Bio 595 **Computers in Biomedical Research** Fall 2003

Simulation methods

A model is any representation of a real system using words, diagrams, physical pieces, and/or mathematical equations. Virtually all of science deals with the creation, evaluation, and verification of models of nature. Examples are almost everything you “learned” in school: the Bohr model of the atom, the Watson-Crick DNA helix, the gene (beads on a string) ,an enzyme (the “lock and key” model), and metabolic pathways (e.g., the Krebs cycle is a conceptual model).

Mathematical models:

- can be simple (with one dependent variable in an explicit equation)
- or, can be quite complex, involving simultaneous equations with several mutually dependent variables. The latter are said to be “multicomponent” models.

There are two ways mathematical models are implemented: by theoretical derivation and simulation; or by empirical derivation through statistical analysis.

In doing a computer simulation using mathematical models, you can simulate data, then compare the real with the calculated results. In developing your model on the computer, use a graphical (rather than tabular) display to reduce the program’s results to an understandable form.

Sequence of steps in creating a model: (1) analyze the system you want to simulate, and determine basic components required for the model (it is helpful to draw a block diagram); (2) Define key variables in the system: express each variable as a function of other system variables and parameters; (3) derive the necessary equations either empirically or analytically; (4) program the equations into the computer, input the parameters, and run the program.

The value of a simulation is that it allows quantification of our understanding of a phenomenon, and formulation of hypotheses we can then proceed to confirm (or reject) by using the scientific method. Also, developing a math model of a phenomenon allows us to perfect our conceptual model of that phenomenon. Simulations also allow performance of "what if" types of experiments. Computer simulations are an important teaching tool. In business, "what if" capabilities are an important part of spreadsheet power, allowing changes to be made in a value to see what happens to the results.

Simple models: a single explicit equation can serve as an analog or model of a simple biological process: almost any biological process can be described by a response curve relating intensity or amount of response to that of the stimulus magnitude in an assumed cause-and-effect relationship.

Multicomponent models: are composed of clusters of submodels or modules which have multiple interdependencies. Each module can be developed independently and validated by comparing its output with that of the real system. A single component may consist of subcomponents, which in turn can consist of sub-subcomponents, etc: a hierarchy of structure. Often components and subcomponents are lumped together into sub-models which define the net effect on the system as a whole.

Empirical models are developed in the following manner: (1) first we obtain experimental data to see the actual response of the system to a given stimulus; (2) we then obtain a math formula which responds the same way; (3) we now use this equation to simulate the system’s performance under untested as well as experimentally tested conditions, to validate the model. These models don't depend on a theoretical understanding of the processes involved.

Analytical models: here we use a theoretical understanding of the phenomenon to develop an equation to predict the system’s behavior.

Both of these model types can be joined to yield final simulation.
To develop a simple empirical model, observe or record the actual cause (stimulus) and effect (response) relationship of a real-world system. Obtain these data as a y (output) vs. x (input) graph; next, reduce to equation of the form y = fn(x). Then, find appropriate parameters for coefficients and exponents in the model system, either by trial and error, or by curve fitting.

To begin to learn simulation, you should be able to:

- program the computer to simulate data from model equations;
- know basic curve shapes fundamental to modeling;
- know how to fit curves to data using commercial software like CricketGraph;
- know how to perform simple numerical integration.

Example of a simple model: population growth (e.g., yeast growth). Population growth is a function of population density: the more yeast or bacteria we have, the faster they grow (assuming that there is adequate nutrient):

growth = fn (population density), or
G = f (N)

If we assume that growth is in direct proportionality to the number of cells, then the growth rate equation can take the form

$$G = kN$$

The curve for this equation is a straight line with slope = k. This equation is of limited interest: what we really want is the population size at any time during the growth process, rather than the growth rate. To get this we must rewrite the equation as a differential equation in which growth rate is appropriately expressed as the derivative of population size N as a function of time t, or dN/dt. So the equation now takes the form

$$dN/dt = k\,N$$

Solve this by integrating (that is, we're using the equation for the slope of the growth curve to derive the equation for the growth curve itself). We collect all terms dealing with the dependent variable N on the left side of the equation, and the independent variable and constants on the right side to get

$$\int_{N_o}^{N_t} \frac{dN}{N} = \int_{o}^{t} kdt$$

The result of integrating without limits is

$$\ln N = k\,t + c$$

To integrate between limits, we take the integral at the upper limit minus the integral at the lower limit, to get

$$\ln N_t - \ln N_o = k * t - k * 0$$

Since the difference between logs equals the log of a quotient, we get

$$\ln (N_t/N_o) = k * t$$

We take exponents of both sides of the equation to get

$$N_t/N_o = e^{k\,t}$$

or:

$$N_t = N_o * e^{k\,t}$$

This equation is the classic growth curve: it expresses population N_t at time t as function of initial population N_o ($t = 0$) and proportionality constant k (growth rate constant). To model this on the computer, try values, for example, of $k = .02$/hour and $N_o = 2$/ml; graph N up to $t = 50$ hours.

<u>Here are some primary equation types</u> (see accompanying plots):

1. Straight line: $y = Ax + B$ (not graphed on accompanying page)

2. Exponential: $y = A\,e^{n\,x}$
    1: A=1, n=0.05 2: A=1, n=0.1
    3: A=100, n=-0.05 4: A=100, n=-0.01

3. Power function: $y = A\,x^{n}$
    1: A=1.5, n=0.7 2: A=100, n=-0.5
    3: A=100, n=-0.2 4: A=0.7, n=1.5

4. Hyperbola: $y = A\,x / (B + x)$
    1: A=100, B=50 2: A=100, B=20

5. Exponential saturation: $y = A\,(1 - e^{n\,x})$
    3: A=100, n=-0.1 4: A=100, n=-0.03

6. Sigmoid: $y = A / (1 + B\,x^{n})$
    1. A=100, n=-3, B=500,000 2: A=100, n=-3, B=100,000
    3: A=100, n=-3, B=5000

7. Exponential sigmoid: $y = A / (1 + B\,e^{n\,x})$
    1: A=100, B=50, n=-0.03 2: A=100, B=50, n=-0.05
    3: A=100, B=50, n=-0.1 4: A=100, B=50, n=-0.2

8. Modified inverse: $y = A / (B + x)$
    1: A=2000, B=40 2: A=2000, B=20 3: A=2000, B=0

9. Modified power function: $y = A^{x\,n} + B$
    1: A=0.00001, B=0.25, n=3 2: A=0.001, B=0.25, n=3
    3: A=-0.0001, B=100, n=3 4: A=-0.00001, B=100, n=3

10. Maxima function: $y = A\,x\,e^{n\,x}$
    1: A=2, n=-0.02 2: A=2.5, n=-0.01
    3: A=7.5, n=-0.03 4: A=15, n=-0.06

<u>Curves resulting from compound equations</u>: curves showing a maximum or minimum reflect two or more forces competing with each other. For example, the product of equations (6) and (9) above would yield:

$$y = A/(1+B\,x^{n}) * (C\,x^{m} + D)$$

The product of equations (2) and (7) yield:

$$y = e^{n\,x} * (1 - A\,e^{m\,x}) \quad \text{where } m > n$$

To get a minimum, the sum of two equations for exponential growth and exponential decay would yield:

$$y = B\,e^{-n\,x} + A\,e^{m\,x}$$

<u>Fitting polynomials</u> to experimental data: these functions have little theoretical significance and should be used sparingly. It's easy (for a commercial program like CricketGraph) to fit data to a polynomial, but using this instead of one of the basic curve types may lead to

missing an opportunity to begin to understand the underlying basis for a relationship between two variables.