Initial-Value Problems for Ordinary Differential Equations

NTNU

Tsung-Min Hwang

December 8, 2003

In this chapter, we discuss numerical methods for solving ordinary differential equations of initial-value problems (IVP) of the form

$$\begin{cases} y' = f(x, y), & x \in [a, b] \\ y(x_0) = y_0, \end{cases} \tag{1}$$

where y is a function of x, f is a function of y and x, x_0 is called the initial point, and y_0 the initial value. The numerical values of $y(x)$ on an interval containing x_0 are to be determined.

1 – Existence and Uniqueness of Solutions

Theorem 1 *If* $f(x,y)$ *is continuous in a region* Ω*, where*

$$\Omega = \{(x,y);\ |x-x_0| \le \alpha,\ |y-y_0| \le \beta\} \tag{2}$$

then the IVP (1) has a solution $y(x)$ *for* $|x-x_0| \le \min\{\alpha, \frac{\beta}{M}\}$*, where* $M = \max_{(x,y)\in\Omega} |f(x,y)|$.

1 – Existence and Uniqueness of Solutions

Theorem 1 *If $f(x,y)$ is continuous in a region Ω, where*

$$\Omega = \{(x,y);\ |x - x_0| \leq \alpha,\ |y - y_0| \leq \beta\} \tag{2}$$

then the IVP (1) has a solution $y(x)$ for $|x - x_0| \leq \min\{\alpha, \dfrac{\beta}{M}\}$, where $M = \max_{(x,y)\in\Omega} |f(x,y)|$.

Theorem 2 *If f and $\frac{\partial f}{\partial x}$ are continuous in Ω, then the IVP (1) has a unique solution in the interval $|x - x_0| \leq \min\{\alpha, \frac{\beta}{M}\}$.*

1 – Existence and Uniqueness of Solutions

Theorem 1 *If* $f(x,y)$ *is continuous in a region* Ω*, where*

$$\Omega = \{(x,y);\ |x-x_0| \le \alpha,\ |y-y_0| \le \beta\} \tag{2}$$

then the IVP (1) has a solution $y(x)$ *for* $|x-x_0| \le \min\{\alpha, \dfrac{\beta}{M}\}$*, where* $M = \max_{(x,y)\in\Omega} |f(x,y)|$.

Theorem 2 *If* f *and* $\frac{\partial f}{\partial x}$ *are continuous in* Ω*, then the IVP (1) has a unique solution in the interval* $|x-x_0| \le \min\{\alpha, \frac{\beta}{M}\}$.

Theorem 3 *If* f *is continuous in* $a \le x \le b$, $-\infty < y < \infty$ *and*

$$|f(x,y_1) - f(x,y_2)| \le L|y_1 - y_2|$$

for some positive constant L*, (that is,* f *is Lipschitz continuous in* y*), then IVP (1) has a unique solution in the interval* $[a,b]$.

☞ Numerical integration of ODEs :

Given

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0.$$

Find approximate solution at discrete values of x

$$x_j = x_0 + jh$$

where h is the "stepsize".

☞ Graphical interpretation

2 – Euler's Method

On of the simplest methods for solving the IVP (1) is the Euler's method.

2 – Euler's Method

On of the simplest methods for solving the IVP (1) is the Euler's method.

Subdivide $[a, b]$ into n subintervals of equal length h with mesh points $\{x_0, x_1, \ldots, x_n\}$ where

$$x_i = a + ih, \forall\, i = 0, 1, 2, \ldots, n.$$

2 – Euler's Method

On of the simplest methods for solving the IVP (1) is the Euler's method.

Subdivide $[a, b]$ into n subintervals of equal length h with mesh points $\{x_0, x_1, \ldots, x_n\}$ where

$$x_i = a + ih, \forall\, i = 0, 1, 2, \ldots, n.$$

Recall the Taylor's Theorem

$$\begin{aligned} y(x_{i+1}) &= y(x_i) + (x_{i+1} - x_i)y'(x_i) + \frac{(x_{i+1} - x_i)^2}{2} y''(\xi_i) \\ &= y(x_i) + hy'(x_i) + \frac{h^2}{2} y''(\xi_i) \\ &= y(x_i) + hf(x_i, y(x_i)) + \frac{h^2}{2} y''(\xi_i) \end{aligned} \tag{3}$$

for some $\xi_i \in [x_i, x_{i+1}]$.

We have the formulation of Euler's mehtod

$$\begin{aligned} x_{k+1} &= x_k + h = x_0 + (k+1)h \\ y_{k+1} &= y_k + hf(x_k, y_k) \end{aligned}$$

We have the formulation of Euler's mehtod

$$\begin{aligned} x_{k+1} &= x_k + h = x_0 + (k+1)h \\ y_{k+1} &= y_k + hf(x_k, y_k) \end{aligned}$$

On the other hand, from (3),

$$f(x_i, y(x_i)) = y'(x_i) = \frac{y(x_{i+1}) - y(x_i)}{h} + \frac{h}{2} y''(\xi_i)$$

We have the formulation of Euler's mehtod

$$\begin{aligned} x_{k+1} &= x_k + h = x_0 + (k+1)h \\ y_{k+1} &= y_k + hf(x_k, y_k) \end{aligned}$$

On the other hand, from (3),

$$f(x_i, y(x_i)) = y'(x_i) = \frac{y(x_{i+1}) - y(x_i)}{h} + \frac{h}{2} y''(\xi_i)$$

It implies that Euler's method is a first-order $(O(h))$ method.

We have the formulation of Euler's mehtod

$$\begin{aligned} x_{k+1} &= x_k + h = x_0 + (k+1)h \\ y_{k+1} &= y_k + hf(x_k, y_k) \end{aligned}$$

On the other hand, from (3),

$$f(x_i, y(x_i)) = y'(x_i) = \frac{y(x_{i+1}) - y(x_i)}{h} + \frac{h}{2} y''(\xi_i)$$

It implies that Euler's method is a first-order $(O(h))$ method.

