supply a model name. In the input field at the top of the form, type “SCUBA”. Next, we need to define the units for the model. Our sample problem is defined in SI units with temperatures specified in degrees Celsius so we need to click on the radio buttons labeled “SI” and “C”. The problem is specified using gauge pressures, so we also need to click on the “Gauge” radio button. Whenever a set of radio buttons appear in a window, at least one must be checked.

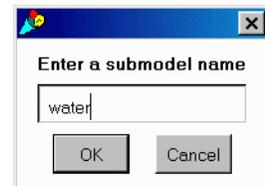
SINDA/FLUINT allows you to define the Stephan-Boltzmann constant (“sigma”) to be applied to all radiation conductors. If you have a set of RADKs output from a radiation analysis package such as RadCAD[®] the conductors may already have sigma included. If this is the case you need to select “user will 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. Only one node exists: a boundary node representing the cooling water. However, it must exist in a thermal submodel, which we will create now. (SINDA/FLUINT and SinapsPlus 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 “WATER” as the submodel name.

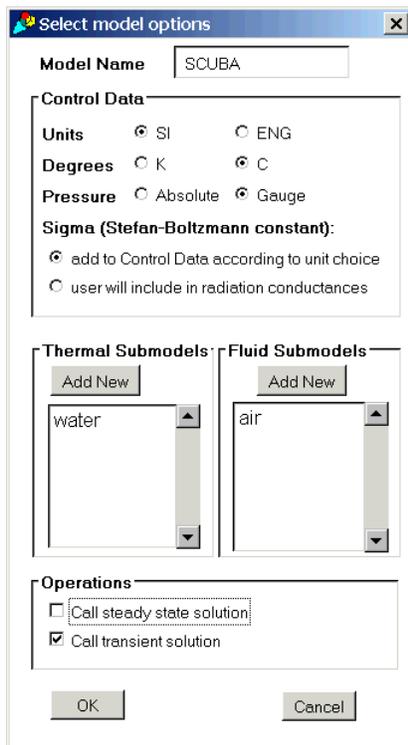
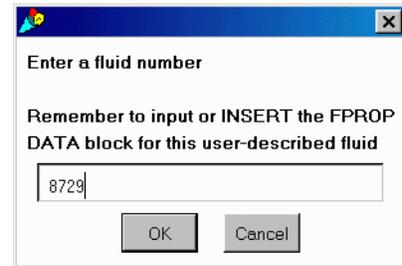


Now the fluid submodel that will contain the network for the bottles, valves, etc. will be created. Select Add New under the Fluid Submodels portions of the window. When the pop-up query form appears, supply the name “AIR” as the submodel name. After clicking “OK”, another window will appear with a list of available working fluids as shown at the left. C&R maintains collections of reusable fluid property files (FPROP DATA) for the convenience of its users for fluids such as cryogenes, propellants, fire retardants, and alternate refrigerants. However, in this tutorial a very simple perfect gas description of air will be input. Click on “User Fluid” to highlight it and then click “OK” to dismiss the window. Another pop-up



query form will appear requesting a fluid ID. Type in the fluid ID “8729”. After completing the simple model wizard we will define the fluid properties. The fluid input window should now appear as shown on the right below. Click “OK” to save the inputs to memory.

For this problem we are only interested in obtaining a transient solution. In the Operations portion of the select model options window, we need to uncheck Call steady state 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.



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, **SCUBA**, in the model list portion of the window. Double click on the model name **SCUBA** in the model browser to return to the *model-level control panel*.

At this point we have created a simply 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 *SinapsPlus*, 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 *SinapsPlus*.

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 *SinapsPlus*. 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.

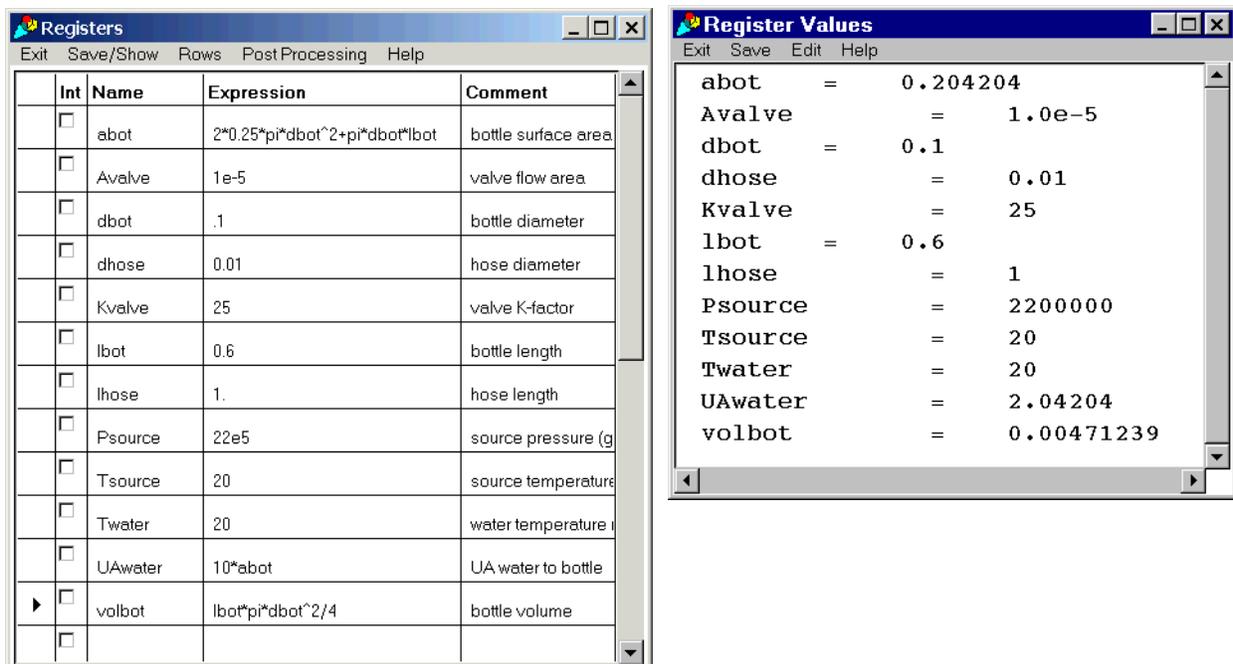
In this problem, key dimensions, performance parameters, initial/boundary conditions, etc. will be input as registers, making them easy to change later. These include the bottle dimensions (diameter ‘dbot’, length

'lbot', surface area 'abot', and volume 'volbot'). Some of these parameters are calculated on the basis of other parameters. For example, the cylindrical bottle volume and surface area are calculated from the bottle length and diameter using built-in values of pi (π).

The register form may be accessed via the **Model Control** tab of the *Model Control Panel* under **Registers**. The form is depicted at the right in the completed state, and after alphabetizing. 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.

To validate your inputs, select the **Save/Show**→**Show Values**. This will create a pop-up window that displays the current values of the registers. The window is shown at the right below. Before proceeding, make sure your registers are producing the correct values.



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

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

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 SinapsPlus 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.

