

5 Modifications Applied to *Tmin*

As stated in Chapter 1, the following modifications to *Tmin* were as follows: 4 new alternating stress-to-number of cycles (S-N) curves, a 2-D vertical piping span, an output of the data to a Microsoft Word[®] document, (*Doc*), and additional user-friendly items. Chapter 3 documented the addition of the S-N curves in the ASME format. This chapter will detail the output screens for the 2-D vertical piping span, for the Microsoft Word[®] document and the user-friendly items. Since this program was written in Visual Basic[®], the primary computer operating platform is Microsoft Windows 95[®] and up.

Some of the user-friendly additions included help files. Within each of the property command buttons, a help file was added. This was to help the end user in understanding what process is done or to show what the user needs to know before entering the data for the piping span being evaluated. Finally, the visual output of the 2-D vertical piping span will be presented.

5.1 User-Input Additions

Since this program was written to save engineering time and money at Dupont facilities by having floor personnel use *Tmin*. The additions would make *Tmin* a user-friendlier computer program. The additions included to *Tmin* are as follows:

- *Save file, Save As file, Open file, New* calculation buttons
- Help windows and help buttons
- Pipe-span choice input form
- 2-D vertical piping span
- Output of data to a Microsoft Word[®] document

Save file, Save As file, Open file, New calculation:

The end users of *Tmin* requested that previously input *Tmin* models and calculation runs be saved for later use. To do this *Save*, *Save As*, and *Open* commands were to be created using a drop-down menu as seen in Figure 5-1 [34, 36].

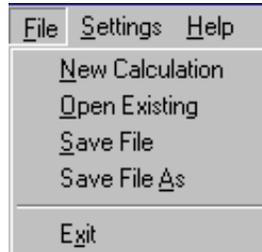


Figure 5-1. File Drop-Down Menu Additions

Piping span data input uses a pre-existing form that was already created in a previous version of *Tmin* used buttons called command buttons. The command buttons are seen in Figure 5-2. Pressing a command button opens a dialog box, which enabled the user to either manually enter in a variable or select one from a list. These input* variables are then stored by the program and saved for later use. The data input are as follows: Area, Plant, User, Title, Material Name, Pipe Size and Schedule, Temperature, Pressure, Specific Gravity, and Insulation type. As seen in Figure 5-2, some default variables are in place, such as the specific gravity of the internal fluid, the number of cycles (discussed in Chapter 2), and the piping insulation type. When all fields are input, the *Calculate* button can be clicked. The user will proceed through a few screens until the final output screen is displayed. In the final output screen shows the pipe-wall thickness values for the piping span.