The Improved Euler's method

$$\begin{aligned} x_{k+1} &= x_k + h = x_0 + (k+1)h \\ y^*_{k+1} &= y_k + hf(x_k, y_k) \\ y_{k+1} &= y_k + \frac{h}{2}\left[f(x_k, y_k) + f(x_{k+1}, y^*_{k+1})\right] \end{aligned} \tag{4}$$

Example 1 *Use Euler's method to integrate*

$$\frac{dy}{dx} = x - 2y, \quad y(0) = 1.$$

Example 1 *Use Euler's method to integrate*

$$\frac{dy}{dx} = x - 2y, \quad y(0) = 1.$$

The exact solution is

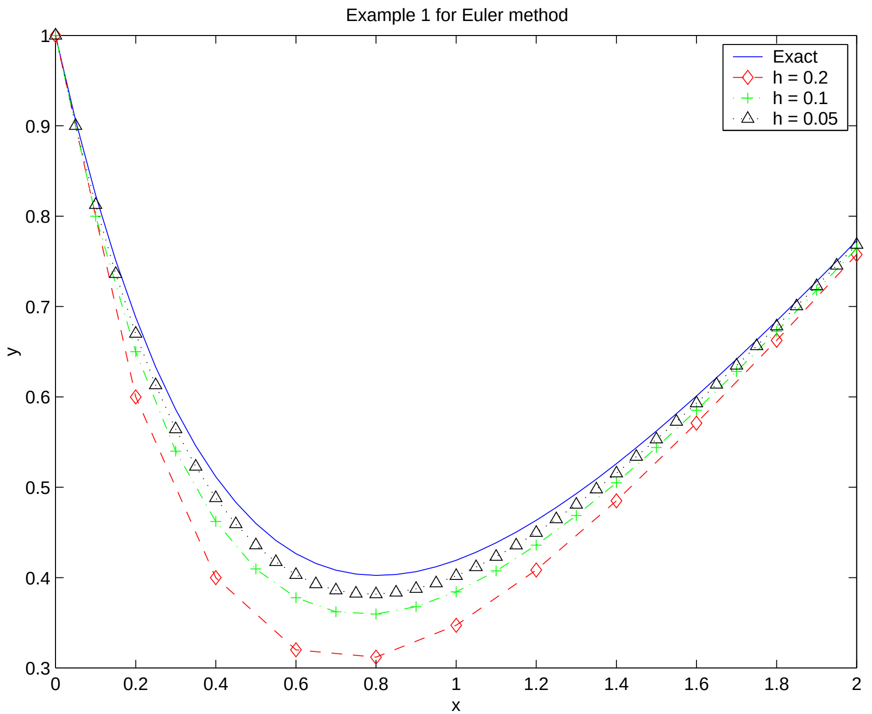
$$y = \frac{1}{4}\left[2x - 1 + 5e^{-2x}\right].$$

Example 1 *Use Euler's method to integrate*

$$\frac{dy}{dx} = x - 2y, \quad y(0) = 1.$$

The exact solution is

$$y = \frac{1}{4}\left[2x - 1 + 5e^{-2x}\right].$$



3 – Runge-Kutta Methods

One of the most important methods for solving the IVP (1) is the Runge-Kutta method.

3 – Runge-Kutta Methods

One of the most important methods for solving the IVP (1) is the Runge-Kutta method.

Recall the Taylor's Theorem

$$y(x+h) \quad = \quad y(x) + hy'(x) + \frac{h^2}{2}y'' + \frac{h^3}{2}y''' + \dots$$

3 – Runge-Kutta Methods

One of the most important methods for solving the IVP (1) is the Runge-Kutta method.

Recall the Taylor's Theorem

$$
\begin{aligned}
y(x+h) &= y(x) + hy'(x) + \frac{h^2}{2}y'' + \frac{h^3}{2}y''' + \dots \\
&= y(x) + hy'(x) + \frac{h^2}{2}y'' + O(h^3)
\end{aligned}
$$

By differentiating $y(x)$, we have

$$y' = f(x,y) \equiv f$$

By differentiating $y(x)$, we have

$$\begin{aligned} y' &= f(x,y) \equiv f \\ y'' &= \frac{d}{dx} f = f_x + f_y y' = f_x + f_y f \end{aligned}$$

By differentiating $y(x)$, we have

$$
\begin{aligned}
y' &= f(x,y) \equiv f \\
y'' &= \frac{d}{dx} f = f_x + f_y y' = f_x + f_y f \\
y''' &= f_{xx} + f_{xy} f + f \frac{d}{dx} f_y + f_y \frac{d}{dx} f
\end{aligned}
$$

By differentiating $y(x)$, we have

$$
\begin{aligned}
y' &= f(x,y) \equiv f \\
y'' &= \frac{d}{dx} f = f_x + f_y y' = f_x + f_y f \\
y''' &= f_{xx} + f_{xy} f + f \frac{d}{dx} f_y + f_y \frac{d}{dx} f \\
&= f_{xx} + f_{xy} f + f \left(\frac{f_y}{dy} \frac{dy}{dx} + f_{yx} \right) + f_y (f_x + f_y f)
\end{aligned}
$$

By differentiating $y(x)$, we have

$$
\begin{aligned}
y' &= f(x,y) \equiv f \\
y'' &= \frac{d}{dx} f = f_x + f_y y' = f_x + f_y f \\
y''' &= f_{xx} + f_{xy} f + f \frac{d}{dx} f_y + f_y \frac{d}{dx} f \\
&= f_{xx} + f_{xy} f + f\left(\frac{f_y}{dy}\frac{dy}{dx} + f_{yx}\right) + f_y(f_x + f_y f) \\
&= f_{xx} + f_{xy} f + f(f_{yx} + f_{yy} f) + f_y(f_x + f_y f)
\end{aligned}
$$

Then

$$y(x+h) = y + hy' + \frac{h^2}{2}y'' + O(h^3)$$

Then

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

Then

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

Then

$$\begin{aligned} y(x+h) &= y + hy' + \frac{h^2}{2}y'' + O(h^3) \\ &= y + hf + \frac{h^2}{2}(f_x + f_y f) + O(h^3) \\ &= y + \frac{h}{2}f + \frac{h}{2}f + \frac{h^2}{2}(f_x + f_y f) + O(h^3) \\ &= y + \frac{h}{2}f + \frac{h}{2}(f + hf_x + hf_y f) + O(h^3) \end{aligned}$$

Then

$$
\begin{aligned}
y(x+h) &= y + hy' + \frac{h^2}{2}y'' + O(h^3) \\
&= y + hf + \frac{h^2}{2}(f_x + f_y f) + O(h^3) \\
&= y + \frac{h}{2}f + \frac{h}{2}f + \frac{h^2}{2}(f_x + f_y f) + O(h^3) \\
&= y + \frac{h}{2}f + \frac{h}{2}(f + hf_x + hf_y f) + O(h^3)
\end{aligned}
$$

