MSDM5004
Numerical Methods and Modeling in Science
Spring 2024

Lecture 7

Prof Yang Xiang
Hong Kong University of Science and Technology

Chapter 9

Numerical solution of ordinary differential equations (ODEs)

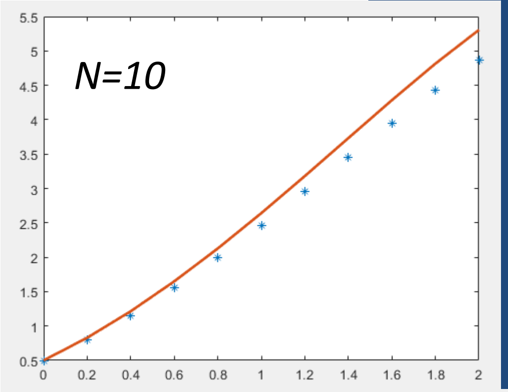
I. Introduction

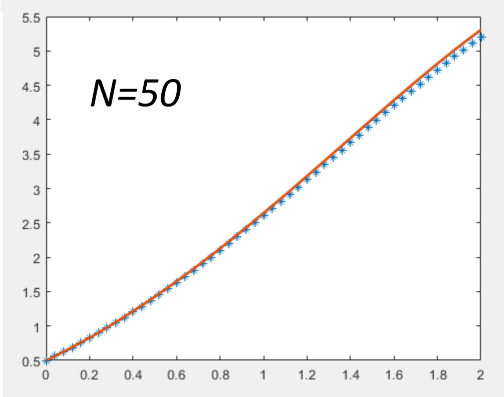
Initial value problem of ODE

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha$$

Approximation to y will be generated at discrete points,

called mesh points, in the interval $[a, b]$.





$$t_i = a + ih, \quad \text{for each } i = 0, 1, 2, \ldots, N.$$

$$y_0 = y(t_0)$$

$$y_i \approx y(t_i), \quad i = 1, 2, \cdots, N$$

Idea 1

$$y'(t_i) = f(t_i, y(t_i))$$

Approximating the derivative by some finite difference scheme

$$y'(t_i) \approx \frac{y(t_{i+1}) - y(t_i)}{h}$$

Or

$$y'(t_{i+1}) \approx \frac{y(t_{i+1}) - y(t_i)}{h}$$

Idea 2

$$y'(t) = f(t, y(t)),\ t \in [t_i, t_{i+1}]:$$

$$\Longrightarrow y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(s, y(s))\mathrm{d}s$$

Approximating the integral using some numerical method

e.g., using trapezoidal rule to approximate the integral

$$y(t_{i+1}) \approx y(t_i) + \frac{h}{2}\left[f(t_i, y(t_i)) + f(t_{i+1}, y(t_{i+1}))\right]$$

2. Euler method

$$y_{i+1} = y_i + hf(t_i, y_i) \qquad i = 1, 2, \cdots, N$$

Sometimes, it also called forward Euler method.

Idea of derivation

$$\frac{y(t_{i+1}) - y(t_i)}{h} \approx y'(t_i) = f(t_i, y(t_i))$$

first order approximation

$$y_i \approx y(t_i) \qquad i = 1, 2, \cdots, N$$

A linear equation

$$y' = \lambda y \qquad\qquad \lambda \text{ constant}$$

Using Euler method,

$$y_{i+1} = y_i + h\lambda y_i = (1 + h\lambda)y_i$$

$$y_i = (1 + h\lambda)^i y_0$$

Note: the exact solution of this ODE is $y(t) = e^{\lambda t} y(t_0)$

For a fixed time T, the numerical solution is

$$(1+h\lambda)^{T/h} y_0 \to e^{\lambda T} y_0, \quad h \to 0$$

$$(N \to +\infty)$$

$$\boxed{\left(1+\frac{1}{x}\right)^x \to e, \quad x \to +\infty}$$

Local error

Using Taylor expansion, the exact solution satisfies

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(\xi), \quad \xi \in [t_i, t_{i+1}]$$

