9. Ordinary Differential Equations.

NUMERICAL ANALYSIS. Prof. Y. Nishidate (323-B, nisidate@u-aizu.ac.jp)
http://web-int.u-aizu.ac.jp/~nisidate/na/

Initial Value Problem

Let us consider methods of finding the numerical solution of an initial value problem for the first order differential equation:

$$y'(x) = f(x, y), \qquad y(a) = y_a$$

The above equation is an equation with independent variable x and dependent variable $y = y(x)$. We assume f is a "nice" function of the variables x and y, which provides the $y'(x)$ when evaluated at the point (x, y). We are going to find the solution $y(x)$ on some interval $[a, b)$ where we know the initial value $y(a) = y_a$.

Euler's Method

The simplest method is derived from the Taylor series expansion

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \tfrac{h^2}{2}y''(\xi_n)$$
$$h = x_{n+1} - x_n \qquad x_n < \xi_n < x_n + h$$

The algorithm of the *Euler's method* is as follows:

$$y_{n+1} = y_n + hf(x_n, y_n)$$

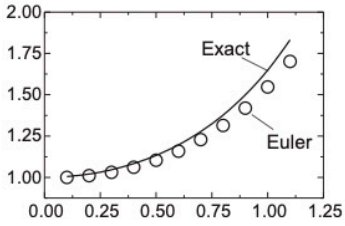
From Taylor's expansion it is evident that the local truncation error of the Euler's method is

$$\varepsilon_{n+1} = \frac{h^2}{2}y''(\xi_n) = O(h^2)$$

The global error is the sum of the local errors

$$E(h) = \sum \varepsilon_{n+1} = NCh^2 = \frac{b-a}{h}Ch^2 = O(h)$$

A simple look at Figure below shows the main drawback of the Euler's method. When the true solution goes upward, then the numerical solution always lags below because it uses the slope from the past.



Predictor-Corrector Method

A *predictor-corrector* method consists of two stages: 1) *predict* solution and then examine slope at that point; 2) using this slope, go back and make a second more *correct* guess.

Both predictor and corrector should have local truncation errors of the same order. Let us use the midpoint formula to *predict*

$$y_{n+1}^{(pre)} = y_{n-1} + 2hy'_n$$

Computing the slope y'_{n+1} we then average the two end slopes as a more reasonable choice and compute the *correct* value by the trapezoid formula

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}\left(y'^{(pre)}_{n+1} + y'_n\right) \\ &= y_n + \frac{h}{2}\left\{f(x_{n+1}, y^{(pre)}_{n+1}) + f(x_n, y_n)\right\} \end{aligned}$$

The local truncation error for the predictor is

$$E_p(h) = \frac{h^3}{3}y'''(\xi_p)$$

while the local truncation error for the corrector is

$$E_c(h) = -\frac{h^3}{12}y'''(\xi_c)$$

The first question that occurs is how to start the predictor-corrector procedure. Our first prediction requires knowing one previous point in addition to current slope. One answer is to use the differential equation and its derivatives to form a Taylor expansion

$$y_1 = y_0 + hy'_0 + \frac{h^2}{2!}y''_0 + \frac{h^3}{3!}y'''_0 + \ldots$$

about the starting point and from the series to compute the first point. Since the error term in each step depends on the third derivative it is desirable to use Taylor expansion up to h^3 term.

Runge-Kutta Methods

To achieve a higher order of accuracy for Taylor series we will have to find various derivatives $y''(x)$, $y'''(x)$, ... This is difficult and often impossible task. Fortunately we can replace the derivatives by evaluations of $f(x, y)$ at intermediate points so as to achieve the same desired accuracy. Methods derives in this way are called *Runge-Kutta methods*. Our first task is to derive a Runge-Kutta method of order two.

Second order Runge-Kutta method. The basic idea is to replace the approximation

$$\begin{aligned} &y(x+h) = y(x) + hy'(x) + \tfrac{h^2}{2}y''(x) + O(h^3) \\ &= y(x) + hf(x,y) + \tfrac{h^2}{2}(f_x(x,y) + f_y(x,y)f(x,y)) + O\left(h^3\right) \end{aligned}$$

by

$$y(x+h) = y(x) + w_1k_1 + w_2k_2$$

where $k_1 = hf(x, y)$, $k_2 = hf(x + \alpha h, y + \beta k_1)$ and w_1, w_2, α and β are chosen to preserve the order of approximation. The motivation is that k_1 and k_2 are approximations to Δy. Expanding k_2 about the point (x, y) we have

$$k_2 = h(f(x,y) + \alpha h f_x(x,y) + \beta k_1 f_y(x,y)) + O(h^2)$$

so that

$$\begin{aligned} &y(x+h) = y(x) + w_1hf(x,y) + w_2h[f(x,y) + \alpha h f_x(x,y) \\ &+\beta h f(x,y) f_y(x,y)] + O(h^3) \end{aligned}$$