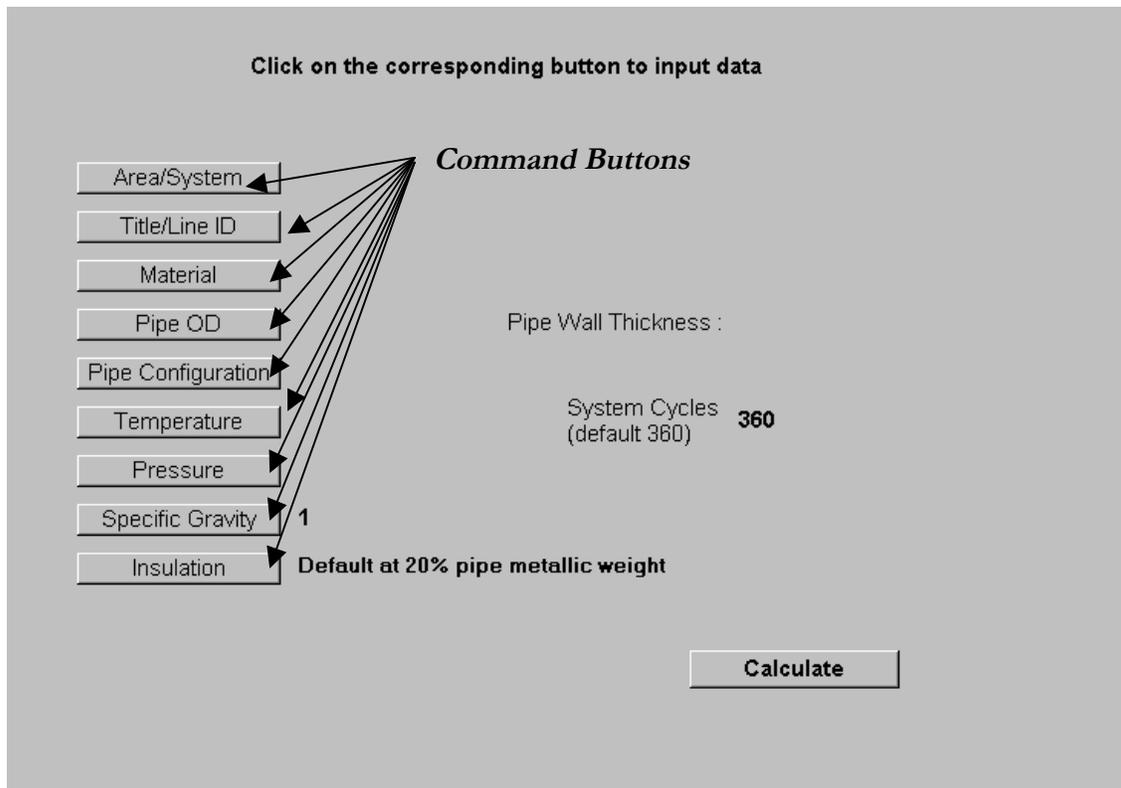


Figure 5-2. Input Form of *Tmin* Showing Command Buttons

Some problems were faced with saving and opening the data input by the user. One problem was input boxes that detail the piping configuration would need to be filled in automatically. The second problem was to graphically duplicate the piping spans originally chosen by the user. Possible solutions are discussed later in this section. The data used by the program was saved to a file and as a result, the *Save* and *Save As* commands worked.

The *Tmin* configuration files would be saved with an extension of "*Tmn*" and are saved to the "C:\My Documents" folder by the default of Visual Basic®. A screen-shot of the "*Save As*" input box is seen in Figure 5-3. The "*Save*" input box is nearly identical to this figure [35]. The next item is the *New Calculation* button.

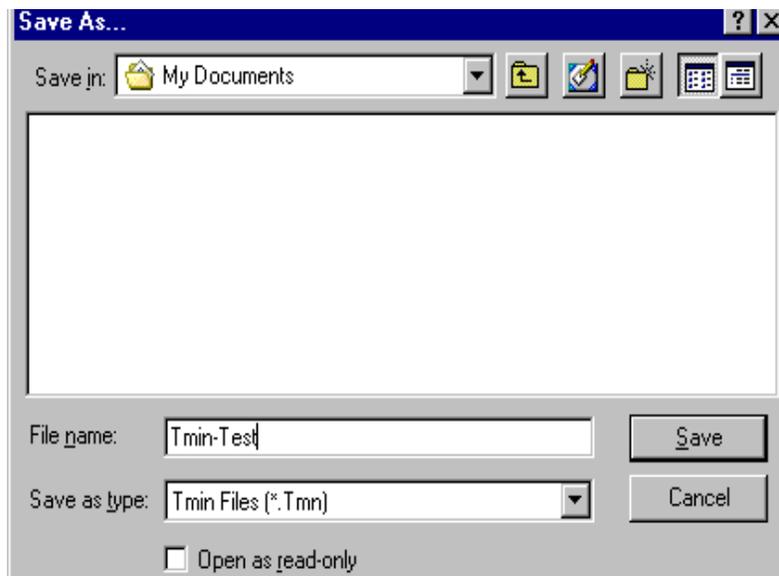


Figure 5-3. Save As Input Box for Saving Previous Piping

The *New Calculation* option works as follows: when this button is chosen, the variables are cleared only after responding *Yes* to a *Yes/No* prompt that warning of the loss of input data when *Yes* is selected. This ensures that the user will not inadvertently loose his/her data.

All input variables that were saved in the *Tmn* file are not retrieved properly into *Tmin*. In order to use the *Open* command and properly re-initialize the program with these variables, the program must ignore an error check within the *Tmin* code [35]. This error check determines whether the data was input from a saved file, or is from a current user input case. As a result of this problem, a Visual Basic[®] module would need to be written that would input the variables from the saved file. However, the saved files cannot directly save the piping configuration because of the current *for-loop* statements within the code. These *for-loop* statements change dependant on the type of piping configuration chosen. Since the piping configuration could not be re-initialized, DuPont was contacted about this situation and they decided that due to time constraints, that this addition would be added to the next update of *Tmin*.

