

CHAPTER 1 STARTING POINTS

Our goal: The purpose of this chapter is to help you find out what you need to read first about ASCEND IV in order to accomplish some portion of your mathematical modeling tasks. Since there is no single “best order” to learn in for all people, we list the introductory documents and their “sound bytes” concisely, in the hope that this makes your search task less difficult. If ASCEND IV is new to you, work through the first three listed in sequence, then branch to the special topics you need most¹. Without further ado, your goals.

1.1 PRIMAL SUBJECTS

- Chapter 2 **Building and solving a small mathematical model** from a “simple” problem description of a water tank. This is basic mathematical modeling of a physical system. If you have never, ever used ASCEND IV, you should probably start here to build and solve a model.
- Chapter 3 **Making any model easier to share with others** by adding basic methods, scripts, and model interfaces.
- Chapter 4 **Reusing a model for plotting and case studies** with an introduction to type refinement and inheritance. Defining and executing a case study to generate data and plots which indicate how your mathematical model responds to alternative input values.
- Chapter 5 **Managing modeling project files** with REQUIRE and PROVIDE. ASCEND will automatically load the other type definition files you need when working on a model if you follow some simple rules.

1. If you last used ASCEND as ASCEND III running on an HP or Apollo, ASCEND IV is new to you.

- Chapter 6 **Defining a plot which gathers scattered data** from your models into a plt_plot that can be viewed from the Browser window.
- *howto-specify*
(Art,Ben, in progress) **Defining a “square” or “well-posed” problem** when your model gets big. Writing a “specify” method is the only reliable way to go, and even this is not simple unless you plan ahead. Degrees of freedom can be tricky.
- Chapter 7 **Defining new types of variables** or constants when the standard library does not have what you want.
- Chapter 8 **Entering correlation equations with units** and how we support degrees Fahrenheit.
- Chapter 9 **Defining new units of measure** based on SI or other existing units.
- *howto-library1*
(NOTES, check methods, etc) **Getting it right the first time.** Modeling reliably in teams requires communicating all problem aspects including the goals to be met, the mathematical problem to be solved, the solution process, and the testing criteria that define an acceptable solution. You can do all these in ASCEND IV.
- Chapter 10 **Making basic models easy to use later** by adding METHODS. Defining more standard methods and your own methods so you do not have to remember how you made the model work yesterday, last week, last year, or in your last incarnation. It’s almost automatic.

1.2 ENGINEERING SUBJECTS

- Chapter 11 **Defining a chemical mixture and physical property calculations** for use in process simulation. Equilibrium thermodynamics, phases, species, and all that jazz. Adding species and correlations to the database.
- *howto-column1*
(Art, in progress) **Defining a steady-state distillation column** in a flowsheet using the column library that comes with ASCEND IV.
- *howto-reactor*
(Duncan, in progress) **Defining a chemical reactor model** in a flowsheet. Not a task for the faint of heart, but probably far easier than defining a new reactor in almost any commercial simulator.
- Chapter 12 **Defining a simple dynamic model (initial value problem)** and watching it respond. Water level in a tank.

- *howto-dynamic2*
(Duncan, in progress) **Defining a complex dynamic model** using dynamic libraries.
Dynamic vapor-liquid flash tank.
- *howto-column2*
(Duncan, in progress) **Simulating a dynamic distillation column** in a flowsheet using
ASCEND.
- *howto-control*
(Duncan, in progress) **Controlling dynamic systems, disturbances, and all those pesky
graphing tools** using the Script window and Tcl.
- Chapter 13 **Writing a conditional model** where which equations apply is
determined by variable values or boundary expressions.
- Chapter 14 (Ben, in
progress) **Defining a dynamic model with end-point conditions** (boundary
value problem) using our collocation (bvp) library.