The numerical solution using Euler method satisfies

$$y_{i+1} = y(t_i) + hy'(t_i) \qquad \text{when } y_i = y(t_i)$$

Taking difference, the local error

$$e_{i+1} = y(t_{i+1}) - y_{i+1} = \frac{h^2}{2}y''(\xi) = O(h^2)$$

Global error

$$y'(t_i) = f(t_i, y(t_i))$$

$$|y(t_i) - y_i| \le \frac{hM}{2L}\left[e^{L(t_i - a)} - 1\right]$$

$\boxed{O(h) \quad \text{error}}$

where $|y''(t)| \le M, \quad \text{for all } t \in [a,b]$

$$|f(t, y_1) - f(t, y_2,)| \le L|y_1 - y_2|$$ Lipschitz condition

Idea of the proof

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2}y''(\xi)$$

$$y_{i+1} = y_i + hf(t_i, y_i)$$

$$y(t_{i+1}) - y_{i+1} = y(t_i) - y_i + h[f(t_i, y(t_i)) - f(t_i, y_i)] + \frac{h^2}{2}y''(\xi)$$

$$y(t_{i+1}) - y_{i+1} = y(t_i) - y_i + h[f(t_i, y(t_i)) - f(t_i, y_i)] + \frac{h^2}{2}y''(\xi)$$

Further using the Lipschitz condition

$$|f(t_i, y(t_i)) - f(t_i, y_i)| \leq L|y(t_i) - y_i|,$$

we have

$$|y(t_{i+i}) - y_{i+1}| \leq (1 + hL)|y(t_i) - y_i| + \frac{M}{2}h^2.$$

It can be proved that

$$|y(t_i) - y_i| \leq \frac{hM}{2L}\left[e^{L(t_i - a)} - 1\right]$$

Note that $t_i = ih + a$.

An example

$$y' = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5.$$

With $N = 10$ we have $h = 0.2$, $t_i = 0.2i$, $y_0 = 0.5$

Using Euler method,

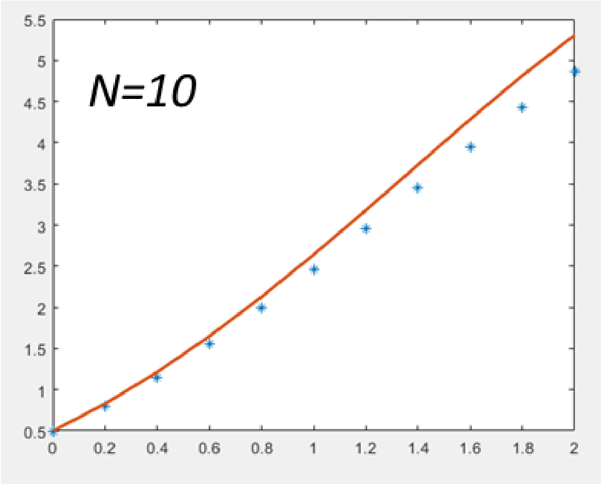
$$y_{i+1} = y_i + h(y_i - t_i^2 + 1)$$

$$y_1 = y_0 + h(y_0 - t_0^2 + 1) = 0.5 + 0.2(0.5 - 0^2 + 1) = 0.8$$

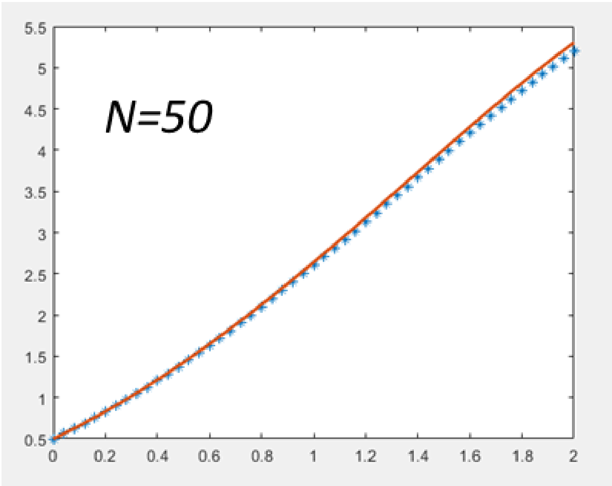
$$y_2 = y_1 + h(y_1 - t_1^2 + 1) = 0.8 + 0.2(0.8 - (0.2)^2 + 1) = 1.152$$

$$\dots$$

t_i	y_i
0.0	0.5000000
0.2	0.8000000
0.4	1.1520000
0.6	1.5504000
0.8	1.9884800
1.0	2.4581760
1.2	2.9498112
1.4	3.4517734
1.6	3.9501281
1.8	4.4281538
2.0	4.8657845

