

## NUMERICAL METHODS: SOME SIMULATION ALGORITHMS

### Quotes of the Day

"It is the mark of an instructed mind to rest satisfied with the degree of precision which the nature of the subject permits and not to seek an exactness where only an approximation of the truth is possible."

--- Aristotle

## I. BACKGROUND

### (1) Why Numerical Methods?

- Numerical methods are needed for solving complex differential equations used to model physical, biological, and socioeconomic systems even though some of them (but few) may be analytically solvable.
- Numerical methods are imperative for differential equations that can not be solved analytically.

### (2) A Theoretical Basis for Numerical Methods

According to the Taylor Theorem, a function  $Y(x)$  can be expanded, about a point  $x = x_m$ , into a series, which is known as the Taylor series:

$$Y(x) = Y_m + Y'_m(x - x_m) + Y''_m \frac{(x - x_m)^2}{2!} + Y'''_m \frac{(x - x_m)^3}{3!} + \dots + Y_m^{(n)} \frac{(x - x_m)^n}{n!} + Y_{\xi}^{(n+1)} \frac{(x - x_m)^{n+1}}{(n+1)!}$$

where  $Y_m = Y(x_m)$ ,  $Y'_m, Y''_m, Y'''_m, \dots, Y_m^{(n)}$  are the first, second, third, ..., and nth derivative of  $Y(x)$  evaluated at  $x = x_m$ , and  $x_m < \xi < x$ .

The Taylor series expansion provides a theoretical basis for evaluating and comparing commonly used numerical methods. We can use the series to approximate the solution  $Y(x)$  at  $x = x_m + \Delta x$  simply by replacing  $x$  by  $x_m + \Delta x$  in the Taylor series. In the context of differential equations describing temporal dynamics, we use  $t$  for  $x$  and thus have:

$$Y_{t+\Delta t} = Y_t + Y'_t(\Delta t) + Y''_t \frac{(\Delta t)^2}{2!} + Y'''_t \frac{(\Delta t)^3}{3!} + \dots + Y_t^{(n)} \frac{(\Delta t)^n}{n!}$$

We can select the first few terms in the Taylor series to approximate the solution of a function, but this will result in truncation error. Apparently, the more terms a procedure uses, the more accurate the approximation is.

#### (1) First-order approximation:

$$Y_{t+\Delta t} = Y_t + Y'_t(\Delta t) + \epsilon_T^1,$$

$$\epsilon_T^1 = K(\Delta t)^2$$

where  $\epsilon_T^1$  is the truncation error, and K is a constant. (Note: Compare the above equation with the definition of the first-order derivative!)

(2) Second-order approximation:

$$Y_{t+\Delta t} = Y_t + Y'_t(\Delta t) + Y''_t \frac{(\Delta t)^2}{2!} + \epsilon_T^2$$

where the truncation error is reduced to  $\epsilon_T^2 = K(\Delta t)^3$ .

(3) kth-order approximation:

$$Y_{t+\Delta t} = Y_t + Y'_t(\Delta t) + Y''_t \frac{(\Delta t)^2}{2!} + Y'''_t \frac{(\Delta t)^3}{3!} + \dots + Y^{(k)}_t \frac{(\Delta t)^k}{k!} + \epsilon_T^k$$

in which the truncation error is reduced by several orders of magnitude as compared to that in the first-order approximation.

$$\epsilon_T^k = K(\Delta t)^{k+1}$$

## II. EULER'S AND RUNGE-KUTTA METHODS

There exist many different numerical methods for solving differential equations (see Dorn and McCracken 1972, Press et al. 1992). Two types of numerical methods can be distinguished: (1) one-step, self-starting methods (e.g., Euler's and modified Euler's methods, Runge-Kutta methods) and (2) Multistep, non-self-starting methods (e.g., predictor-corrector methods).