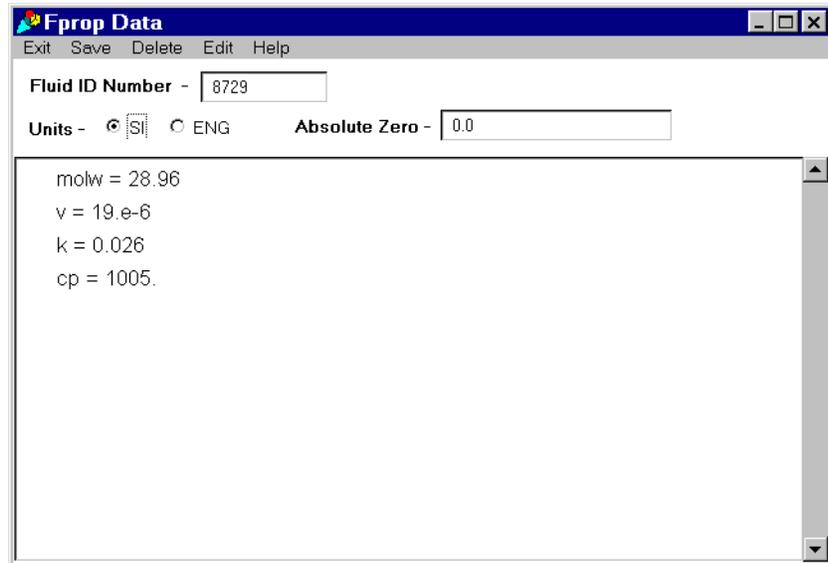
The *model control panel* should appear similar to that shown in the lower left of Figure 4. Note the title of the window should read as “Model Control - Model = SCUBA, Image = myimage.”

* A full list of built-in constants and functions may be found in the SinapsPlus User's Manual.

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

Fluid Properties Input—Normally, users would include or insert a reusable FPROP DATA (fluid properties) block if no built-in library fluid is appropriate. For the sample problem we will directly input the fluid properties.

To define the fluid properties for air click on the Model Control tab. Select the Fluid Properties Data→Input/Edit option on the left menu. This leads to a blank form into which fluid properties for one working fluid can be input. The bottom half of the form is a text input area. In the top half, specify “8729” as the Fluid ID Number, press the SI radio button and input Absolute Zero of 0.0. This defines the local unit set for this fluid description such that if it is reused in another model unit conversions are handled automatically.



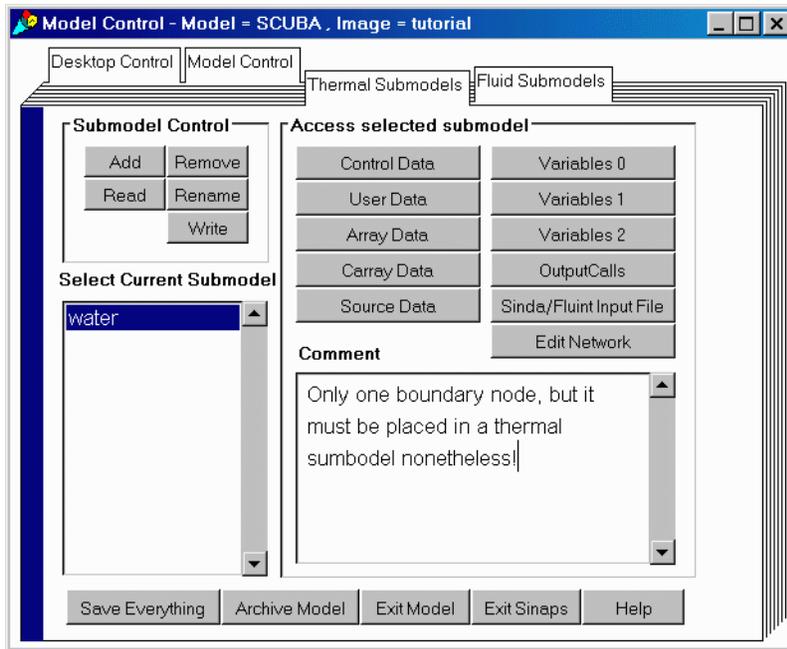
In the lower portion of the screen type the following *being careful not to put text in column 1*, which has special meaning in FPROP blocks:

```

molw = 28.96
v = 19.e-6
k = 0.026
cp = 1005.

```

The final block should appear as shown above. Press Alt-X or choose Exit→Save and Exit.



Submodel Control Panel—

Each submodel in SINDA/FLU-INT 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 the left. (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

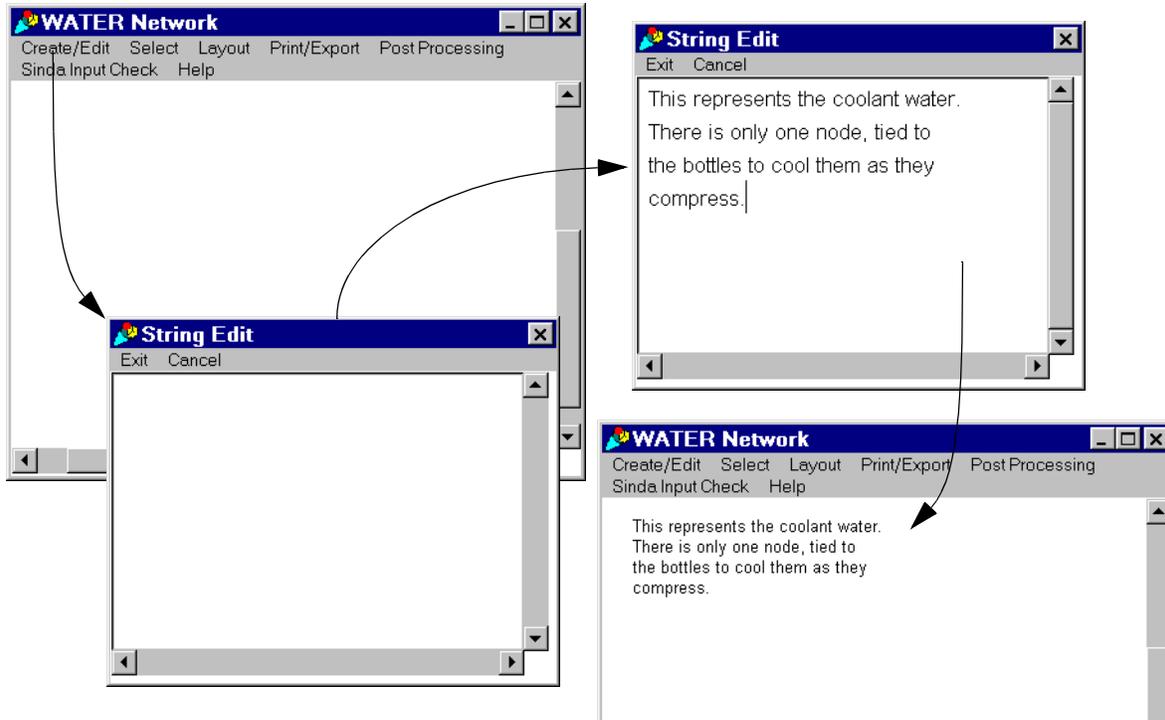
type in the comment field.) This single control panel provides access to all thermal submodels defined within the model. A similar fluid submodel control panel is access by clicking on the tab labeled *Fluid Submodels*.

The submodel control panels similar to the *top-level model browser* in that they contain 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/fluid 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—Only one node exists: a boundary node representing the cooling water. However, it must exist in a thermal submodel, which we will create now. (SINDA/FLUINT and SinapsPlus are designed to handle many more submodels. Most models will usually have at least two or three submodels.) Click on the submodel name “water” to highlight it in the model list of the *Submodel Control Panel*. With the model name highlighted click on the *Edit Network* button to open a network diagram. 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. A blank screen appears as shown below at the upper left. It is important that this window not be too large to make subsequent operations clear: resize it (drag the lower left hand corner) until it is about the same aspect ratio as that shown (no larger than about a quarter of your screen). 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 below at the lower left.