Help Windows:

The second bullet in the added features list involves a Help window when the user selects a command button. The Help file explains what calculations are taking place or what this command button does. Some discrepancy as to what is taking place, or what value format needs to be input to the input boxes has been stated by end-users as to what is occurring in these calculations. As a result, these Help windows will assist in the end-user in understanding what is occurring during the final calculations. Figure 5-4 shows a help button that was added and will appear when the user presses the Pressure command button.

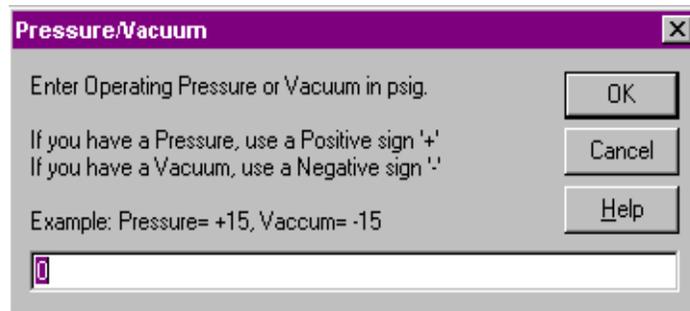


Figure 5-4. Help Button in an Input Box

Once the user clicks the **H**elp button, a help pop-up window will appear and overlay the command Input window to explain how to input the data. The Help window can be moved or resized so the user can leave the Help window open while inputting the data. This Help window is seen in Figure 5-5.

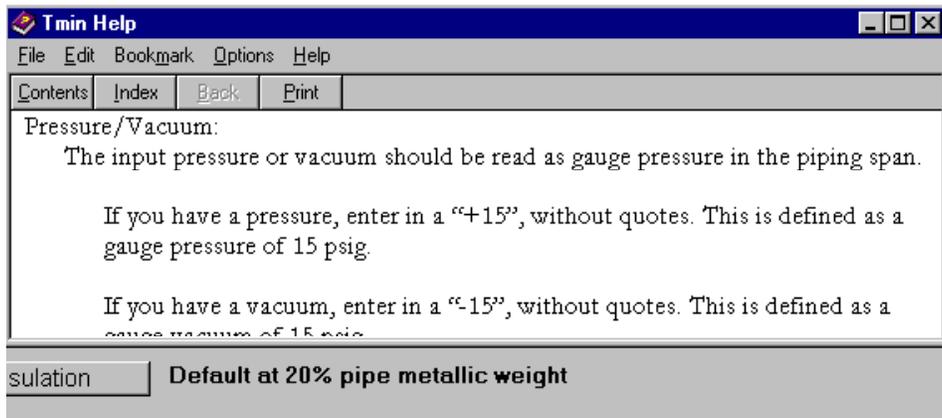


Figure 5-5. Help Pop-Up Window Example

Having Help buttons creates a computer program that is more user friendly and will give users an idea of how to and what to input. This is in contrast to a computer program that alienates the user into using a little black box with little or no understanding of its capacities. The Help files were created using a detailed Visual Basic® help book written by Reselman, et al. [39]. Without this book the Help files for *Tmin* could not have been created. In Appendix E, a detailed list was created to help the reader create a Help file for his or her own projects.

Pipe-span choice form:

This program originally analyzed horizontal piping spans that include elbows, pipe Tee's and straight pipe spans. The addition of the 2-D vertical piping span required a new input form to for making pipe span choice at the request of DuPont. The form created is seen in Figure 5-6. For the horizontal piping span choices, DuPont requested a button for the straight pipe to be titled "Straight Pipe," for the Elbow section, a "Bend," and for the Tee piping section, they wanted it called a "Tee." For the 2-D vertical piping span, this button was called the "2-D vertical piping span." This form allowed the user to choose the piping configuration to be analyzed by selecting a fill-in button. *Tmin* would then automatically go to that piping span form chosen. The Visual Basic® code for this form can be found in Appendix D.