Apply Taylor's Theorem on f

$$
f(x+h, y+hf) = f(x,y) + hf_x + hff_y + O(h^2)
$$

Then

$$
\begin{aligned}
y(x+h) &= y + hy' + \frac{h^2}{2}y'' + O(h^3) \\
&= y + hf + \frac{h^2}{2}(f_x + f_y f) + O(h^3) \\
&= y + \frac{h}{2}f + \frac{h}{2}f + \frac{h^2}{2}(f_x + f_y f) + O(h^3) \\
&= y + \frac{h}{2}f + \frac{h}{2}(f + hf_x + hf_y f) + O(h^3)
\end{aligned}
$$

Apply Taylor's Theorem on f

$$
\begin{aligned}
& f(x+h, y+hf) = f(x,y) + hf_x + hff_y + O(h^2) \\
\Rightarrow \quad & f + hf_x + hff_y \approx f(x+h, y+hf)
\end{aligned}
$$

Then

$$
\begin{aligned}
y(x+h) &= y + hy' + \frac{h^2}{2}y'' + O(h^3) \\
&= y + hf + \frac{h^2}{2}(f_x + f_y f) + O(h^3) \\
&= y + \frac{h}{2}f + \frac{h}{2}f + \frac{h^2}{2}(f_x + f_y f) + O(h^3) \\
&= y + \frac{h}{2}f + \frac{h}{2}(f + hf_x + hf_y f) + O(h^3)
\end{aligned}
$$

Apply Taylor's Theorem on f

$$
\begin{aligned}
& f(x+h, y+hf) = f(x,y) + hf_x + hf f_y + O(h^2) \\
\Rightarrow\quad & f + hf_x + hf f_y \approx f(x+h, y+hf) \\
\Rightarrow\quad & y(x+h) = y + \frac{1}{2}hf + \frac{h}{2}f(x+h, y+hf) + O(h^3)
\end{aligned}
$$

Let

$$F_1 = hf(x, y), \qquad F_2 = hf(x + h, y + F_1)$$

Let

$$F_1 = hf(x, y), \qquad F_2 = hf(x + h, y + F_1)$$

Then

$$y(x + h) = y + \frac{1}{2}(F_1 + F_2).$$

Let

$$F_1 = hf(x, y), \qquad F_2 = hf(x + h, y + F_1)$$

Then

$$y(x + h) = y + \frac{1}{2}(F_1 + F_2).$$

This is called the second-order Runge-Kutta method.

Let

$$F_1 = hf(x,y), \qquad F_2 = hf(x+h, y+F_1)$$

Then

$$y(x+h) = y + \frac{1}{2}(F_1 + F_2).$$

This is called the second-order Runge-Kutta method. Two function evaluations are required at each step.

Let

$$F_1 = hf(x, y), \qquad F_2 = hf(x + h, y + F_1)$$

Then

$$y(x + h) = y + \frac{1}{2}(F_1 + F_2).$$

This is called the second-order Runge-Kutta method. Two function evaluations are required at each step.

Algorithm 1 *Algorithm for second-order Runge-Kutta mehtod :*

For $k = 0, 1, 2, \ldots$

$$x_{k+1} = x_k + h = x_0 + (k + 1)h$$

$$F_1 = hf(x_k, y_k)$$

$$F_2 = hf(x_{k+1}, y_k + F_1)$$

$$y_{k+1} = y_k + \tfrac{1}{2}(F_1 + F_2)$$

End for

General form of second-order Runge-Kutta method :

$$y(x+h) = y + \omega_1 hf(x,y) + \omega_2 hf(x+\alpha h, y+\beta hf) + O(h^3)$$

where $\omega_1, \omega_2, \alpha, \beta$ are constants to be defined, and

$$\omega_1 + \omega_2 = 1, \quad \omega_2 \alpha = \frac{1}{2}, \quad \omega_2 \beta = \frac{1}{2}$$

General form of second-order Runge-Kutta method :

$$y(x+h) = y + \omega_1 hf(x,y) + \omega_2 hf(x+\alpha h, y+\beta hf) + O(h^3)$$

where $\omega_1, \omega_2, \alpha, \beta$ are constants to be defined, and

$$\omega_1 + \omega_2 = 1, \quad \omega_2\alpha = \frac{1}{2}, \quad \omega_2\beta = \frac{1}{2}$$

By letting $\omega_1 = 0, \omega_2 = 1, \alpha = \beta = \frac{1}{2}$ leads to the modified Euler's mehtod.

General form of second-order Runge-Kutta method :

$$y(x+h) = y + \omega_1 hf(x,y) + \omega_2 hf(x+\alpha h, y+\beta hf) + O(h^3)$$

where $\omega_1, \omega_2, \alpha, \beta$ are constants to be defined, and

$$\omega_1 + \omega_2 = 1, \quad \omega_2\alpha = \frac{1}{2}, \quad \omega_2\beta = \frac{1}{2}$$

By letting $\omega_1 = 0, \omega_2 = 1, \alpha = \beta = \frac{1}{2}$ leads to the modified Euler's mehtod.

Fourth-Order Runge-kutta method

$$y(x+h) = y + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4) + O(h^5)$$

where

$$\begin{cases} F_1 = hf(x,y) \\ F_2 = hf(x+\frac{1}{2}h, y+\frac{1}{2}F_1) \\ F_3 = hf(x+\frac{1}{2}h, y+\frac{1}{2}F_2) \\ F_4 = hf(x+h, y+F_3) \end{cases}$$