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



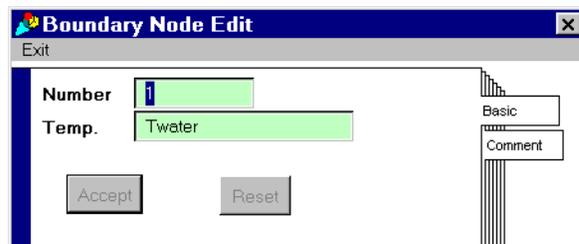
Once you are satisfied with the comment, click **Exit**→**Save and Exit** from the pull down menu. The comment will appear as you define it when the mouse cursor is within the network diagram window. It will follow the mouse until you press the mouse button, at which location the comment will be “dropped”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

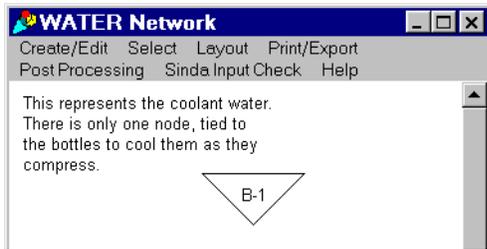
If you aren’t satisfied with the comment, and wish to edit it, select it with the mouse button. A red square will appear in the upper left hand corner to confirm the selection. Then, choose **Edit** from underneath the **Create/Edit**→**Comment** pull-down. The original comment edit window will appear to accept any changes. Or, as a short-cut, simply 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** pull-down 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).

Adding a Node—We are now ready to add a nodes to the diagram. To add an individual boundary 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 1, and the temperature *T*water. 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.





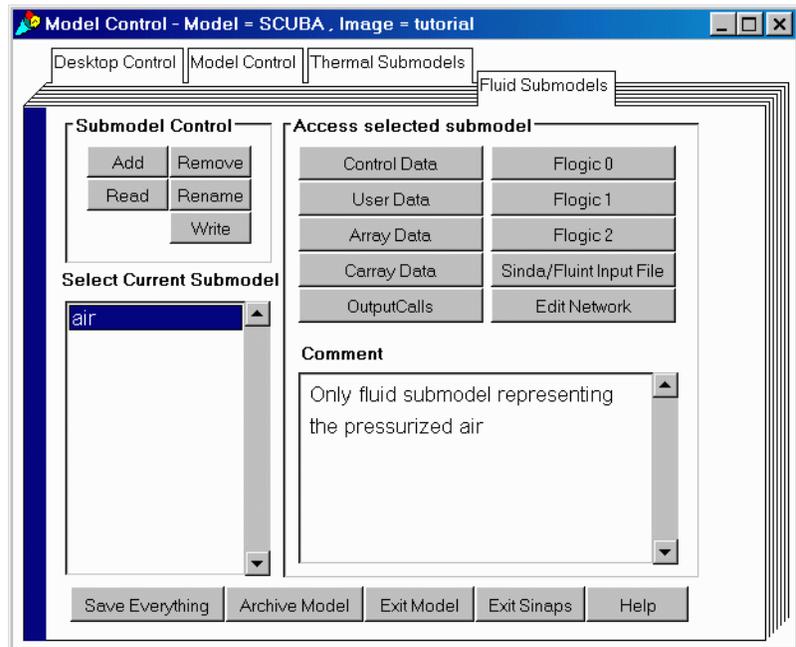
When Exit→Quit Window and Create is chosen from the pull-down menu (or simply click on the “x” in the upper right of the node edit form), the form will disappear and a triangular icon with a “B-1” in the middle will appear. Drop this icon near the top of the screen as shown at the left. Ties will later be made to this node from the fluid submodel that will be created next.

Do not dismiss this network window. Instead, collapse it to the task bar (or an icon for unix systems) for now. If it were dismissed, press **Edit Network** on the water submodel control panel to reopen it.

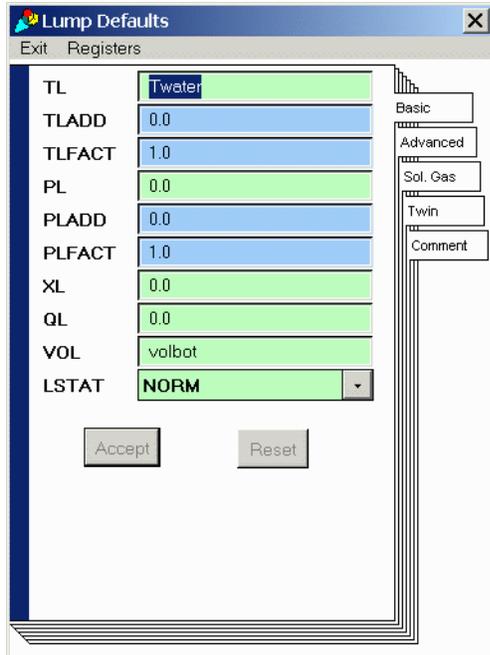
A **Save Everything** should be performed at this point.

Fluid Network Diagram—

Now the model of the bottles, valves, etc. will be created. Select the Fluid Submodels tab the Model Control Panel. Remember that we created the fluid submodel “AIR” with the simple model building template and it now appears in the submodel list in the browser window of the fluid submodel control panel as shown at the right. Note that a comment has been added to describe the submodel.



The network diagram, which can be opened by selecting **Edit Network** from the Fluid Submodel Control Panel, contains the equivalent information of the traditional HEADER FLOW DATA block. Most importantly, it also contains a diagram of the network. Like the thermal diagram, the top-level menu bar contains many options and leads to many more suboptions, only a few of which will be covered here.

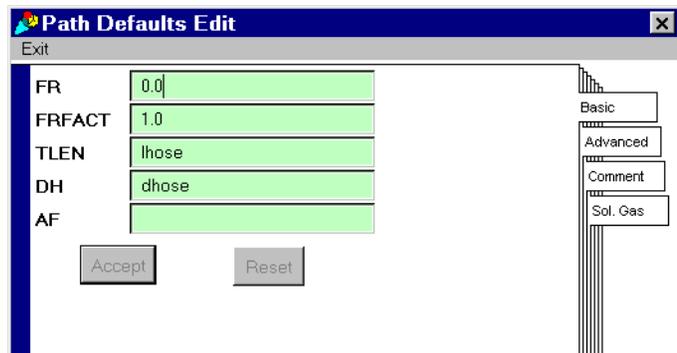


Normally a good first step is to tell the program which working fluid will be used in this submodel. In this case the simple model builder has done this so we will simply verify the input. Choose **Create/Edit→Header→Set Fluid ID**. The ID of “8729” should appear in the window. Press “Cancel” to dismiss the window.

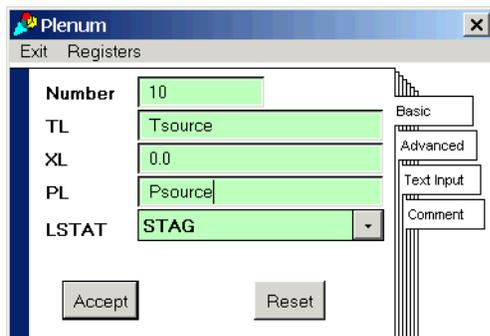
Another useful starting point is to define the default values of pressures, temperatures, flow rates, etc. to avoid repetitive inputs later while the network is being defined. (This would be a good time to go back and review the registers that were set up.) Select **Choose Create/Edit→Lumps→Edit default**. Specify Twater in the TL field (being the initial temperature of the bottles), and Vol-Bot in the VOL field (being the volume of the bottles). The initial pressure is already correct at zero (gauge). Normally, the quality XL could also be changed as needed, but since the fluid ID starts with “8” it is intrinsically a gas and therefore quality is understood to be always unity. Accept the form when it appears as shown at the left, and then press Alt-Q or Exit→Quit Window.

Next we will set path defaults. Select **Create/Edit→Paths→Edit Default**. The initial flow rate (FR) of zero is fine as it is, but change the default length (TLEN) and hydraulic diameter (DH) to be lhose and dhose, respectively. This will aid in the subsequent input of the STUBE connectors, which model the lines.

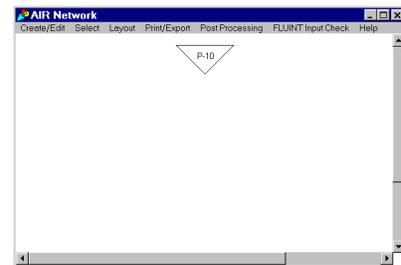
Accept the form when it appears as shown at the right, and then press Alt-Q.