Comparing two expressions for $y(x + h)$ it is possible to see that they will be identical if

$$w_1 + w_2 = 1, \qquad w_2\alpha = \frac{1}{2}, \qquad w_2\beta = \frac{1}{2}$$

This gives three equations for the four unknowns. Usually it is best to take the weights w_1 and w_2 equal. Thus we choose $w_1 = w_2 = 1/2$ and $\alpha = \beta = 1$. The Runge-Kutta algorithm of the second order is as follows:

$$\begin{aligned} &y_{n+1} = y_n + \tfrac{1}{2}k_1 + \tfrac{1}{2}k_2 \\ &k_1 = hf(x_n, y_n), \qquad k_2 = hf(x_{n+1}, y_n + k_1) \end{aligned}$$

This algorithm has a local truncation error $O(h^3)$.

Fourth order Runge-Kutta method. The best known Runge-Kutta method is given below:

$$\begin{aligned} y_{n+1} &= y_n + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_1}{2}) \\ k_3 &= hf(x_n + \tfrac{h}{2}, y_n + \tfrac{k_2}{2}) \\ k_4 &= hf(x_{n+1}, y_n + k_3) \end{aligned}$$

This algorithm has a local truncation error $O(h^5)$.

Multistep Methods

A multistep method takes the advantage of the fact that there is additional information available about the function that was obtained at previous steps. The principle behind the multistep method is to utilize the past values of y and/or y' to construct a polynomial that approximates the derivative and extrapolate this into the next interval. The Adams method is a typical multistep method.

Adams method. To derive the relations for the Adams method, we write the differential equation $dy/dx = f(x, y)$ in the form

$$dy = f(x, y)dx$$

and integrate it between x_n and x_{n+1}

$$\int_{x_n}^{x_{n+1}} dy = y_{n+1} - y_n = \int_{x_n}^{x_{n+1}} f(x, y)dx$$

Suppose that we fit a second degree polynomial through the last three points (x_n, y_n), (x_{n-1}, y_{n-1}), (x_{n-2}, y_{n-2}). The quadratic approximation to the derivative function can be presented in the following form:

$$\begin{aligned} f(x, y) &= \tfrac{1}{2h^2}(f_n - 2f_{n-1} + f_{n-2})x^2 \\ &+ \tfrac{1}{2h}(3f_n - 4f_{n-1} + f_{n-2})x + f_n \end{aligned}$$

After integration we have the Adams formula to advance y:

$$y_{n+1} = y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2}) + O\left(h^4\right)$$

The major advantage of multistep methods is that we can usually obtain higher accuracy with less work than for single step methods. There are several disadvantages however. One is that multistep methods are not self-starting methods. In order to apply a multistep method, the information on several previous steps is required. In the beginning this information does not exist.

Higher-Order Equations

Most differential equations for physical problems are of higher order. For example,

$$m\frac{d^2x}{dt^2} + b\frac{dx}{dt} + kx = f(x, t)$$

represents a vibrating system in which a linear spring with spring constant k restores a displaced mass m against a resistance force whose resistance is b times the velocity. The function $f(x, t)$ is external force acting on the mass.

A higher-order differential equation can be reduced to a system of first order equations. By solving for the second derivative, we can usually express a second-order equation as

$$\frac{d^2x}{dt^2} = f(t, x, \frac{dx}{dt}), \quad x(t_0) = x_0, \quad x'(t_0) = x_0'$$

The initial value of the function and its first derivative should be specified. We convert this to a pair of first-order equations by defining the derivative as a second function. Then, since $d^2x/dt^2 = (d/dt)(dx/dt)$,

$$\begin{aligned} \tfrac{dx}{dt} &= y \qquad x(t_0) = x_0 \\ \tfrac{dy}{dt} &= f(t, x, y) \qquad y(t_0) = x_0' \end{aligned}$$