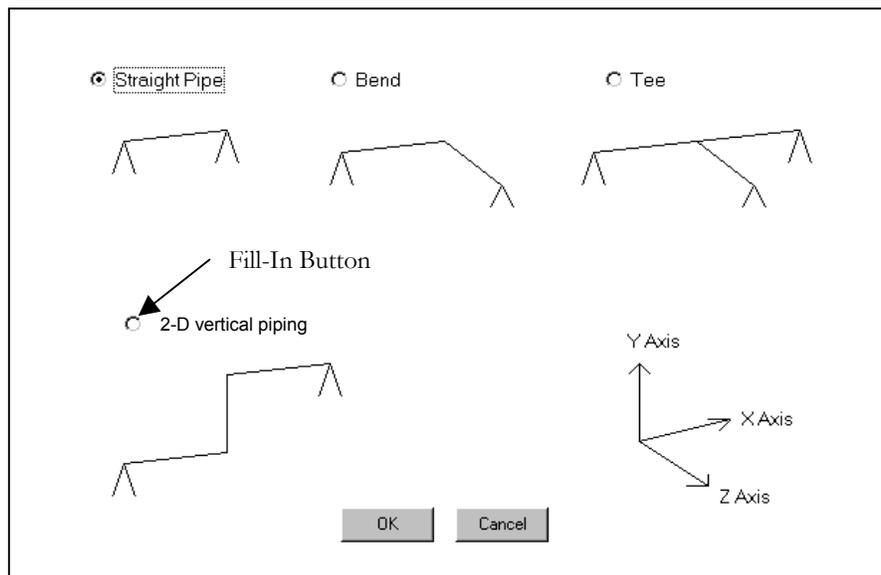


Figure 5-6. Piping-Span Input Form in *Tmin*

Output of data to a Microsoft Word® document:

The previous version of *Tmin* allowed the user to print out the data and final results directly to a printer. This resulted in problems. Problems arose from incompatible printers and incorrect internal fonts. Another problem was the inability to save the output data to a file. This meant that the final output pipe span configuration could not be emailed to anyone. The updated version of *Tmin* will send the final results and the piping configuration to a Microsoft Word® document.

To send the data to the Microsoft Word® document a Microsoft template, (*Dot*), was created. “Bookmarks” were created within the template. When the user presses the “Open Document” button in the output screen, the program takes the data used and passes it to the bookmarks [35, 37]. This process was relatively simple in comparison to capturing the final piping configuration image. In order to capture a piping configuration image, a complex computer code that was found on an Internet web-site [40]. The code could copy the configuration as a bitmap image and save the file created to the computers root directory. Once the image is saved, the program needed to copy this image to the template. To do this, a macro was generated within the template that would retrieve the bitmap and paste it into a text-box created in the template. Again, this seemed relatively easy, however, with use of the “Open-Document” button, the image would not appear. The solution was found in a single line of code written by Evangelos Petroutsos [37]. Through the help of this code, the macro created in the template was called to “run” once the “Open-Document” button was pressed. Examples of these output documents can be found in Appendix E. The Visual Basic® computer code used for the “bookmarks” and the code to call the macro can be found in Appendix D. In the next section, the 2-D vertical piping span to be included will be detailed.

5.2 Creation of 2-D Vertical Piping Span Output Form

This section details the user inputs that will be seen by the end user when the 2-D vertical piping span is chosen. When the user chooses the vertical span, seen in Figure 5-6,

the form (in Figure 5-7) will be shown. The 2-D vertical piping span graphic configuration is seen at the bottom of the form. Choices for pipe span length and elbows are in the menu boxes. Warnings to the right of the dialogue boxes signal the user that the field has not been entered. As an example, the elbow dialog box has the warning. “Warning! No Elbow Chosen.” When the user chooses a Short Radius Bend Elbow (for example), the text for the selection will be posted in the dialogue box and the warning will disappear.