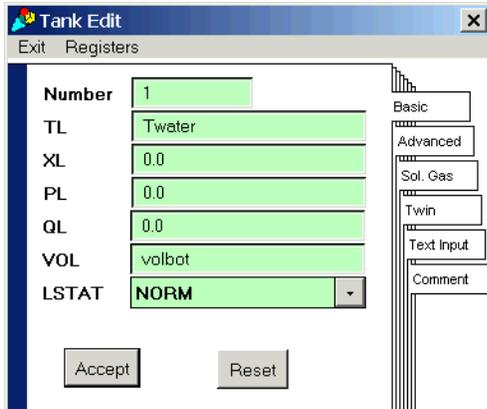


Next we will create a plenum representing the pressure source. Select **Create/Edit→Lumps→Add Single→Plenum**. In the form that appears, specify “10” as the Number, “Tsource” as the temperature TL, and “Psource” as the pressure PL. The value for “LSTAT” needs to be changed to “STAG”, representing the stagnation pressure of a large air source. When the form appears as depicted below at the left, accept it and press Alt-X. A triangular icon will appear on the screen to be placed. Locate it in the upper middle of the diagram as shown below at the right.



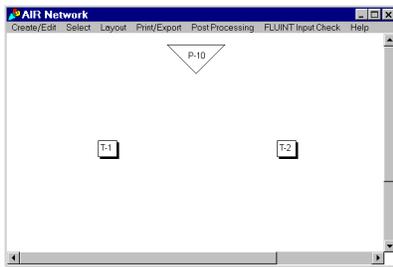
The tanks representing the bottles will be added next. While multiple tanks could be generated using a single form, in this case it is convenient to enter the “multiple” mode where identical (or at least similar) tanks will continue to be generated until the user stops.



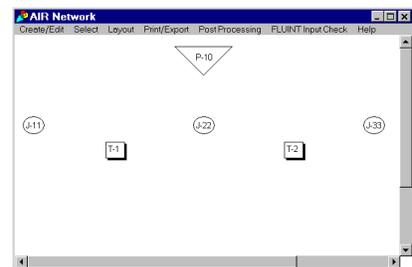


Select Create/Edit→Lumps→Add Multiple→Tank. Use “1” as the Number of the first tank. Notice the temperature, pressure, and volume are already correct, having been defined as default values. Accept the form when it appears as shown at the left. Under the Exit pull-down on this form, note that you have two options: Quit Window and Create (Alt-X) and Quit Window No Create (Alt-X). Choose the first option (or press Alt-X) to create Tank #1, and place it on the diagram. The same form will then reappear. Input “2” as the Number of this second tank (if it does not already appear as “2”), and again choose the first option (or press Alt-X). Tank 2 will then appear to be placed on the diagram, and again the form will reappear. This time, choose the sec-

ond option or press Alt-Q to terminate this tank generation mode. The diagram should appear roughly as shown below at the left.

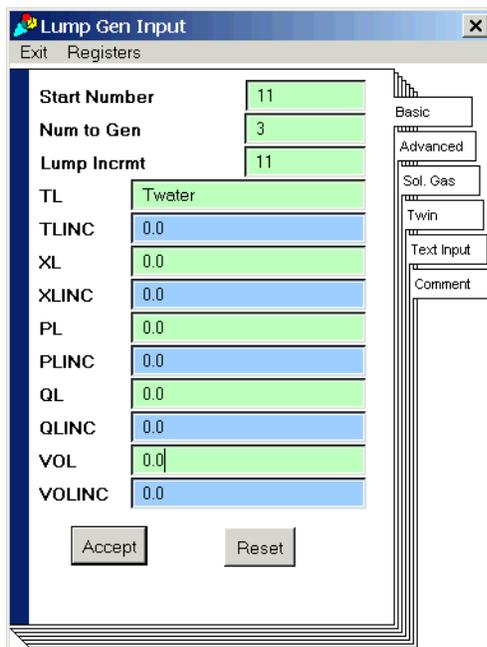


Three junctions, numbered 11, 22, and 33 will be created to represent either dead ends of lines, or places where the pressurization valve can be attached.



To create the junctions, select Create/Edit→Lumps→ Add

GEN→Junction. Use “11” as the Number of the first junction, “3” as the number to generate, and “11” as the increment. Replace the entry “volbot” for the VOL with 0.0. The form should appear as shown below at the left. Accept and press Alt-X, and then place the junctions that appear such that the network has the appearance of the one shown above at the right.



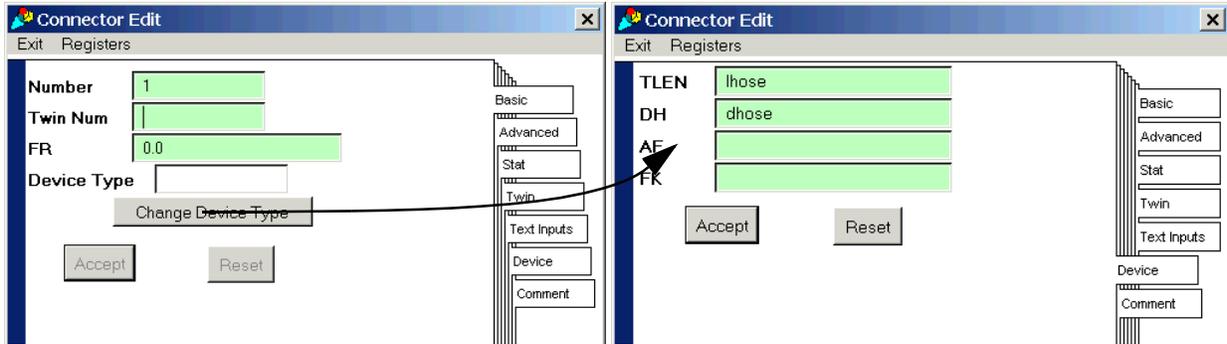
A **Save Everything** should be performed at this point.

STUBE connectors (numbered 1 through 4) will now be created between the junctions and tanks. The diameter of all such lines will be hose, and the length hose, which have already been defined as path defaults.

The direction of positive flow rate will be selected by the order in which the tanks and junctions are selected, but as long as a negative flow rate does not cause any concern, then no concern need be given to this order of selection. In fact, since the flow will enter one junction in one run, and then another junction in another run, reverse flow in at least one path is inevitable.

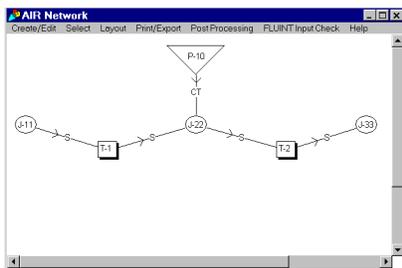
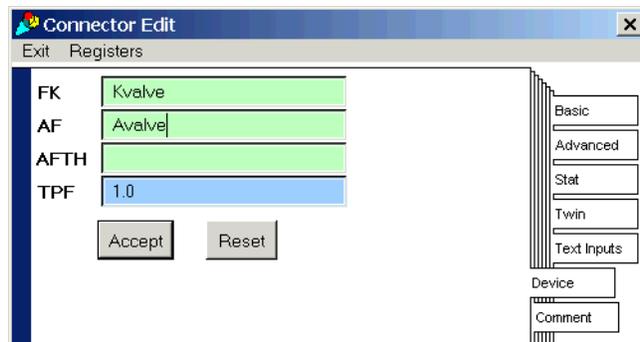
The paths will therefore extend from junction 11 to tank 1, from tank 1 to junction 22, etc. (from left to right in the above network diagram). Since the numbering scheme is not amenable to a GEN operation, the multiple add mode will be used instead. Select Create/Edit→Paths→ Add Multiple. A box will appear asking you to select either tubes or connectors. An STUBE is a type of connector, so pick that choice. A dialog box as shown below at the left will appear. Input “1” as the Number of the first path

(if it does not already appear as “1”), accept, then press **Change Device Type**, then choose **Stube** in the pop-up list. A form such as the one below at the right will appear. Since the default data is correct, accept it and press **Alt-X**. You will now be prompted to place the first path by the cross-hair cursor. Note that text has appeared in the *System Transcript window* prompting you to “Pick the upstream lump or macro”. Pick junction 11, then tank 1. The first path should appear, and then the **Connector Edit** form will reappear.