This pair of first-order differential equations is equivalent to the original second-order equation. Same approach can be used to transform a differential equation of any order into a system of first order differential equations. The mth-order differential equation

$$\begin{aligned} &y^{(m)} = f(x, y, y', ..., y^{(m-1)}) \\ &y(x_0) = A_1, \quad y'(x_0) = A_2, ..., \quad y^{(m-1)}(x_0) = A_n \end{aligned}$$

is converted into a system of m first-order differential equations by setting $y_1 = y$ and

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ &... \\ y_m' &= f(x, y_1, y_2, ..., y_m) \end{aligned}$$

with initial conditions

$$y_1(x_0) = A_1, \quad y_2(x_0) = A_2, ..., \quad y_m(x_0) = A_n$$

First Order Systems

A first order system of ordinary differential equations can be presented in the form

$$Y'(x) = F(x, Y), \quad Y(a) = Y_a$$

where Y and F are vectors containing m components. As before, we consider an independent variable x on the interval $[a, b]$. Instead of one dependent variable we now have m dependent variable functions $y_1(x)$, $y_2(x)$,...,$y_m(x)$, which are the components of the vector $Y(x)$. For each (x, Y) we assume that there are n real valued functions $f_1(x, Y)$, $f_2(x, Y)$,...,$f_m(x, Y)$. Vectors Y and F have the following appearance:

$$Y(x) = \left\{ \begin{array}{l} y_1(x) \\ ... \\ y_m(x) \end{array} \right\} \quad F(x, Y) = \left\{ \begin{array}{l} f_1(x, Y) \\ ... \\ f_m(x, Y) \end{array} \right\}$$

The methods and error analysis to solve systems of first order differential equations are analogous to the methods used for single differential equations.

Fourth order Runge-Kutta method for systems. In the case of a system of m equations we have the following adaptation of the fourth order Runge-Kutta method:

$$\begin{aligned} k_{1i} &= hf_i(x_n, y_{1n}, y_{2n}, ..., y_{mn}) \\ k_{2i} &= hf_i(x_n + \tfrac{h}{2}, y_{1n} + \tfrac{k_{11}}{2}, ..., y_{mn} + \tfrac{k_{1m}}{2}) \\ k_{3i} &= hf_i(x_n + \tfrac{h}{2}, y_{1n} + \tfrac{k_{21}}{2}, ..., y_{mn} + \tfrac{k_{2m}}{2}) \\ k_{4i} &= hf_i(x_n + h, y_{1n} + k_{31}, ..., y_{mn} + k_{3m}) \\ y_{in+1} &= y_{in} + \tfrac{1}{6}(k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i}) \end{aligned}$$

for each $i = 1, 2, ..., m$.

It should be noted that before we calculate k_{21}, for example, we must calculate all of the $k_{11}, k_{12}, ..., k_{1m}$.

Explicit relations for the coefficients of the Runge-Kutta method in the case of **two-equation system** are given below:

$$\begin{aligned} k_{11} &= hf_1(x_n, y_{1n}, y_{2n}) \\ k_{12} &= hf_2(x_n, y_{1n}, y_{2n}) \\ k_{21} &= hf_1(x_n + \tfrac{h}{2}, y_{1n} + \tfrac{k_{11}}{2}, y_{2n} + \tfrac{k_{12}}{2}) \\ k_{22} &= hf_2(x_n + \tfrac{h}{2}, y_{1n} + \tfrac{k_{11}}{2}, y_{2n} + \tfrac{k_{12}}{2}) \\ k_{31} &= hf_1(x_n + \tfrac{h}{2}, y_{1n} + \tfrac{k_{21}}{2}, y_{2n} + \tfrac{k_{22}}{2}) \\ k_{32} &= hf_2(x_n + \tfrac{h}{2}, y_{1n} + \tfrac{k_{21}}{2}, y_{2n} + \tfrac{k_{22}}{2}) \\ k_{41} &= hf_1(x_n + h, y_{1n} + k_{31}, y_{2n} + k_{32}) \\ k_{42} &= hf_2(x_n + h, y_{1n} + k_{31}, y_{2n} + k_{32}) \end{aligned}$$

Advancement of function values is performed as follows:

$$\begin{aligned} y_{1n+1} &= y_{1n} + \tfrac{1}{6}(k_{11} + 2k_{21} + 2k_{31} + k_{41}) \\ y_{2n+1} &= y_{2n} + \tfrac{1}{6}(k_{12} + 2k_{22} + 2k_{32} + k_{42}) \end{aligned}$$