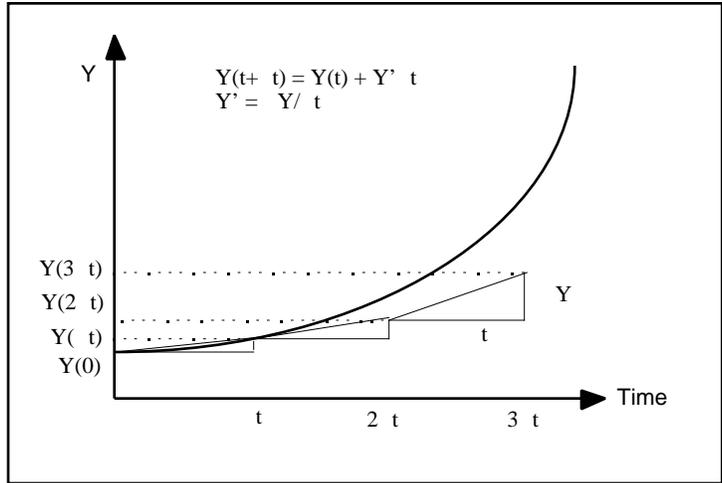
Amongst the simplest and most commonly used are Euler's method and Runge-Kutta methods. These numerical techniques are all one time-step methods; that is, only information at one time interval is used to estimate change in state variables. They agree with the Taylor series through terms in  $(\Delta t)^k$ , where k is the order of the method. For example, k is 2 in the 2nd-order Runge-Kutta methods and 4 in the 4th-order Runge-Kutta method. Thus, Euler's method may be considered as a 1st-order Runge-Kutta method.

### 1. Euler's Method

Euler's method is one of the oldest and best known numerical methods for integrating differential equations. It simply uses the estimate of the rate of change ( $Y'$ ) at time t to calculate the change in the state variable over the time interval  $(\Delta t)$ , assuming the rate of change is constant over  $\Delta t$ . The formula for Euler's method is simply

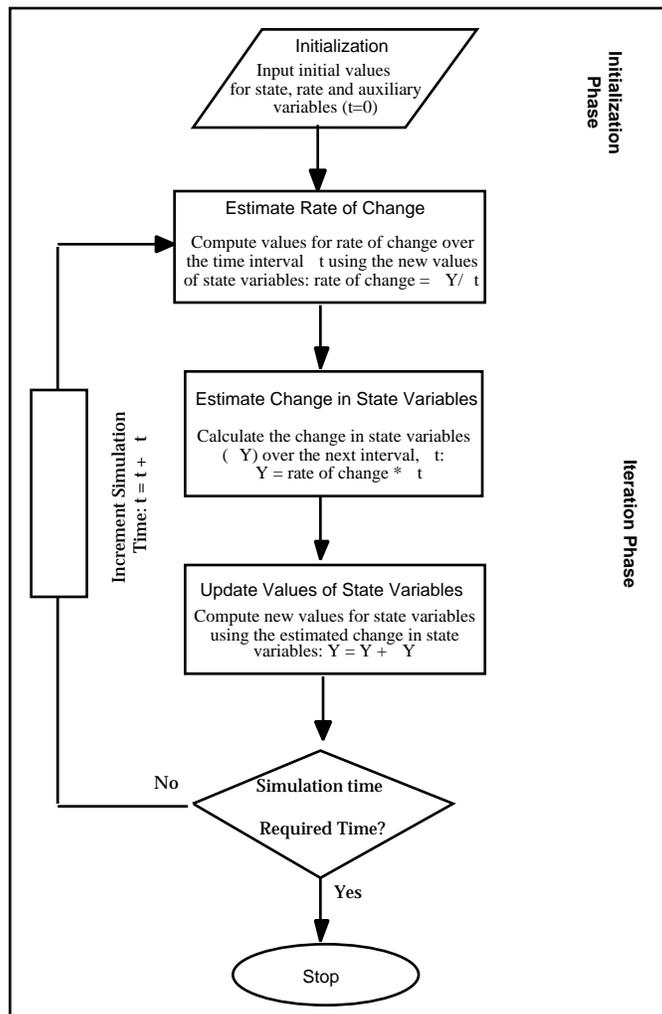
$$Y_{t+\Delta t} = Y_t + Y'_t(\Delta t)$$

Although it is extremely simple and computationally efficient, Euler's method may introduce significant truncation and integration errors especially when  $\Delta t$  is relatively large. Besides, it is often unstable, meaning that small roundoff and truncation errors may propagate as simulation time progresses. This error propagation may lead to artifactual behaviors that do not really come from the differential equations.