Press **Alt-X** and place the second identical **STUBE** connector between tank 1 and junction 22. Continue with a third **STUBE** between junction 22 and tank 2, and a fourth between tank 2 and junction 33. Press **Alt-Q** to stop the multiple add mode.

Create a **CTLVLV** (control valve) connector by a similar process. Choose **Create/Edit**→**Paths**→**Add Single**, then choose connector. This time, after pressing **Change Device Type**, then choose **Ctrlvlv** in the pop-up list, and supply **Kvalve** as the **FK** value, and **Avalve** as the **AF** value, as shown at the right. Accept the data, press **Alt-X** to generate the connector between the plenum and junction 22. The network should appear as shown below at the left.



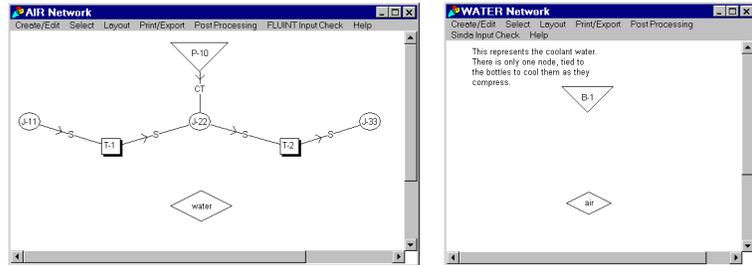
A **Save Everything** should be performed at this point.

Given the construction of this network, the objective of the problem translates into finding out to which junction the **CTLVLV** connector should attach, where junction 11 represents point A in Figure 9, junction 22 represents point B, and junction 33 represents point C. Due to symmetry, there will be no difference between point A and C, so only two cases need to be evaluated. The network, however, appears dedicated to be investigating only one such possibility: the connection to point B. (This is a good first case as a model check, since we would expect both tanks to respond identically in this configuration. If they do not, a network construction mistake has been made.)

How can this configuration be compared to attachment of the pressurization source at point A or C? One option is to create two or more **CTLVLV** connectors to other junctions, and then turn all but one off. (Setting the **FK** factor negative on a **CTLVLV** connector shuts off the flow, which is why this particular connector was chosen since it is otherwise identical to a **LOSS** connector.) Turning on and off valves can also be done within a single run, such that both cases are performed in a single run. However, in this introductory problem we will choose a very simple option: perform a run, move the **CTLVLV** connector to a new junction, and then perform another run.

Finally, heat transfer ties must be added between the tanks and the boundary node in submodel “water.” During these manipulations both network diagram windows to be open and visible, so take a little time to open both windows if they are not already open. (You may have to find the Edit Network option in both submodel control panels.) Shrink these windows down in necessary so that both fit on your screen, and position them such that both are visible at once: perhaps one in the upper left hand corner and the other in the lower right hand corner.*

To make a connection (tie, conductor) between one network diagram window and other, a *connect point* is required. Ties are FLUINT entities, and must be created in the fluid network diagram. From the menu in the fluid network window, select **Create/Edit→Links→Connect**

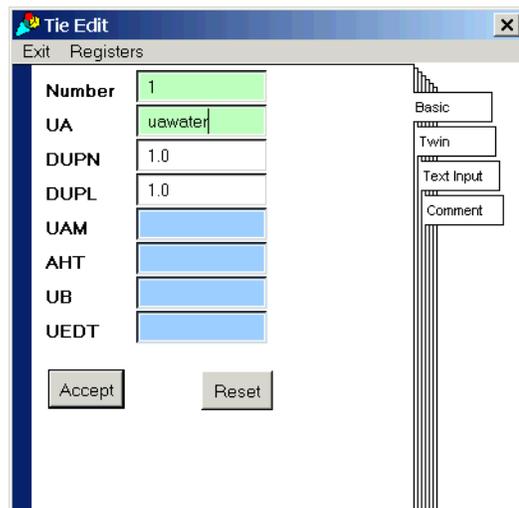


Points→Add. A selector window

will appear asking to which thermal submodel the connect point should link. There is only one choice (water), which is already highlighted, so press OK. A diamond icon with the name “water” will appear to be placed. Locate it underneath junction 22 in the center of the fluid diagram. This is a link to the thermal diagram, and a link back to the fluid diagram has also been created (and placed in the corresponding location in the thermal diagram). If this is not evident, double click on the connect point that was just created. The thermal diagram should appear. Use the scroll bars to move back and forth in that diagram until the connect point labeled “air” is visible. Move this icon (shift and drag) if it is not close to the boundary node. The two networks should appear roughly as shown above at the right above.

A **Save Everything** should be performed at this point. If problems are experienced during the generation of ties, you can then return to this point by double clicking “myimage.im” on a PC, or typing “sinaps myimage.im &” in a shell window on a Unix machine.

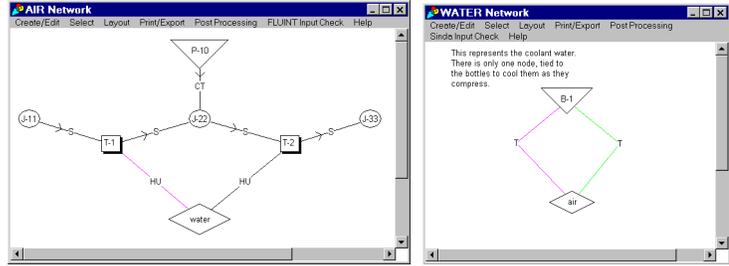
In the fluid network diagram, choose **Create/Edit→Ties→Add Single**, and then choose HTU from the selection list. A cross-hair cursor will immediately appear. Note the prompt instructions in the *System Transcript window* if you are unsure of what to select next. Select tank 1 and the new connect point labeled “water” in either order. The thermal submodel will then pop up to the top of the screen. Click the boundary node and the “air” connect point in either order. An HTU tie edit form will appear. Add UAwater as the tie conductance in that form (shown at the right), then accept and press Alt-X to generate the tie. The tie will appear as “HU” in the fluid diagram, and as the less-specific “T” in the thermal diagram, where it is green to indicate that it is not owned by (and therefore cannot be edited in) that diagram.



Number	1
UA	uawater
DUPN	1.0
DUPL	1.0
UAM	
AHT	
UB	
UEDT	

* These instructions are overly cautious since they are intended for the novice user. Once the user has a feel for SinapsPlus windows, how they work for the host windowing system, and how to navigate using connect point clicking, then it is rather easy to make ties and intermodel conductors. However, it is a common problem to begin such operations with one required network diagram closed, and it is easy to leave too many unnecessary windows open at once, obscuring the ones that are currently required.

We are ready to generate the second tie reusing the same connect point. However, to prevent the two ties from residing on top of each other in the thermal diagram, hold shift and drag the “T” label in the thermal diagram to bend the tie out of line. Now select again **Create/Edit→Ties→Add Single**, choose HTU, and select tank 2 and the connect point. In the thermal diagram, select the node and connect point once more. Again provide UWater as the UA value, and bend the new tie out of line if you wish in the thermal diagram. The final networks should appear roughly as shown above at the right.