Algorithm 2 *Algorithm for fourth-order Runge-Kutta mehtod :*

For $k = 0, 1, 2, \ldots$

$$x_{k+\frac{1}{2}} = x_k + \frac{1}{2}h$$

$$x_{k+1} = x_k + h = x_0 + (k+1)h$$

$$F_1 = hf(x_k, y_k)$$

$$F_2 = hf(x_{k+\frac{1}{2}}, y_k + \frac{1}{2}F_1)$$

$$F_3 = hf(x_{k+\frac{1}{2}}, y_k + \frac{1}{2}F_2)$$

$$F_4 = hf(x_{k+1}, y_k + F_3)$$

$$y_{k+1} = y_k + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4)$$

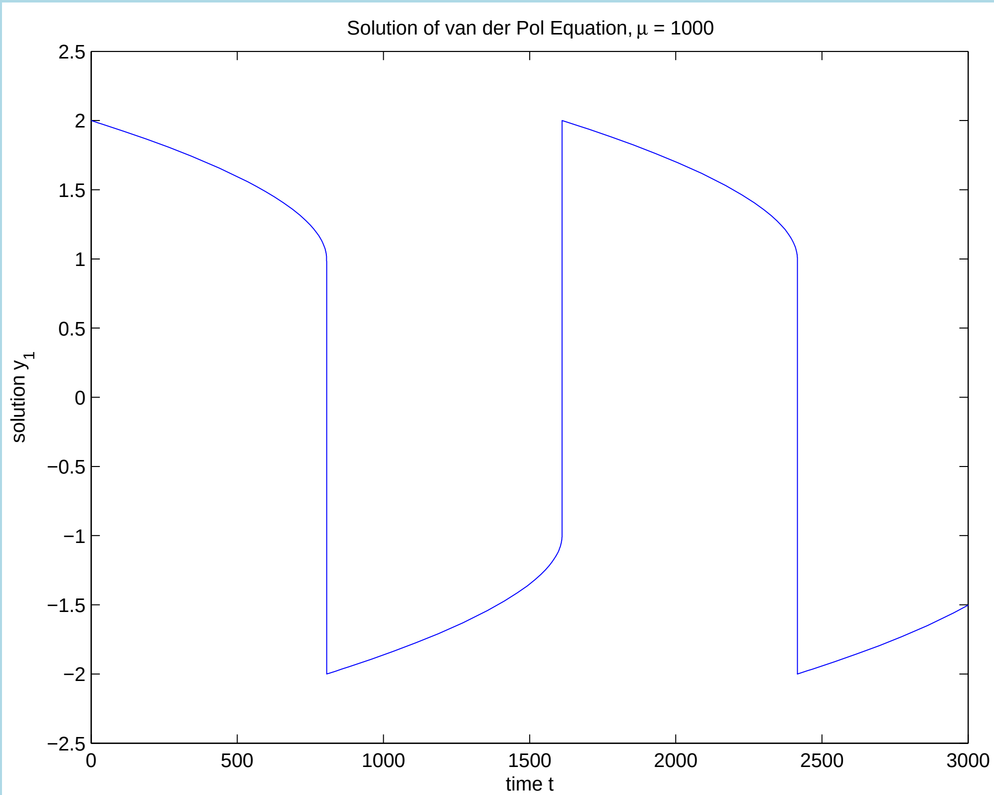
End for

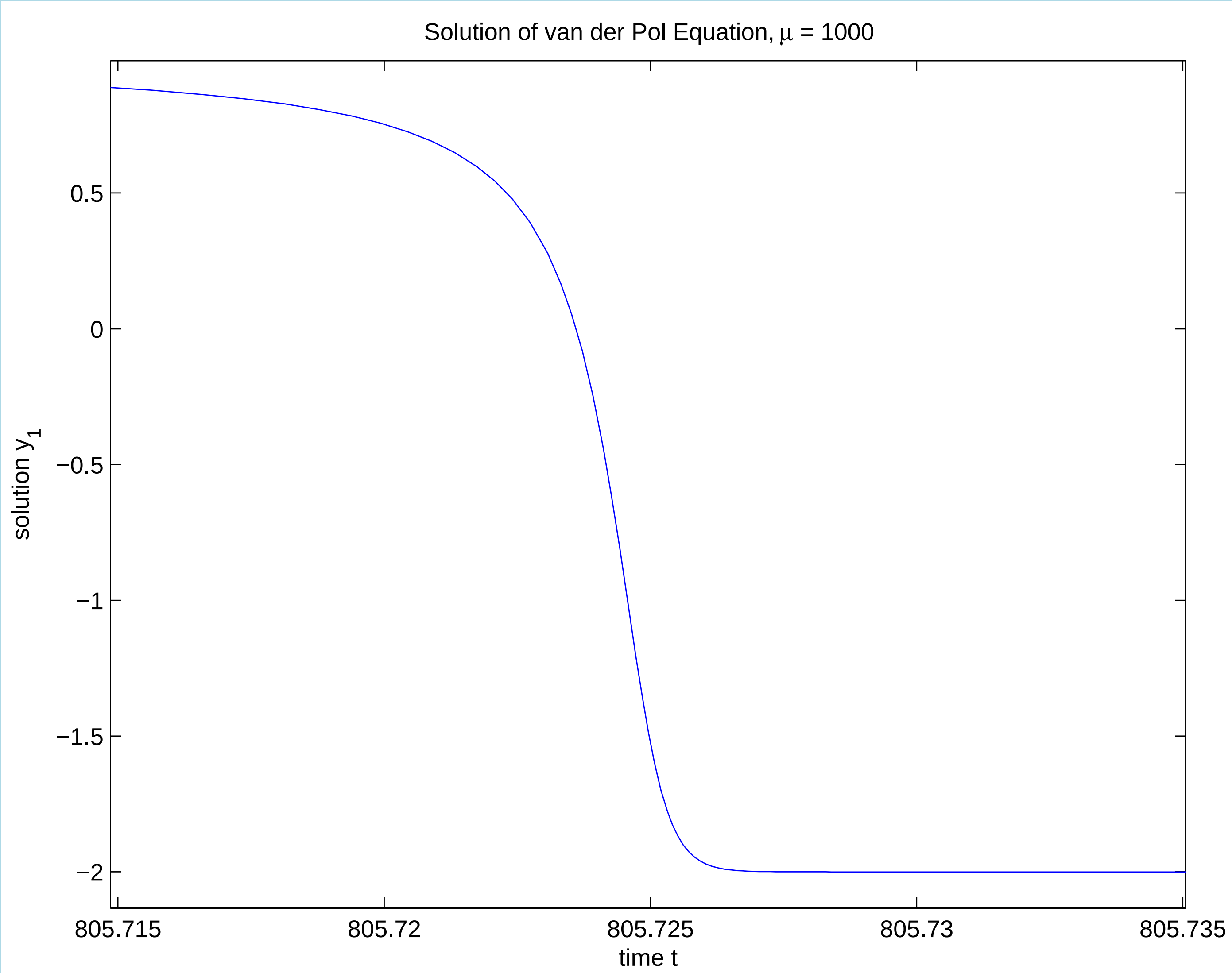
The fourth Runge-Kutta method involves a local truncation error of $O(h^5)$.

