CalCS.bas, and FrFact.bas are files that have been translated from FORTRAN codes into Visual Basic for use in Excel and to interface with the PVT add-in from Dr. John McMullan. To read the files into your spreadsheet, open a new spreadsheet. Once the program is open, click on Tools and then Macro and then Visual Basic Editor (alternatively, you could simply hit Alt+F11). This opens the Visual Basic for Applications Environment. You should see a box entitled Project – VBAProject in the upper left-hand side of the environment. The highlight should also be over Sheet1. Click on Sheet1 and turn it blue. Then go to the top menu and select File and then Import File. Find the file out in the directory you selected and select the file. You should now have an entry in the Project – VBAProject called Modules under the Sheets. Import both the other file in a similar manner. Double click on the Modules entry that is under your Sheet1. You should have two values – Module333 and Module444 (if you don't and you actually have Gas Props, Oil Props and Water Props, that means that you have selected the Modules entry that is under the PVT add-in). If you want to look at the code, double click on the program you want to look at (Module333 is the friction factor code and Module444 is the Cullender and Smith code). This should bring up the function in the right-hand window. Next, click on the Tools entry in the top menu; then click on References. This will bring up a menu that says References – VBAProject. Look for the PVT listing and click on the box next to PVT and make sure there is a check mark in the box. Once you have done this, click on OK. If you do not see the PVT listing, that means that you have not loaded the PVT add-in and you must do so. Assuming that you found the PVT listing and all is well, you next, click on Debug in the top menu and then Compile VBAProject. You should not get any messages if your code compiles. It is then ready to use.

You can test this code by exiting out of the VBA environment and returning to your open spreadsheet. In one of the cells, type in
=CalCS(2000,110,245,0.75,4915,10000,10,0,2.441,0.00084)
followed by the return key. You should get 2796.598 as a value in the cell. The functions are now ready for your use.

This is written for Excel – other spreadsheets also use Visual Basic for Applications (VBA), but I only have access to Excel. If you use another spreadsheet, you are on your own! The parameters for the functions are:

1. CalCS(Pstart,Tstart_degF,Tlast_degF,G,QG_mscfd,Length_feet,Intervals, Angle_radians,Dia_inches,Roughness_inches) where Pstart for this code is the downstream pressure (wellhead pressure for flow in a tubing string; separator pressure for flow in a flowline), Tstart_degF and Tlast_degF are the downstream and upstream temperatures in degrees Fahrenheit, G is the specific gravity of the gas, Length_feet is the length of the flow tube in feet, Intervals is the number of panels you want to break the length into, Angle_radians is the inclination angle of the tube (0 degrees is vertical), Dia_inches is the internal diameter of the flow tube in inches and Roughness_inches is the relative roughness factor in inches.
2. FrFact(REY,ED_Dimensionless) where REY is the Reynolds number, and ED_dimensionless is the dimensionless relative roughness. Note that you will not be calling FrFact directly. It is called by CalCS.

To use these codes, simply call the routines wherever you need a value for what they calculate in the same manner as the example above. It is probably best if you use a new cell each time you use the routine, but you do not have to – you are on your own to debug your code if you do. For the CalCS routine, note that a higher number of Intervals increases the accuracy of your calculations, but the code will run slower. Anything over 30 is likely wasted effort. This code assumes that there are no gas impurities. Modifying it to include the impurities is not difficult.