In these diagrams you will notice that the one tie has been colored magenta to distinguish it from the other. To repeat this optional operation, go to the thermal diagram and select one of the ties (such that a red splotch appears next to it). Select **Create/Edit→Conductors→Color** and choose magenta or cyan or some other convenient color. The color of the tie will change in both diagrams. Note that for this operation and for the “move one end” operation, ties are treated as if they were conductors in thermal networks, since these are the only two operations that can be performed from within the thermal window.

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

Global Control Data—Open this form via the option on the model control tab. This window should appear as depicted in the upper portion of the picture below.

Control Data (Global)

TIMEO	0.0	PATMOS	-101325
TIMEND	(air.pl1 > 0.999*air.pl10)? timen :	ACCELX	0.0
ABSZRO	-273.15	ACCELY	0.0
SIGMA	sbconsi	ACCELZ	0.0
NLOOPS	1000	UID	<input checked="" type="radio"/> SI <input type="radio"/> ENG
DTIMES	0.0		

Control Data (Thermal/Fluid)

DTSIZF	0.1	FRAVER	0
DTMAXF	1.0E30	RSSIZF	0.5
DTMINF	0.0	RSMAFX	1.0
DTTUBF	0.1	RSTUBF	0.5
OUTPTF	1.0	ITHLDF	0
OPITRF	0	RMFRAC	0.5
RERRF	0.01	RMRATE	
REBALF	0.01	RMSPLT	0.1
ITROTF	0		

Buttons: Accept, Reset

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.

Values which defined the model units such as **ABSZRO** (which should be set to -273.15 meaning degrees Celsius), **PATMOS** (which should be set to -101325 pascals for gauge pressure), and the UID radio-button have already been defined by the simple model wizard.

The value for **TIMEND**, the problem end time, needs to be defined as well. Since we named the pressure source plenum #10 and the bottles tank #1 and #2. The expression for **TIMEND** to stop the transient when the bottle pressure is 99.9% of that of the pressure source is then:

```
(air.pl1 > 0.999*air.pl10)? timen : 1.0e30
```

In other words, the end time is “infinite” (1.0e30) until the pressure in the first bottle exceeds 99.9% that of the pressure source, then the problem end time is set equal to the current time (timen or “time now”), which terminates the transient solution. This logic could equally well have been supplied in **FLOGIC 2** of the fluid submodel (or **VARIABLES 2** of the thermal submodel) using Fortran-style statements as follows:

```
if(air.pl1 .gt. 0.999*air.pl10)then
    timend      = timen
else
    timend      = 1.0e30
endif
```

Note that the full expression for **TIMEND** is not visible. It can be edited by scrolling the expression inside the visible field, or by holding the cursor over that field, pressing the right button, and selecting **edit** in a window.

“Accept” the page, which should appear as shown above at the top, and move to the fluids tab.

The only values that needs be changed in the fluid and thermal submodel control constants are **OUTPTF** and **OUTPUT** respectively, the output intervals. The simple model wizard set both values to $0.01 * \text{timend}$. Set both these control constants to 1.0 as shown in the middle of the above figure. This will cause all **OUTPUT CALLS** to be invoked every second during the transient solution. “Accept” the pages and move to the thermal tab.

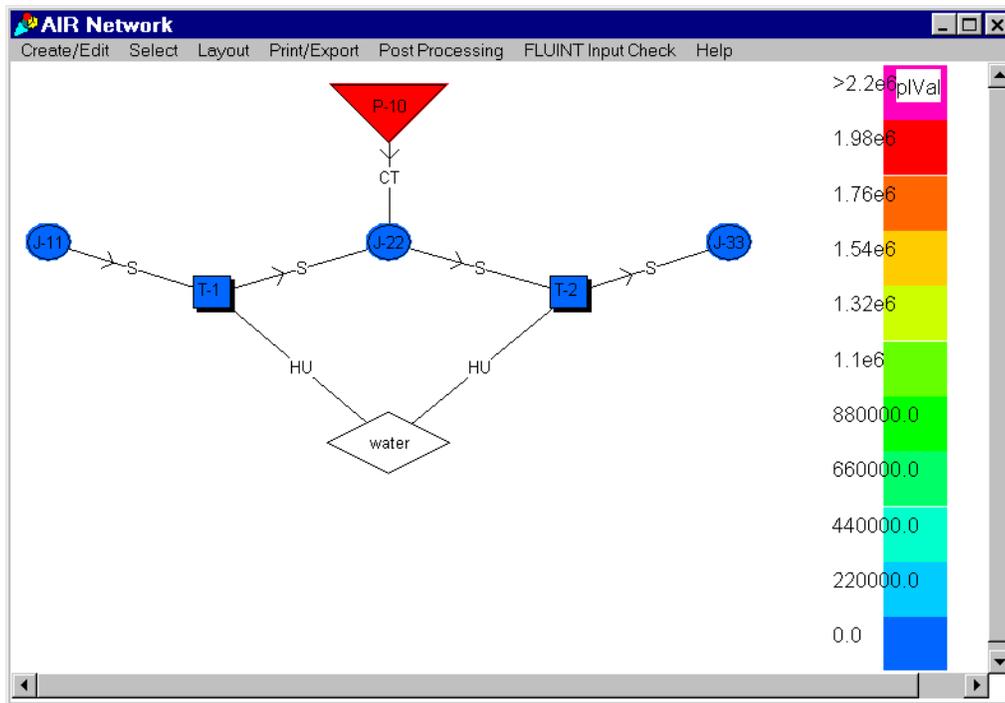
Checking the Model

Input Checks—The basic inputs to this fluid model are flow areas, initial temperatures, and initial pressures. To make sure that the input values and expressions are correct, we will color the lumps and paths successively by these values using the options under **FLUINT Input Check**.

To color lumps by pressure, choose the option **Select→Everything** (or **Alt-A**), then select **FLUINT Input Check→Lumps→PL**. A vertical color bar will appear under the mouse cursor, waiting to be dropped onto the screen as were nodes, lumps, and comments. Like any other icon, color bars may be moved within the diagram window. They may also be changed to horizontal etc. (Color bars and their options are described more fully described later under the postprocessing section.)

Once the color bar is placed, the lumps will be colored according to their initial pressure. All lumps should be blue except the plenum, which should be red. On the color scale, the highest value should be $2.2e6$, and

the lowest value zero. Analogous operations can be performed to color lumps by temperature, and paths by flow area (these operations are not shown in the figures that follow).



Before proceeding, the input checks should be cleared using FLUINT 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 locating 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 SinapsPlus, 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 SinapsPlus 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 SinapsPlus) machine via a method that preserves its binary integrity (e.g., disk, binary ftp).

* WARNING: A similar option appears on the submodel control panels, 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 SinapsPlus.

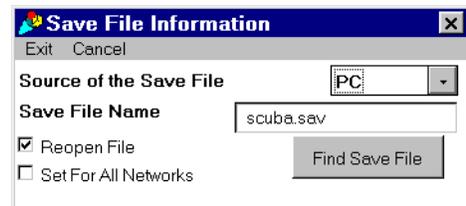
Launching SINDA/FLUINT—If SINDA/FLUINT resides on the same machine as does *SinapsPlus* along with an appropriate compiler, then SINDA/FLUINT may be launched directly from within *SinapsPlus* without creating an ASCII input file. Simply select **Run Sinda/Fluint**→**Preprocess and Run**→**Normal Operation** from the Model Control Panel. The SINDA/FLUINT job is now running as a separate background process. (On a PC, a rather large DOS window will appear while the processor is running, along with a status window. On a Unix machine, the run can be monitored from the shell from which *SinapsPlus* was launched using standard Unix commands.) Once it is complete, and once the processor output file (scuba.out) has been reviewed to make sure that the run was successful (error messages, if any, will pop to the screen), the save file (scuba.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 TRANSIENT 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 *SinapsPlus* (if it was exited) and should be in the SCUBA model desktop, with the AIR fluid 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.