4 – Multistep Methods

Example 2 (Stiff problem, van der Pol equation)

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= \mu(1 - y_1^2)y_2 - y_1. \end{aligned}$$




Solution of van der Pol Equation, μ = 1000
0.5
0
−0.5
−1
−1.5
−2
solution y_1
805.715
805.72
805.725
805.73
805.735
time t

Definition 1 *A multistep method for solving the initial-value problem*

$$y' = f(x, y),\ a \le x \le b,\ y(a) = \alpha$$

is

$$\begin{aligned} w_{i+1} &= a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} + \\ &\quad h\left[b_m f(x_{i+1}, w_{i+1}) + b_{m-1} f(x_i, w_i) + \cdots + b_0 f(x_{i+1-m}, w_{i+1-m})\right] \end{aligned}$$

for $i = m-1, m, \ldots, N-1$*, where the starting values*

$$w_0 = \alpha,\ w_1 = \alpha_1,\ \cdots, w_{m-1} = \alpha_{m-1}$$

are specified and $h = (b-a)/N$*.*

Definition 1 *A multistep method for solving the initial-value problem*

$$y' = f(x,y),\ a \leq x \leq b,\ y(a) = \alpha$$

is

$$\begin{aligned} w_{i+1} &= a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} + \\ &\quad h\left[b_m f(x_{i+1}, w_{i+1}) + b_{m-1} f(x_i, w_i) + \cdots + b_0 f(x_{i+1-m}, w_{i+1-m})\right] \end{aligned}$$

for $i = m-1, m, \ldots, N-1$, *where the starting values*

$$w_0 = \alpha,\ w_1 = \alpha_1,\ \cdots, w_{m-1} = \alpha_{m-1}$$

are specified and $h = (b-a)/N$.

☞ If $b_m = 0$, the method is called explicit or open

Definition 1 *A multistep method for solving the initial-value problem*

$$y' = f(x,y),\ a \le x \le b,\ y(a) = \alpha$$

is

$$\begin{aligned} w_{i+1} &= a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} + \\ &\quad h\left[b_m f(x_{i+1}, w_{i+1}) + b_{m-1} f(x_i, w_i) + \cdots + b_0 f(x_{i+1-m}, w_{i+1-m})\right] \end{aligned}$$

for $i = m-1, m, \ldots, N-1$*, where the starting values*

$$w_0 = \alpha,\ w_1 = \alpha_1,\ \cdots, w_{m-1} = \alpha_{m-1}$$

are specified and $h = (b-a)/N$*.*

☞ If $b_m = 0$, the method is called explicit or open

☞ If $b_m \neq 0$, the method is called implicit or closed.

Example 3

☞ *Explicit fourth-order Adams-Bashforth method:*

$$\begin{aligned} w_0 &= \alpha,\ w_1 = \alpha_1,\ w_2 = \alpha_2,\ w_3 = \alpha_3, \\ w_{i+1} &= w_i + \frac{h}{24}\left[55f(x_i, w_i) - 59f(x_{i-1}, w_{i-1})\right. \\ &\qquad \left. +37f(x_{i-2}, w_{i-2}) - 9f(x_{i-3}, w_{i-3})\right] \end{aligned}$$

for each $i = 3, 4, \ldots, N-1$.

Example 3

☞ *Explicit fourth-order Adams-Bashforth method:*

$$w_0 = \alpha,\ w_1 = \alpha_1,\ w_2 = \alpha_2,\ w_3 = \alpha_3,$$
$$w_{i+1} = w_i + \frac{h}{24}\left[55f(x_i, w_i) - 59f(x_{i-1}, w_{i-1}) + 37f(x_{i-2}, w_{i-2}) - 9f(x_{i-3}, w_{i-3})\right]$$

for each $i = 3, 4, \ldots, N-1$.

☞ *Implicit fourth-order Adams-Moulton method:*

$$w_0 = \alpha,\ w_1 = \alpha_1,\ w_2 = \alpha_2,$$
$$w_{i+1} = w_i + \frac{h}{24}\left[9f(x_{i+1}, w_{i+1}) + 19f(x_i, w_i) - 5f(x_{i-1}, w_{i-1}) + f(x_{i-2}, w_{i-2})\right]$$

for each $i = 2, 3, \ldots, N-1$.

- Derivation of the multistep methods:

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} y'(x)dx = \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

$$\Rightarrow \quad y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

➾ Derivation of the multistep methods:

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} y'(x)dx = \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

$$\Rightarrow \quad y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

$P(x)$: an interpolating polynomial to $f(x, y(x))$ and $y(x_i) \approx w_i$.

➾ Derivation of the multistep methods:

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} y'(x)dx = \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

$$\Rightarrow \quad y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

$P(x)$: an interpolating polynomial to $f(x, y(x))$ and $y(x_i) \approx w_i$. Then

$$y(x_{i+1}) \approx w_i + \int_{x_i}^{x_{i+1}} P(x)dx$$

- Derivation of the multistep methods:

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} y'(x)dx = \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

$$\Rightarrow \quad y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

$P(x)$: an interpolating polynomial to $f(x, y(x))$ and $y(x_i) \approx w_i$. Then

$$y(x_{i+1}) \approx w_i + \int_{x_i}^{x_{i+1}} P(x)dx$$

- Adams-Bashforth explicit m-step technique:

- Derivation of the multistep methods:

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} y'(x)dx = \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

$$\Rightarrow \quad y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

$P(x)$: an interpolating polynomial to $f(x, y(x))$ and $y(x_i) \approx w_i$. Then

$$y(x_{i+1}) \approx w_i + \int_{x_i}^{x_{i+1}} P(x)dx$$

- Adams-Bashforth explicit m-step technique:

The binomial formula is defined as

$$\binom{s}{k} = \frac{s(s-1)\cdots(s-k+1)}{k!}$$

We introduce the backward difference notation ∇

$$\nabla f(x_i) = f(x_i) - f(x_{i-1})$$

and

$$\nabla^k f(x_i) = \nabla^{k-1} f(x_i) - \nabla^{k-1} f(x_{i-1}) = \nabla \left(\nabla^{k-1} f(x_i) \right),$$

for $i = 0, 1, \ldots, n-1$.

