

Tools and Mindset Needed for these Modules

“Tell me and I’ll forget; show me and I may remember; involve me and I’ll understand.

-Reputably a Chinese proverb

Introduction

These modules are different from textbooks and cookbooks. Rather than give you facts about computing or step-by-step directions for using various tools, we attempt to present you with a scientific problem to solve and then lead you through some of the steps that might be followed in solving that problem with the help of computation. The goal is to learn some of the computational tools and mathematical models used in practical science, as well as the process itself that a scientist may use in working towards the solution to a problem. Put together this is what we call *computational scientific thinking*. So you see it’s not just learning how to run some piece of software, but also understanding how that software goes about its business, how all the pieces fit into the scientific problem-solving process, and developing a mindset in which you feel comfortable using the computer to explore science.

As is often true in science, there are multiple tools that may be used to find the solution to a problem, and the “best” tool may be the one you know best or the one you have available. For this reason we provide examples of the use of three computational tools that typify those used in science:

1. **Python**, and specifically **Vpython** (Python combined with the **Visual** package), is a modern compiled programming language that is unusually easy to learn, very powerful, free, and widely used by both beginners and professionals. We prefer having you use it for these projects since a compiled language helps make the mathematical underpinnings of the model clearer, since its ease of modification encourages experimentation, and since Python can be extended all the way to serious research-level problems.
2. **Excel**, the spreadsheet component of Microsoft Office. Because many people are familiar with Excel, this is often a good place to start. Nevertheless, we recommend that at some point in your development you also try Python and Vensim, but particularly Python. **Apache OpenOffice Calc**, the spreadsheet component of [Apache OpenOffice](#), is free and is just as good for our applications. Although we have not tested them all, our examples should work with OpenOffice Calc.
3. **Vensim PLE**, a software package designed for modeling quantities that change over time. It uses visual programming in which the user constructs a graphical representation of a model with arrows and blocks, and thus hides the programming and mathematics underneath the graphics. **Vensim Personal Learning Edition (PLE)** is available free online for educational use at: <http://www.vensim.com/freedownload.html> .