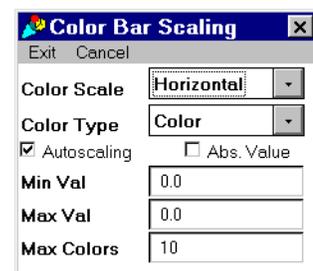


If the save file was generated on a different machine, and has not been placed in the same location as *SinapsPlus*, do so now.*

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 *SinapsPlus* does not find the save file, it will advise of that failure.

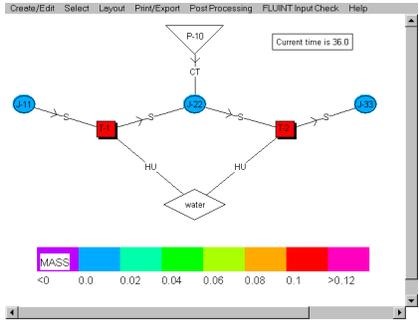
Next, select **Postprocessing**→**Set Starting Time/Record**.† A list of times (with corresponding integer record numbers) stored on the save file will appear. In this particular sample problem, *about* 36 such rows should appear, with times increasing from zero to *about* 36 in increments of one. Select the last (bottom) row. You will then be prompted if you want this starting time to apply to all the network diagrams. Choose “yes.” While plots versus time, register, etc. do not need a starting record to be identified, other postprocessing operations (bar plots, plots versus position, colorization, etc.) that are made at a single time point will require this time point to be identified.

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, horizontal is chosen (see right). Actually, color bar sizes and styles can also be edited after their creation as well simply by double-clicking the color bar after it has been created.



* The “Find Save File” option, which would otherwise be used to locate the save file, will not be covered in this tutorial.

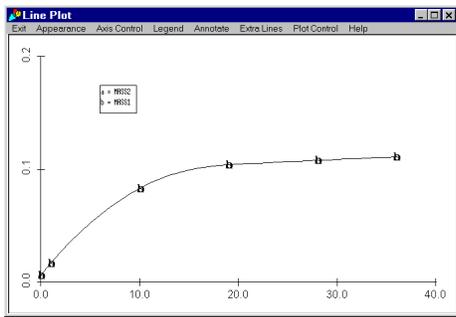
† This step is actually unnecessary for X-Y and polar plots versus time, which access all records in the save file.



appear, with the two tanks having an equal mass of about 0.11 kg. (Note, the color bar in that diagram has been edited for esthetics.)

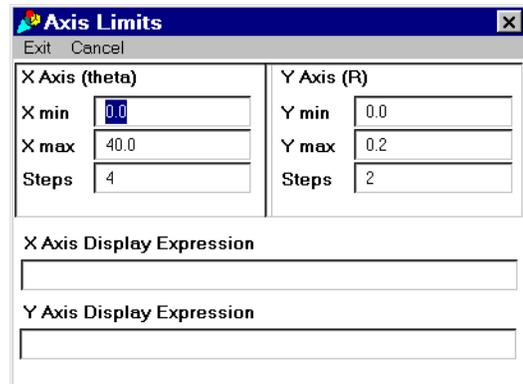
Finally, the thickness of paths can also be simultaneously postprocessed by a wide variety of values using **Postprocessing**→**Thickness**→**Paths** after selecting all paths (Alt-A). Choose for example “DPRESS” or delta pressure (pressure drop, about the sixth item in the list), and place the resulting thickness scale results in the diagram. Thickness scales may also be customized by double-clicking them.

As can be seen by this postprocessing, the pressures have equilibrated. But of primary interest is the mass in the tanks, and perhaps the temperature history required to explain that mass.



Plotting by Time—Select both tanks in the model, perhaps by pressing Alt-U to unselect everything, then touching each one of the tanks such that the red splotch is visible. Select **Postprocessing**→**Plotting**→**X-Y Plots**→**lumps**, then select “MASS” from the list. A new window will appear containing an X-Y plot as shown at the left (with a legend added per the **Legend**→**on/ move/refresh** option in the plot pull-down menu). Note that both the “a” line and “b” line are coincident, and therefore the labels are obscuring each other. This means that the two tanks indeed responded identically, which is a good sign that the model was constructed correctly.

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. For example, to display pounds instead of kilograms, input “y*2.2” as the Y Axis Display Expression.



There are *many* powerful line plot options available. The user is left to explore them as time is available.

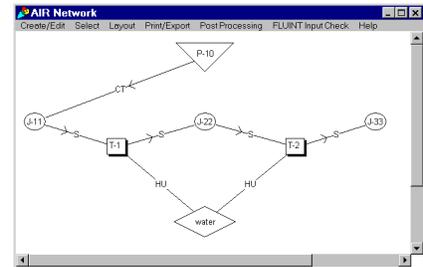
To understand the temperature history of these tanks, repeat the postprocessing operations listed above to plot the TL, and leave both the mass and temperature plots on the screen although perhaps collapsed to an icon. These will be used to compare with an alternate case. The temperatures of the tanks exceed 100 degrees during the compression process.

Text Output—To find the exact value of the mass in each tank, the data in the above plot can be sent to a file or viewed in the window using the **Plot Control**→**File Ops** →**Points**→**Show in Window** option. Or, text can be the primary result of postprocessing using the **Postprocessing**→**Text**→**Lumps** option.

Alternate Design Evaluation

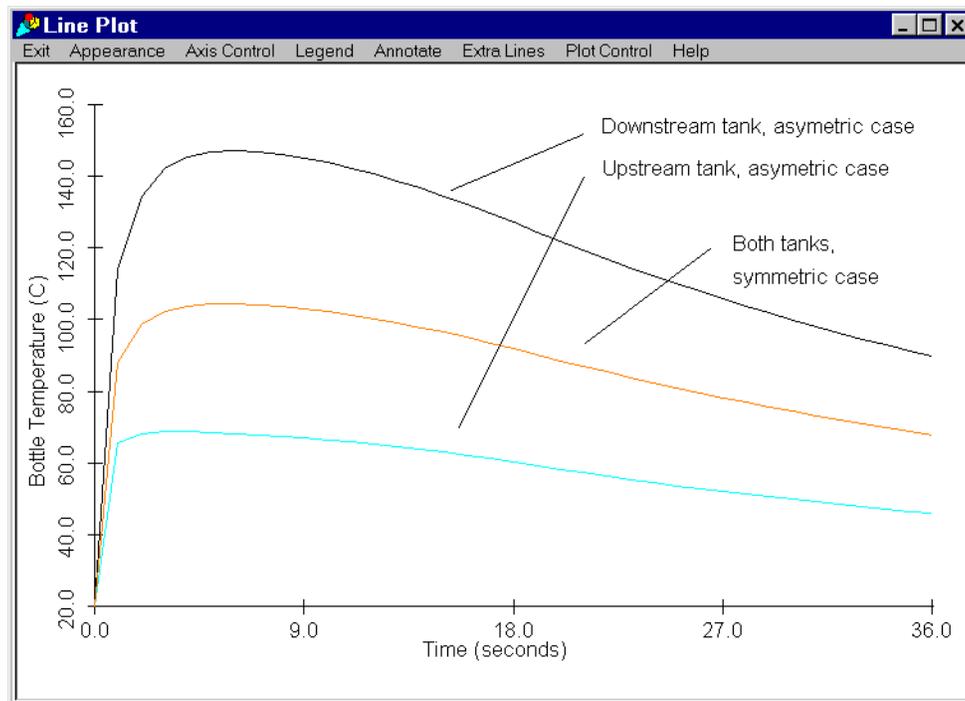
We now want to compare the results of the above analysis with an asymmetric case, with the pressure source attached to point A. We must therefore move the CTLVLV path to junction 11.