The screenshot shows a software dialog box titled "2-D Vertical Piping Span Form". It contains five rows of input fields:

- Length of Pipe Span L1:** A text box containing "Straight Pipe" and a warning: "*Warning! No Span Type Chosen for L1*"
- Elbow at end of Pipe L1:** A dropdown menu with "Standard Long Radius Bend" selected and "Short Radius Bend" as an option. Warning: "*Warning! No Elbow Type Chosen*"
- Length of Pipe Span L2:** A text box containing "Straight Pipe" and a warning: "*Warning! No Span Type Chosen for L2*"
- Elbow at top of Pipe L2:** A dropdown menu with "Standard Long Radius Bend" selected and "Short Radius Bend" as an option. Warning: "*Warning! No Elbow Type Chosen*"
- Length of Pipe Span L3:** A text box containing "Straight Pipe" and a warning: "*Warning! No Span Type Chosen for L3*"

At the bottom of the form are "OK" and "Cancel" buttons. Below the buttons is a 2-D diagram of a piping system. The diagram shows a vertical pipe section connected to a horizontal pipe section. Three spans are labeled: L1 (the lower-left horizontal span), L2 (the vertical span), and L3 (the upper-right horizontal span). A coordinate system is shown with a vertical "Y Axis" and a horizontal "X Axis". Text in the top-left corner of the diagram area reads: "Units are Length - ft. Load - lb."

Figure 5-7. 2-D Vertical Piping Span Form

To ensure that the correct pipe span lengths are input, the user will see the value appear on the form for that span length. The procedures for entering information regarding the lower-left piping span will be detailed next. When the user chooses the top input field, labeled “Length of Pipe Span L1,” a pop-up window will appear asking for the span length. See Figure 5-8. In this pop-up box, the user can enter in the span length for the lower-left piping span, once this is done; the user presses the OK button and can proceed to the next input field.



Figure 5-8. Input Box for Obtaining Pipe-Span Lengths

After the user enters in the span length, a second pop-up box appears, Figure 5-9, asking if valves are present in that span. If the choice is No, the user can then proceed to the next input menu.

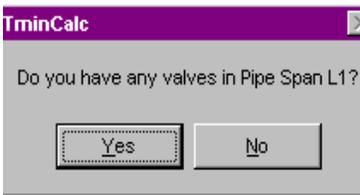


Figure 5-9. Input Box for a Valve in the Span Length

If the choice is yes, a second pop-up window (similar to Figure 5-8) will appear asking the distance of the valve along the piping span. An error check is present in this box. If the user enters in a distance length equal to or greater than the pipe-span length, an error box appears. The appearance of the error box forces the user to re-enter the distance to the valve. The distance to valve error box is used for the assumption that only straight valves are in place and none are present that have an elbow in-line with the valve. Finally, the weight of the valve will be input in another input box. This form is the valve connections form in Figure 5-10. As seen in this figure, several choices are available for valve connections. Some valves can be assembled by either a butt-type weld, by a threaded connection, or other attachment methods. These choices are needed for correct evaluation of the stress-intensity-factors detailed in Chapter 4.

Valve Connection Type

Choose a connection type for the Valve

Butt welded or weld neck flange

Double-welded slip-on flange

Socket weld flange

Lap joint flange (with AMSE B16.9 stub)

Threaded flange

Don't know, or Mixed connection type

OK

Figure 5-10. Valve Connection Form

Next, the 2-D vertical piping span form will show the distance to the valve and its corresponding weight seen in Figure 5-11. Following the procedure of the lower pipe span, the user will then be asked the same questions through the input boxes for the vertical segment and upper-right piping segment lengths.

Length of Pipe Span L1	Straight Pipe	Straight Pipe
Elbow at end of Pipe L1	Standard Long Radius Bend Short Radius Bend	Short Radius Bend
Length of Pipe Span L2	Straight Pipe	Straight Pipe
Elbow at top of Pipe L2	Short Radius Bend 3D Bend	3D Bend
Length of Pipe Span L3	Straight Pipe	Straight Pipe

OK Cancel

Units are
Length - ft.
Load - lb.

Vertical Pipe Span--Two Axis

Figure 5-11. Example Span with an Additional Weight

In addition, since there are two elbows in the piping span, there are elbow choices available. These elbow choices are identical to the horizontal elbow span choices. Some types of elbow's that are available are a *Standard Long Radius Bend*, *Short Radius Bend* and a *3D*. If the user does not know the elbow choice, an option for *Don't Know* is also available.

Once the user has confirmed all input data and is ready to continue, the calculation button is pressed. The calculations that take place are detailed in Chapter 4. After all calculations are complete; a final output form is displayed. On this output form, several pipe-wall thickness values are shown, which are detailed in Chapter 2. When the final calculations are complete for the 2-D vertical piping span graphic configuration, the user will not know where the critical section would be in the span. Therefore, the 2-D vertical piping span form was created so that the user would know the position of the critical piping section. Displaying a red pipe section shows the critical piping section. The example calculation shows the red piping section in Figure 5-12. The computer codes for these functions can be found in Appendix D. Other additions that were included are detailed in the next section.