A geometrical representation of Euler's method: a series of Euler approximations (straight lines) to a true solution (curved line) over  $\Delta t$  solution intervals (Haefner 1996)

A general procedure of numerical approximation of solutions to differential equations



## 2. Runge-Kutta Methods

To reduce truncation and integration errors, Runge-Kutta methods require additional evaluations within each iteration. Two most commonly used Runge-Kutta methods are the second- and fourth-order Runge-Kutta Methods. As compared to Euler's method, the 2nd-order Runge-Kutta method requires the evaluation of the rate of change at two points within a time interval, whereas the 4th-order requires the evaluation at four points within a time interval. The accuracy of numerical solutions is substantially improved at the expense of computational time.

Let  $Y = F(t)$  and  $\frac{dY}{dt} = f(t, Y)$ , a commonly used formula for the 2nd-order Runge-Kutta is

$$Y_{t+\Delta t} = Y_t + \frac{1}{2} [f(t, Y_t) + f(t + \Delta t, Y_{t+\Delta t})],$$

which simply means that  $Y$  is calculated based on an average of the two estimates of the rate of change at two points,  $t$  and  $t + \Delta t$  (namely, the beginning and end of a time interval). The fourth-order Runge-Kutta method uses a weighted average of 4 estimates of the rate of change to calculate the value of the state variable over each  $\Delta t$ . A widely used 4th-order Runge-Kutta formulation is as follows:

$$Y_{t+\Delta t} = Y_t + \frac{1}{6} (F_1 + F_2 + F_3 + F_4)$$

where

$$F_1 = f(t, Y_t) \Delta t$$

$$F_2 = f\left(t + \frac{\Delta t}{2}, Y_t + \frac{1}{2} F_1\right) \Delta t$$

$$F_3 = f\left(t + \frac{\Delta t}{2}, Y_t + \frac{1}{2} F_2\right) \Delta t$$

$$F_4 = f(t + \Delta t, Y_t + F_3) \Delta t$$

## 3. Accuracy Comparison of Euler's and Runge-Kutta Methods

Runge-Kutta methods are usually much more accurate than Euler's method, but higher-order Runge-Kutta techniques may become rather time-consuming when a small  $\Delta t$  is used. However, with larger time steps and thus with about the equivalent amount of computational time, integration errors produced by Runge-Kutta methods are usually significantly smaller than those by Euler's method. These can be demonstrated through the following example on simulating the coffee cooling process (model demo).

### III. NOTES ON SIMULATION ALGORITHMS IN STELLA

STELLA includes three numerical techniques: Euler's method, and the second- and fourth-order Runge-Kutta methods. Here are some very important points (cf. Technical Documentation of STELLA), which one needs to know when using STELLA for any serious, scientific purposes (not just for fun!).

1. For modeling discrete dynamics or discrete objects, use Euler's method only. This includes situations in which you use any of STELLA's discrete objects (queues, conveyors, and ovens), or built-in functions to generate integer values (e.g., IF-THEN-ELSE logic to set 0-1 flags, INT, ROUND, SWITCH, etc). STELLA can be used for simulating discrete systems, but in this case  $\Delta t$  is usually set to 1 and only Euler's method has the appropriate mathematical formulation. Runge-Kutta methods are designed for continuous systems, and inappropriate for simulating discrete dynamics.
2. For modeling continuous systems, the Runge-Kutta methods are generally preferable. For those that exhibit oscillations the Runge-Kutta methods must be used. Runge-Kutta methods are not only more accurate, but also are more stable. On the other hand, error propagation in Euler's method may turn a sustained oscillation into an ever expanding oscillation!
3. The aspects of accuracy and simulation time should be simultaneously considered in choosing a proper  $\Delta t$ .

### REFERENCES

- Dorn, W. S. and D. D. McCracken. 1972. Numerical Methods with Fortran IV Case Studies. Wiley, New York.
- Press, W. H., W.A. Teukolsky, W. T. Vetterling, and B. P. Flannery. 1992. Numerical Recipes in C: The Art of Scientific Computing. 2nd Ed. Cambridge University Press, Cambridge. 994 pp.