Unselect everything, and perhaps reset any postprocessing colorizations or thickening. Now select the CTLVLV by touching its label “CT” such that a red splotch appears next to it. Select **Create/Edit→Paths→move one end**. Cross hairs will appear, prompting us to select the lump from which the path should be disconnected: select junction 22. The program will then wait for us to select to lump to which it should be reconnected: select junction 11. The network should appear as shown at the right.



Now we can rerun the model, using again **Run Sinda/Fluint→Preprocess** and **Run→Normal Operation** from the Model Control Panel.*

Once again, postprocess to create plots of tank mass and temperature. (You can actually convert your existing plots to the new data by using the **Regenerate the Plot** option under **Plot Control**, but you will lose your old plots for comparison.) Ways to directly compare save files from different cases or other advanced plotting manipulations are not covered here. But a composite plot of both runs is presented for information purposes only below.



As the results show, tank 1 is colder than the first run, and tank 2 is hotter since it is not receiving its air from a 20°C source (as in the symmetric case), but from tank 1 which is heating up due to compression.

* This operation overwrites the output files, which should be renamed in OPTIONS DATA to avoid this result. (See also the CHGSAVE utility in the SINDA/FLUINT User's Manual.) Also, “Normal Operation,” a complete recompile/relink, is needed since the network has changed. If multiple CTLVLVs had been used and opened or closed via editing, or if their FK factor had been a function of a register whose value had changed, then the “Rerun Existing Executable” operation could have been used instead, which saves time on large models.

This means that tank 1 will end up with more mass than in the first (symmetric) case, but tank 2 will end up with less. If the goal of the pressurization is total mass in both bottles, then either attachment point will do. But if the goal is even pressurization, then location B should be chosen.

For those readers curious about this seemingly strange effect, in which the temperature of the first case appears to be split into higher and lower temperatures in the second case, the trend continues with more and more bottles in a row: the first bottle gets progressively colder, approaching the inlet condition of 20°C, while the last bottle gets progressively hotter.

Final Notes

This sample problem tutorial has introduced the basic operation of *SinapsPlus*. It did not cover many layout and diagramming options, multiple submodels, advanced postprocessing features, prebuilt models, nor thermal submodels in any detail. 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 *SinapsPlus*, 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.

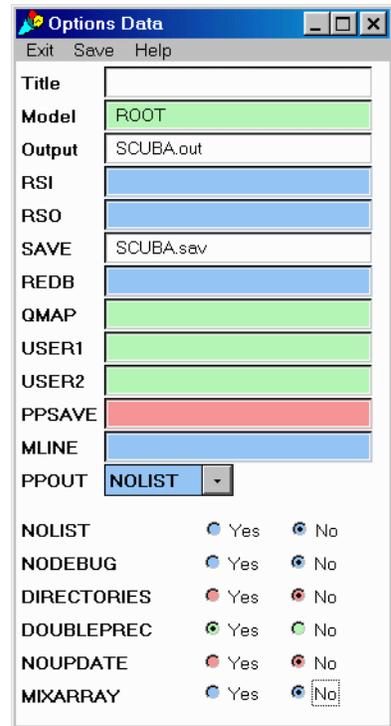
Appendix A

This appendix provides a brief overview of what was created when we opted to create a model using the template wizard versus selecting an empty model.

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 right. 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 SinapsPlus. It is used to name the model on ASCII output pages.

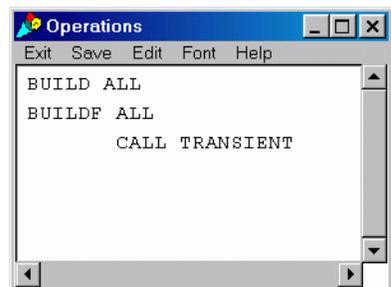
The simple model building wizard defined an *output* file and a *save* file. The output file is an ascii text file which will be created during execution. The user specifies what information is to be written to this file. The save file is a binary format file available for postprocessing within SinapsPlus.

Provide a title (“Bottle Pressurization 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 (“scuba.out”) and a save file (“scuba.sav”), until the form resembles the one shown below. Move the mouse button over to the Exit/Save zone on the pull-down menu bar, and select “Save and Exit.” The options form, with its data, is now saved to memory (but not to disk).



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

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.*

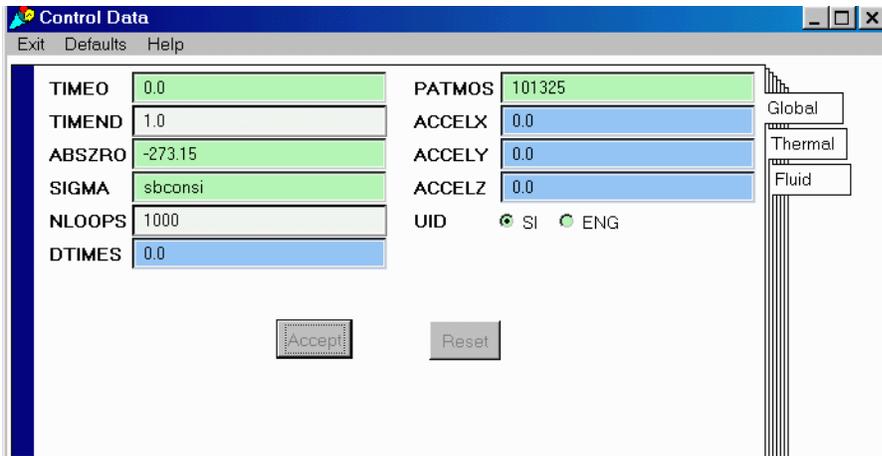


```

BUILD ALL
BUILD F ALL
      CALL TRANSIENT
  
```

For this sample problem, the only logic are the build statements (BUILD and BUILD F) and the call to TRANSIENT. 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 you have time, you should experiment with the options available under “Edit” for future reference. Otherwise, when you are finished, use the “Save and Exit” option available under “Exit.” The Operations window will then be saved to memory and will be closed. 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.



Global Control Data—

Open this form via the menu on the *model control panel*. This window should appear as per the picture below. (If you have completed the above tutorial, an expression will appear in the window for TIMEND.)

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. In addition to an OUTPUT time interval has been set to “0.01*timend” for the thermal and fluid tabs. (Again if you have completed the tutorial, the value of OUTPUTF has been changed to 1.0.) These control constants designate how often the output block will be executed. Even if we don’t have any output logic defined in a thermal submodel, a value needs to be set for OUTPUT. SINDA/FLUINT will abort a transient run if output intervals are not defined.

Output Calls—The building wizard has created calls in *Output Calls* associated with both submodels (thermal and fluid). 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 (see below). Note that calls to NODTAB and SAVE have been created automatically. The NODTAB will provide a tabulation of all node output data. In the fluid model *Output Calls* calls to LMPTAB, PTHTAB, and TIETAB have been created to provide tabulations of all lump, path and heat transfer tie data.

The call to SAVE will create a binary results file named “strut.sav” (as specified in *Options Data*). (Note this is different than a *SinapsPlus* binary file.) This file may then be used to postprocess the model. If multiple submodels have been defined using the building wizard, the call to SAVE will be placed in the output data block associated with the first submodel defined; in this case the thermal submodel “water.”



```
air Output Calls
Exit Save Edit Font Help
c
c these are some convenient/common output instructions that will
c be invoked before and after the steady state run, and every
c OUTPTF seconds or hours
c specified above:
c
c if you are using multipl
c you probably want to rem
c get duplicate outputs
c
c     CALL LMPTAB ('ALL')
c     CALL PTHTAB ('ALL')
c
c if you have heat transfe
c and fluid networks, add:
c
c     CALL TIETAB ('ALL')

water Output Calls
Exit Save Edit Font Help
c
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
c     CALL NODTAB ('ALL')    $ Text (tabulation) node output
c     CALL SAVE ('ALL',0)   $ Binary (postprocessing output
c                           $ for EZ-XY or SinapsPlus
```