We introduce the backward difference notation ∇

$$\nabla f(x_i) = f(x_i) - f(x_{i-1})$$

and

$$\nabla^k f(x_i) = \nabla^{k-1} f(x_i) - \nabla^{k-1} f(x_{i-1}) = \nabla \left(\nabla^{k-1} f(x_i) \right),$$

for $i = 0, 1, \ldots, n-1$.

Let $P_{m-1}(x)$ be the Newton backward difference polynomial through $(x_i, f(x_i, y(x_i)))$, $(x_{i-1}, f(x_{i-1}, y(x_{i-1})))$, . . ., $(x_{i+1-m}, f(x_{i+1-m}, y(x_{i+1-m})))$,

We introduce the backward difference notation ∇

$$\nabla f(x_i) = f(x_i) - f(x_{i-1})$$

and

$$\nabla^k f(x_i) = \nabla^{k-1} f(x_i) - \nabla^{k-1} f(x_{i-1}) = \nabla \left(\nabla^{k-1} f(x_i) \right),$$

for $i = 0, 1, \ldots, n-1$.

Let $P_{m-1}(x)$ be the Newton backward difference polynomial through $(x_i, f(x_i, y(x_i)))$, $(x_{i-1}, f(x_{i-1}, y(x_{i-1})))$, . . ., $(x_{i+1-m}, f(x_{i+1-m}, y(x_{i+1-m})))$, i.e.,

$$P_{m-1}(x) = \sum_{k=0}^{m-1} (-1)^k \begin{pmatrix} -s \\ k \end{pmatrix} \nabla^k f(x_i, y(x_i))$$

where $x = x_i + sh$. Then

$$f(x, y(x)) = P_{m-1}(x) + \frac{f^{(m)}(\xi_i, y(\xi_i))}{m!}(x - x_i)(x - x_{i-1}) \cdots (x - x_{i+1-m})$$

$$
\begin{aligned}
f(x, y(x)) &= P_{m-1}(x) + \frac{f^{(m)}(\xi_i, y(\xi_i))}{m!}(x - x_i)(x - x_{i-1}) \cdots (x - x_{i+1-m}) \\
&= P_{m-1}(x) + \frac{h^m f^{(m)}(\xi_i, y(\xi_i))}{m!} s(s+1) \cdots (s+m-1)
\end{aligned}
$$

for some $\xi_i \in (x_{i+1-m}, x_i)$. Hence

$$f(x, y(x)) = P_{m-1}(x) + \frac{f^{(m)}(\xi_i, y(\xi_i))}{m!}(x - x_i)(x - x_{i-1}) \cdots (x - x_{i+1-m})$$
$$= P_{m-1}(x) + \frac{h^m f^{(m)}(\xi_i, y(\xi_i))}{m!} s(s+1) \cdots (s+m-1)$$

for some $\xi_i \in (x_{i+1-m}, x_i)$. Hence

$$\int_{x_i}^{x_{i+1}} f(x, y(x))dx = \int_{x_i}^{x_{i+1}} \sum_{k=0}^{m-1} (-1)^k \begin{pmatrix} -s \\ k \end{pmatrix} \nabla^k f(x_i, y(x_i))dx$$
$$+ \int_{x_i}^{x_{i+1}} \frac{h^m f^{(m)}(\xi_i, y(\xi_i))}{m!} s(s+1) \cdots (s+m-1)dx$$

$$f(x,y(x)) = P_{m-1}(x) + \frac{f^{(m)}(\xi_i, y(\xi_i))}{m!}(x-x_i)(x-x_{i-1})\cdots(x-x_{i+1-m})$$

$$= P_{m-1}(x) + \frac{h^m f^{(m)}(\xi_i, y(\xi_i))}{m!} s(s+1)\cdots(s+m-1)$$

for some $\xi_i \in (x_{i+1-m}, x_i)$. Hence

$$\begin{aligned}
\int_{x_i}^{x_{i+1}} f(x,y(x))dx \quad &= \quad \int_{x_i}^{x_{i+1}} \sum_{k=0}^{m-1} (-1)^k \left(\begin{array}{c} -s \\ k \end{array} \right) \nabla^k f(x_i, y(x_i))dx \\
&+ \int_{x_i}^{x_{i+1}} \frac{h^m f^{(m)}(\xi_i, y(\xi_i))}{m!} s(s+1)\cdots(s+m-1)dx \\
&= \sum_{k=0}^{m-1} \nabla^k f(x_i, y(x_i)) h (-1)^k \int_0^1 \left(\begin{array}{c} -s \\ k \end{array} \right) ds \\
&+ \frac{h^{m+1}}{m!} \int_0^1 s(s+1)\cdots(s+m-1) f^{(m)}(\xi_i, y(\xi_i))ds
\end{aligned}$$

IVP of ODE

k	0	1	2	3	4	5
$(-1)^k \int_0^1 \binom{-s}{k} ds$	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$

k	0	1	2	3	4	5
$(-1)^k \int_0^1 \begin{pmatrix} -s \\ k \end{pmatrix} ds$	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$

Since $s(s+1)\cdots(s+m-1)$ does not change sign on $[0,1]$, by the Weighted Mean Value Theorem for Integrals, $\exists\, \mu_i \in (x_i, x_{i+1})$ such that

$$
\begin{aligned}
& \frac{h^{m+1}}{m!} \int_0^1 s(s+1)\cdots(s+m-1) f^{(m)}(\xi_i, y(\xi_i)) ds \\
= \quad & \frac{h^{m+1} f^{(m)}(\mu_i, y(\mu_i))}{m!} \int_0^1 s(s+1)\cdots(s+m-1) ds \\
= \quad & h^{m+1} f^{(m)}(\mu_i, y(\mu_i)) (-1)^m \int_0^1 \begin{pmatrix} -s \\ m \end{pmatrix} ds
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
y(x_{i+1}) &= y(x_i) + h\left[f(x_i, y(x_i)) + \frac{1}{2}\nabla f(x_i, y(x_i))\right. \\
&\quad \left. + \frac{5}{12}\nabla^2 f(x_i, y(x_i)) + \cdots\right] \\
&\quad + h^{m+1} f^{(m)}(\mu_i, y(\mu_i))(-1)^m \int_0^1 \begin{pmatrix} -s \\ m \end{pmatrix} ds
\end{aligned}
$$