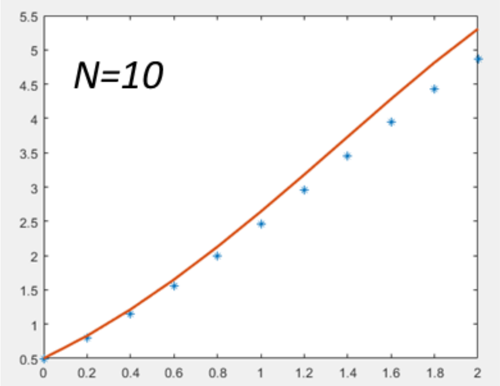


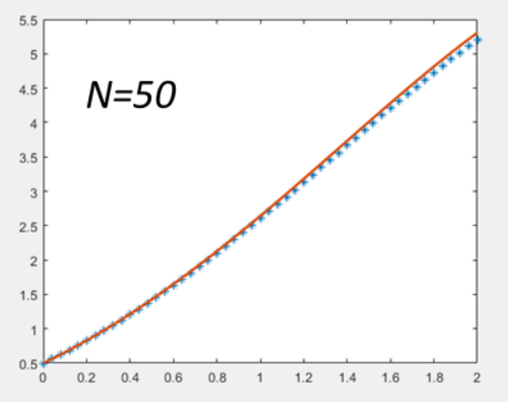
N=50



MATLAB code

```
n=10;
a=0;
b=2;
t=linspace(a,b,n+1);
y=zeros(n+1,1);
dt=(b-a)/n;
y(1)=0.5;
for i=1:n
   y(i+1)=y(i)+(y(i)-t(i)^2+1)*dt;
end;
plot(t,y,'*')
```





3. Backward Euler method

$$y_{i+1} = y_i + hf(t_{i+1}, y_{i+1}) \qquad i = 1, 2, \cdots, N$$

Idea of derivation

$$\frac{y(t_{i+1}) - y(t_i)}{h} \approx y'(t_{i+1}) = f(t_{i+1}, y_{i+1})$$

$$y_i \approx y(t_i) \qquad i = 1, 2, \cdots, N$$

The backward Euler method is an implicit method

$$y_{i+1} = y_i + hf(t_{i+1}, y_{i+1})$$

In general, one needs to solve this nonlinear equation for y_{i+1}

The Euler method (or forward Euler method) is an explicit method

$$y_{i+1} = y_i + hf(t_i, y_i)$$

y_{i+1} is calculated directly from y_i

Implicit methods have better stability than explicit methods

– see more discussion in the methods for PDEs

Linear equation

$$y' = \lambda y \qquad \lambda \text{ constant}$$

Using backward Euler method,

$$y_{i+1} = y_i + h\lambda y_{i+1}$$

$$y_{i+1} = \frac{1}{1 - h\lambda} y_i$$

$$y_i = \frac{1}{(1 - h\lambda)^i} y_0$$

Note: the exact solution of this ODE is $y(t) = e^{\lambda t} y(t_0)$

Local error

$$e_{i+1} = y(t_{i+1}) - y_{i+1} = O(h^2)$$

Global error

$$|y(t_i) - y_i| \leq O(h)$$

4. Trapezoidal method

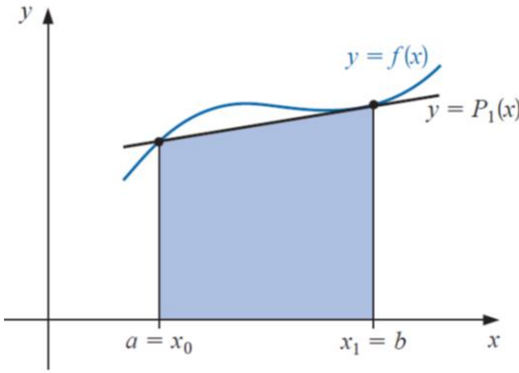
$$y_{i+1} = y_i + \frac{h}{2}\left[f(t_i, y_i) + f(t_{i+1}, y_{i+1})\right] \qquad i = 1, 2, \cdots, N$$

An implicit method

Idea of derivation

$$y'(t) = f(t, y(t)),\ t \in [t_i, t_{i+1}]$$

$$\Longrightarrow y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(s, y(s))\mathrm{d}s$$



Approximating the integral using trapezoidal rule

Other numerical methods for solving ODEs can also be derived by numerical approximation of the integral.

Linear equation

$$y' = \lambda y \qquad \lambda \text{ constant}$$

Using the trapezoidal method,

$$y_{i+1} = y_i + \frac{h}{2}\left[f(t_i, y_i) + f(t_{i+1}, y_{i+1})\right] = y_i + \frac{h}{2}(\lambda y_i + \lambda y_{i+1})$$

$$y_{i+1} = \frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}} y_i$$

$$y_i = \left(\frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}}\right)^i y_0$$

