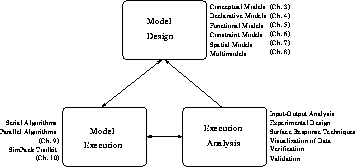
WHAT IS SIMULATION?

Computer simulation is the discipline of designing a model of an actual or theoretical physical system, executing the model on a digital computer, and analyzing the execution output. Simulation embodies the principle of ``learning by doing'' --- to learn about the system we must first build a model of some sort and then operate the model. The use of simulation is an activity that is as natural as a child who *role plays*. Children understand the world around them by simulating (with toys and figurines) most of their interactions with other people, animals and objects. As adults, we lose some of this childlike behavior but recapture it later on through computer simulation. To understand reality and all of its complexity, we must build artificial objects and dynamically act out roles with them. Computer simulation is the electronic equivalent of this type of role playing and it serves to drive synthetic environments and virtual worlds. Within the overall task of simulation, there are three primary sub-fields: model design, model execution and model analysis (see Fig. [1](https://www.cise.ufl.edu/~fishwick/introsim/node1.html#fig1)). The chapter annotations in Fig. 1 relate to the author's recent simulation textbook (see the section READ MORE ABOUT IT) which focuses on the first two sub-fields.

     
Figure 1: Three Sub-Fields of Computer Simulation.

To simulate something physical, you will first need to create a *mathematical model* which represents that physical object. Models can take many forms including declarative, functional, constraint, spatial or multimodel. A multimodel is a model containing multiple integrated models each of which represents a level of granularity for the physical system. The next task, once a model has been developed, is to execute the model on a computer --- that is, you need to create a computer program which steps through time while updating the state and event variables in your mathematical model. There are many ways to ``step through time.'' You can, for instance, *leap* through time using *event scheduling* or you can employ small time increments using *time slicing*. You can also execute (i.e., simulate) the program on a massively parallel computer. This is called *parallel and distributed simulation*. For many large-scale models, this is the only feasible way of getting answers back in a reasonable amount of time.

Simulation of a system can be done at many different levels of fidelity so that whereas one reader will think of physics-based models and output, another may think of more abstract models which yield higher-level, less detailed output as in a queuing network. Models are designed to provide answers at a given abstraction level --- the more detailed the model, the more detailed the output. The kind of output you need will suggest the type of model you will employ. An example of graphical output from a physically-based model (generated using the program AERO is shown as a stereo pair of ``rigid bodies'' in Fig. [2](https://www.cise.ufl.edu/~fishwick/introsim/node1.html#stereo)). You can view this stereo pair without the use of external viewing aids, by diverging the eyes