- Predictor-Corrector Method:

- Predictor-Corrector Method:
  - Using explicit method as a predictor:

$$
\begin{aligned}
w_{i+1}^{(0)} &= a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\
&\quad + h\left[b_{m-1}f(x_i, w_i) + \cdots + b_0 f(x_{i+1-m}, w_{i+1-m})\right]
\end{aligned}
$$

- Predictor-Corrector Method:
  - Using explicit method as a predictor:

$$\begin{aligned} w_{i+1}^{(0)} &= a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\ &\quad + h\left[b_{m-1}f(x_i, w_i) + \cdots + b_0 f(x_{i+1-m}, w_{i+1-m})\right] \end{aligned}$$

  - Using implicit method as a corrector:

$$\begin{aligned} w_{i+1} &= a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\ &\quad + h\Big[b_m f(x_{i+1}, w_{i+1}^{(0)}) + b_{m-1}f(x_i, w_i) \\ &\quad + \cdots + b_0 f(x_{i+1-m}, w_{i+1-m})\Big] \end{aligned}$$

5 – Systems and Higher-Order Ordinary Differential Equations

Consider a system of first-order ODE's.

$$\begin{cases} y_1' &= f_1(x, y_1, y_2, \ldots, y_n) \\ y_2' &= f_2(x, y_1, y_2, \ldots, y_n) \\ &\vdots \\ y_n' &= f_n(x, y_1, y_2, \ldots, y_n) \end{cases}$$

with initial conditions

$$\begin{cases} y_1(x_0) &= y_1^0 \\ y_2(x_0) &= y_2^0 \\ &\vdots \\ y_n(x_0) &= y_n^0 \end{cases}$$

Define

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \qquad F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \qquad Y_0 = \begin{bmatrix} y_1^0 \\ y_2^0 \\ \vdots \\ y_n^0 \end{bmatrix}$$

and transform the system into vector form

$$\begin{cases} Y' & = F(x, Y) \\ Y(x_0) & = Y_0 \end{cases}$$

Define

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \qquad F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \qquad Y_0 = \begin{bmatrix} y_1^0 \\ y_2^0 \\ \vdots \\ y_n^0 \end{bmatrix}$$

and transform the system into vector form

$$\begin{cases} Y' & = F(x, Y) \\ Y(x_0) & = Y_0 \end{cases}$$

☞ The Runge-Kutta methods can be easily extended to vector form.

Define

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \qquad F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}, \qquad Y_0 = \begin{bmatrix} y_1^0 \\ y_2^0 \\ \vdots \\ y_n^0 \end{bmatrix}$$

and transform the system into vector form

$$\begin{cases} Y' & = F(x, Y) \\ Y(x_0) & = Y_0 \end{cases}$$

- ☞ The Runge-Kutta methods can be easily extended to vector form.
- ☞ An higher order ODE can be converted into a system of first-order equations, hence higher-order ODEs can be solved in vector form.

Example 4

$$(\sin x)y''' + \cos(xy) + \sin(x^2 + y'') + (y')^3 = \log x$$

$$y(2) = 7,\ y'(2) = 3,\ y''(2) = -4$$

Example 4

$$(\sin x)y''' + \cos(xy) + \sin(x^2 + y'') + (y')^3 = \log x$$

$$y(2) = 7,\ y'(2) = 3,\ y''(2) = -4$$

Solution: Let $u_1 = y, u_2 = y'$, and $u_3 = y''$

Example 4

$$(\sin x)y''' + \cos(xy) + \sin(x^2 + y'') + (y')^3 = \log x$$
$$y(2) = 7,\ y'(2) = 3,\ y''(2) = -4$$

Solution: Let $u_1 = y, u_2 = y'$, and $u_3 = y''$

$$(\sin x)y''' + \cos(xy) + \sin(x^2 + y'') + (y')^3 = \log x$$
$$\Rightarrow \quad (\sin x)u_3' + \cos(xu_1) + \sin(x^2 + u_3) + (u_2)^3 = \log x$$

Example 4

$$(\sin x)y''' + \cos(xy) + \sin(x^2 + y'') + (y')^3 = \log x$$
$$y(2) = 7,\ y'(2) = 3,\ y''(2) = -4$$

Solution: Let $u_1 = y, u_2 = y'$, and $u_3 = y''$

$$(\sin x)y''' + \cos(xy) + \sin(x^2 + y'') + (y')^3 = \log x$$
$$\Rightarrow \quad (\sin x)u_3^{'} + \cos(xu_1) + \sin(x^2 + u_3) + (u_2)^3 = \log x$$
$$\Rightarrow \quad u_3' = \left[\log x - \cos(xu_1) - \sin(x^2 + u_3) - (u_2)^3\right] / \sin x$$

Example 4

$$(\sin x)y''' + \cos(xy) + \sin(x^2 + y'') + (y')^3 = \log x$$
$$y(2) = 7,\ y'(2) = 3,\ y''(2) = -4$$

Solution: Let $u_1 = y, u_2 = y'$, and $u_3 = y''$

$$\begin{aligned} & (\sin x)y''' + \cos(xy) + \sin(x^2 + y'') + (y')^3 = \log x \\ \Rightarrow\ & (\sin x)u_3' + \cos(xu_1) + \sin(x^2 + u_3) + (u_2)^3 = \log x \\ \Rightarrow\ & u_3' = \left[\log x - \cos(xu_1) - \sin(x^2 + u_3) - (u_2)^3\right] / \sin x \end{aligned}$$

Therefore,

$$\begin{cases} u_1' &= u_2 \\ u_2' &= u_3 \\ u_3' &= [\log x - \cos(xu_1) - \sin(x^2 + u_3) - u_2^3]/\sin x \end{cases}$$

with $u_1(2) = 7, u_2(2) = 3$ and $u_3(2) = -4$.

Let

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad F = \begin{bmatrix} u_2 \\ u_3 \\ \left[\log x - \cos(xu_1) - \sin(x^2 + u_3) - u_2^3\right] / \sin x \end{bmatrix},$$

and $U_0 = \begin{bmatrix} 7 & 3 & -4 \end{bmatrix}^T$, $x_0 = 2$.