Note: the exact solution of this ODE is $y(t) = e^{\lambda t} y(t_0)$

Local error

$$e_{i+1} = y(t_{i+1}) - y_{i+1} = O(h^3)$$

Global error

$$|y(t_i) - y_i| \leq \; O(h^2)$$

5. Linearization of an implicit method

$$y_{i+1} = y_i + \frac{h}{2}\left[f(t_i, y_i) + f(t_{i+1}, y_{i+1})\right]$$

Not known yet

Using linear approximation

$$f(t_{i+1}, y_{i+1}) \approx f(t_{i+1}, y_i) + \frac{\partial f}{\partial y}(t_{i+1}, y_i)(y_{i+1} - y_i)$$

solve for y_{i+1}

6. Runge-Kutta method

$$y'(t) = f(t, y(t)),\ t \in [t_i, t_{i+1}]$$

Idea

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \frac{h^3}{3!}y'''(t_i) + \cdots$$

We hope to have a more accurate approximation of $y(t_{i+1})$

$$y'(t_i) = f(t_i, y(t_i)) \qquad \surd$$

$$y''(t_i) = f_t + f_y y'\big|_{t=t_i} = f_t + f_y f\big|_{t=t_i} \qquad ?$$

$$f_t = \frac{\partial f}{\partial t}$$

$$y''(t_i) = f_{tt} + f_{ty}f + (f_t + f_y f)f_y + f(f_{yt} + f_{yy}f)\big|_{t=t_i} \qquad ?$$

$$\cdots$$

We want to have a second order scheme

$$\begin{aligned} y(t_{i+1}) &= y + hy' + \frac{h^2}{2}y''\Big|_{t=t_i} + O(h^3) \\ &= y + hf + \frac{h^2}{2}(f_t + f_y f)\Big|_{t=t_i} + O(h^3) \end{aligned}$$

Using

$$f(t+h, y+hf) = f + hf_t + hf\,f_y\big|_{(t,y)} + O(h^2),$$

we have

$$y(t_{i+1}) = y + \frac{h}{2}f + \frac{h}{2}f(t+h, y+hf)\Big|_{t=t_i} + O(h^3).$$

$$y(t_{i+1}) = y + \frac{h}{2}f + \frac{h}{2}f(t+h, y+hf)\Big|_{t=t_i} + O(h^3)$$

Therefore, we have a second order scheme

$$y_{i+1} = y_i + \frac{h}{2}f(t_i, y_i) + \frac{h}{2}f(t_i + h, y_i + hf(t_i, y_i)).$$

It can be written as

$$\begin{aligned}\xi_1 &= f(t_i, y_i)\\ \xi_2 &= f(t_i + h, y_i + h\xi_1)\\ y_{i+1} &= y_i + \frac{1}{2}h\xi_1 + \frac{1}{2}h\xi_2\end{aligned}$$

Local error $e_{i+1} = y(t_{i+1}) - y_{i+1} = O(h^3)$ Global error $|y(t_i) - y_i| \leq O(h^2)$

This is a second order Runge-Kutta method.

A general second order Runge-Kutta method

$$\begin{aligned}
\xi_1 &= f(t_i, y_i) \\
\xi_2 &= f(t_i + \alpha h, y_i + \beta h \xi_1) \\
y_{i+1} &= y_i + a h \xi_1 + b h \xi_2
\end{aligned}$$

Parameters of α, β, a, b can be determined by Taylor expansions.

Another second order Runge-Kutta method

$$\begin{aligned}
\xi_1 &= f(t_i, y_i) \\
\xi_2 &= f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h\xi_1) \\
y_{i+1} &= h\xi_2.
\end{aligned}$$

4th order Runge-Kutta method

$$\begin{aligned}
\xi_1 &= f(t_i, y_i) \\
\xi_2 &= f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h\xi_1) \\
\xi_3 &= f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}h\xi_2) \\
\xi_4 &= f(t_i + h, y_i + h\xi_3) \\
y_{i+1} &= y_i + \frac{1}{6}h(\xi_1 + 2\xi_2 + 2\xi_3 + \xi_4)
\end{aligned}$$

7. Multistep methods

$$y' = f(t, y),$$

$$\sum_{m=0}^{M} a_m y_{i+m} = h \sum_{m=0}^{M} b_m f(t_{i+m}, y_{i+m})$$

Leapfrog method

$$y_{i+1} = y_{i-1} + 2hf(t_i, y_i)$$

Second order method.

Adams-Bashforth methods

$$y_{i+M} = y_{i+M-1} + h \sum_{m=0}^{M-1} b_m f(t_{i+m}, y_{i+m})$$

M-step method.

Adams-Moulton methods

$$y_{i+M} = y_{i+M-1} + h \sum_{m=0}^{M} b_m f(t_{i+m}, y_{i+m})$$

Backward differentiation formulae

$$\sum_{m=0}^{M} a_m y_{i+m} = h b_M f(t_{i+M}, y_{i+M})$$