CHAPTER 1 A TYPICAL SCENARIO FOR RUNNING THE ASCEND SYSTEM

The ASCEND system is a modeling environment. We have designed it to allow modelers to pose, debug and solve or optimize models described by up to of the order of a hundred thousand nonlinear algebraic equations on a conventional UNIX workstation or PC running Windows NT4 or Windows 95. The ASCEND system comprises three major parts: the ASCEND modeling language for posing models, the ASCEND interactive environment to allow users to compile, debug and execute models, and a suite of solvers and optimizers.

Detailed information on modeling appears in Chapter 15, Chapter 16, Chapter 17, and the Howto-ASCEND documents. The primary aim of this book is to describe the graphic user interface of ASCEND (a moving target if ever there was one) and to serve as a language reference (Chapter 19). You would typically proceed as follows to use the ASCEND GUI for modeling.

1. Using your favorite text editor (e.g., xemacs[1]), you will create a model of the problem you wish to solve in the ASCEND modeling language. ASCEND models are type definitions. Each model typically includes a declaration of the parts from which it is constructed, including variables, instances of previously defined types and arrays of any of these. Each model also includes the equations it adds to the model definition over and above those equations that its parts will provide. Finally if you abide by our advice on model writing, you will also write three or four methods that you will later run interactively on the compiled model instance to prepare it to be solved.

   If the model is particularly complex, you will probably create your model using types defined earlier by yourself and others. For chemical process flowsheet models, we provide a library of types. We also provide a file that contains most of the types of variables and constants anyone would use to construct a model.

2. Start up the interactive ASCEND user interface by typing

[1] Xemacs is a very powerful text editor which is widely used on UNIX workstations. It is available for free for both UNIX workstations and PCs through the WWW (search on xemacs).

‘ascend’ on the Unix shell command line or double clicking on the ASCEND icon on a PC. A number of different tool sets, each represented by a special window, open up on the screen. The one you will likely focus on at first is the SCRIPT window. In this window under Tools, you will open whichever tool window you want to work with at first, probably the LIBRARY tool set which provides tools to load text files containing ASCEND models. You will likely move and/or resize these windows.

3. Using a tool in the LIBRARY tool set, you will load the files containing the previously defined types of which your model makes use. You will open last the file containing the model you have just written. Unless you are incredibly skilled and/or lucky, you will see several error messages indicating that you have not correctly posed your model as the system attempts to load your new file. Moving back to your favorite editor you will correct syntax errors discovered by this file loading process, attempt again to load, make more corrections, etc.
4. Once your new model description can pass the loading process without errors, you will compile a simulation for it. Again there could be errors.
5. You will export this compiled simulation to the BROWSER tool set so you can look at the model, examining all its parts. If there were compiler errors, you may use tools in this tool set to aid you to find exactly what you have done wrong in posing the model.

   For example, if it is a particularly complex model, you will methodically examine it to see if you have configured it as you wanted.
6. Once you are through inspecting the model and have removed all the errors that of which you are aware, you will prepare the model to be solved. This you will do by asking the system to execute the methods you should have written to go with your model description. If you abide by the style of modeling that we strongly advocate, your model will have these methods attached to it -- written before your first atttempt to compile it. These methods will be for setting initial values for the variables, for scaling the variables, and for setting the “fixed” flags for a sufficient number of variables to make the model instance well-posed. To be well-posed means a number of things, among them that the model has the same number of variables to be calculated as equations available for calculating them.
7. You will next export the model to the SOLVER tool set. When importing a model, the SOLVER tool set analyzes the model to discover how many variables and equations are in its description.

If it is not an optimization problem, the SOLVER looks to see if it is well-posed and, if not, will issue warning messages and open up an interactive tool provided to aid you to make it well-posed right then and there. What you learn while using this tool you will likely encode right away into the model description so the next time you compile this model, it will become well-posed without this interactive step.

8. You can interactively choose among the available solvers and will most likely choose our nonlinear equation solving solver. With fingers crossed, you will ask the solver to start solving.

9. Whether or not it solves successfully, you will likely return to the BROWSER to inspect the results as you can view the value for every variable and equation residual in the model using the BROWSER. If the solving process fails, you can select tools both in the BROWSER and the SOLVER to look for the likely problem. For example, you may have posed your problem and its initial conditions such that the solution is out of bounds. A tool will tell you if the SOLVER has driven any of the variables in the model to their bounds. Another will tell you if some of your variables are poorly scaled. Yet another will investigate the model to see it if is locally singular, and if it is, that tool will report to you exactly which equations (by name) have given it reason to believe that to be so. (In the near future, this tool will also tell you that you should change what you are fixing and what you are calculating to remove this singularity, if such a move would prove useful. It will give you a list of variables from which to choose for each of these trades.)

10. You may wish to see the output in units different from those currently used. Opening the UNITS tools set will allow you to change wholesale from SI to American Engineering and/or to change individual units such as those for pressure from bars to atm.

11. You may have opened the SCRIPT tool set before loading the model files. Before doing all the above steps within ASCEND, you may then have activated a tool to record all the steps you will subsequently take to load, compile, initialize and solve the model. This tool will construct a script from the steps it sees you taking. You wlil likely then edit this script, for example to delete some of the missteps you have taken, and then save it. You may also pick out any of the steps in the script and execute them at any time rather than look for the tools in the tool set windows. You would use a script to aid you to repeat all the above steps quickly while you are debugging a model. You will also prepare a script to hand your model to someone else to execute. Indeed, your first

experience with ASCEND may be to run a script that someone else has provided so you can be sure to run your first example successfully.