Let

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \; F = \begin{bmatrix} u_2 \\ u_3 \\ \left[\log x - \cos(xu_1) - \sin(x^2 + u_3) - u_2^3\right] / \sin x \end{bmatrix},$$

and $U_0 = \begin{bmatrix} 7 & 3 & -4 \end{bmatrix}^T$, $x_0 = 2$. Then the higher-order ODE becomes

$$\begin{cases} U' & = F(x, U) \\ U(x_0) & = U_0 \end{cases}$$

and can be solved by Runge-Kutta methods. ■

Let

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \; F = \begin{bmatrix} u_2 \\ u_3 \\ \left[\log x - \cos(xu_1) - \sin(x^2 + u_3) - u_2^3\right] / \sin x \end{bmatrix},$$

and $U_0 = \begin{bmatrix} 7 & 3 & -4 \end{bmatrix}^T$, $x_0 = 2$. Then the higher-order ODE becomes

$$\begin{cases} U' & = F(x, U) \\ U(x_0) & = U_0 \end{cases}$$

and can be solved by Runge-Kutta methods. ∎

Truncate Taylor series:

$$y_i(x+h) = y_i(x) + hy_i'(x) + \frac{h^2}{2!} y_i''(x) + \cdots + \frac{h^n}{n!} y_i^{(n)}(x)$$

$$\Rightarrow\ Y(x+h) = Y(x) + hY'(x) + \frac{h^2}{2!}Y''(x) + \cdots + \frac{h^n}{n!}Y^{(n)}(x)$$

$$\Rightarrow \; Y(x+h) = Y(x) + hY'(x) + \frac{h^2}{2!}Y''(x) + \cdots + \frac{h^n}{n!}Y^{(n)}(x)$$

The classical fourth-order Runge-Kunge-Kutta formula, in vector form, are

$$Y(x+h) = Y(x) + \frac{h}{6}(F_1 + 2F_2 + 2F_3 + F_4)$$

where

$$\begin{cases} F_1 = F(Y) \\ F_2 = F(Y + \frac{1}{2}F_1) \\ F_3 = F(Y + \frac{1}{2}F_2) \\ F_4 = F(Y + F_3) \end{cases}$$

$$\Rightarrow\ Y(x+h) = Y(x) + hY'(x) + \frac{h^2}{2!}Y''(x) + \cdots + \frac{h^n}{n!}Y^{(n)}(x)$$

The classical fourth-order Runge-Kunge-Kutta formula, in vector form, are

$$Y(x+h) = Y(x) + \frac{h}{6}(F_1 + 2F_2 + 2F_3 + F_4)$$

where

$$\begin{cases} F_1 = F(Y) \\ F_2 = F(Y + \frac{1}{2}F_1) \\ F_3 = F(Y + \frac{1}{2}F_2) \\ F_4 = F(Y + F_3) \end{cases}$$

Example 5 *Compute the solution the following initial-value problem on* $-2 \le t \le 1$.

$$\begin{cases} x' = x + y^2 - t^3, & x(1) = 3 \\ y' = y + x^3 + \cos t, & y(1) = 1. \end{cases}$$

Matlab program for Example 5.

```
format long e
clear all
TSPAN = 1:-0.01:-2;
y0(1) = 3; y0(2) = 1;
[T,Y] = ode45('fun_ex2',TSPAN,y0);
plot(T,Y(:,1),'b-',T,Y(:,2),'r-.');
legend('x-curve','y-curve')
```

ODE function

```
function [DY] = fun_ex2(T,Y)
DY(1,1) = Y(1) + Y(2)*Y(2) - T * T * T;
DY(2,1) = Y(2) + Y(1) * Y(1) * Y(1) + cos(T);
```

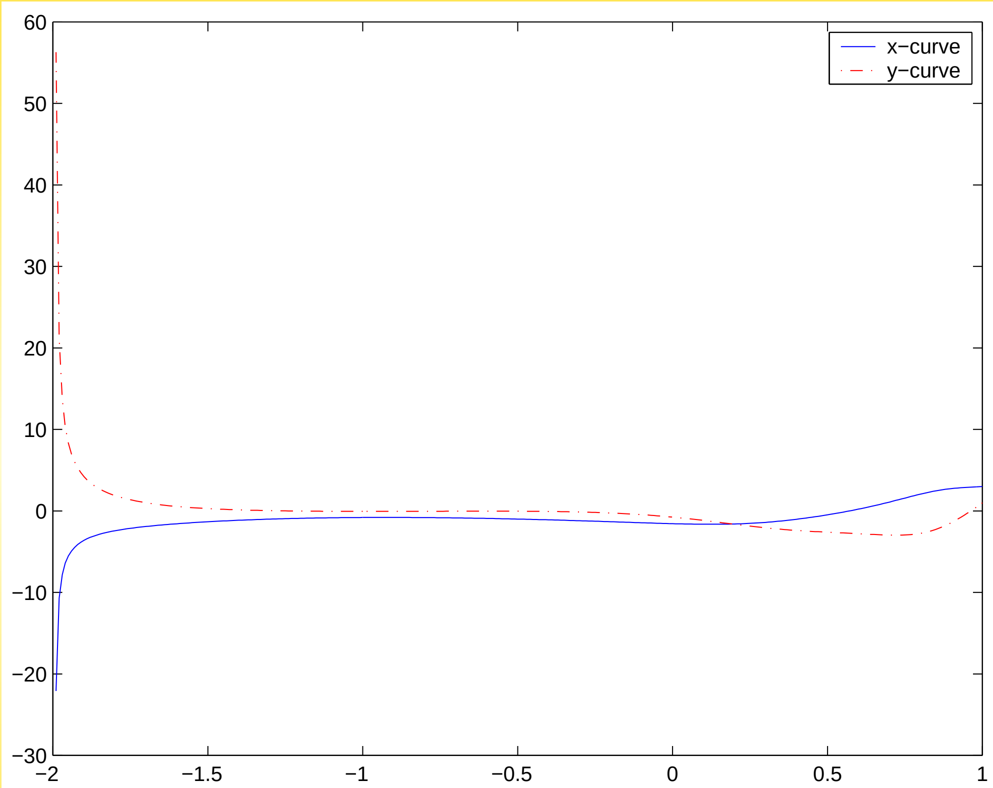


Figure 1: Solution curves for Example 5.