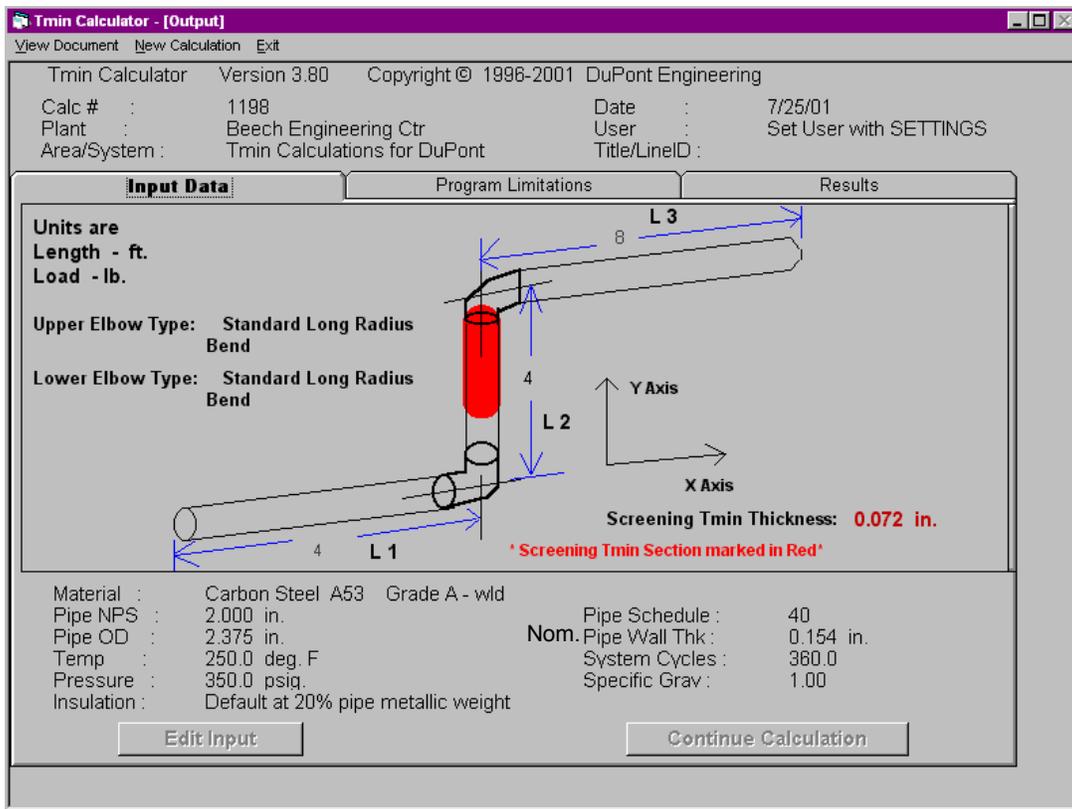


Figure 5-12. Critical-Piping Section Marked in Red on 2-D Vertical Piping Span

5.3 Other *Tmin* Additions

Some code additions requested by Dupont could not be included due to time constraints. The additions that were not included were the creation of an internal Visual Basic[®] database, a login screen, and screen resizing. First, the internal database was never created due to the complexity of re-working the *Tmin* internal codes. Second, the login screen would have saved the users ID for later use in the Microsoft Windows[®] registry and would have included their Name and Title. Due to time constraints the login screen addition was never realized. And finally, the screen resizing would have automatically scaled all forms according to the screen resolution. However, the Tab forms that are used on *Tmin*, do not allow a forced resize to occur.

One more minor addition was implemented into *Tmin*, such as the removal of the 'X' button (close button) on the forms. Users being able to close the program using the 'X' command on the form resulted in leaving the *Tmin* program into the computers internal memory. This created a problem when a newer version of *Tmin* was installed. As a result, the computer would still think the program was running and would not install. A code was found for this problem and through the use of the code the 'X' button could be disabled [37, 38]. In the next chapter numerical examples of the 2-D vertical piping span will be shown.