MA579H Scientific Computing

Numerics of first order ODEs - I

Rafikul Alam
Department of Mathematics
IIT Guwahati

Lecture outline

- Numerical differentiation
- Euler method for ODE

Numerical differentiation

Problem: Given smooth function $f : \mathbb{R} \longrightarrow \mathbb{R}$, compute its first and second derivatives at a point $x \in \mathbb{R}$:

Numerical differentiation

Problem: Given smooth function $f : \mathbb{R} \longrightarrow \mathbb{R}$, compute its first and second derivatives at a point $x \in \mathbb{R}$:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \text{ and } f''(x) = \lim_{h \to 0} \frac{f'(x+h) - f'(x)}{h}$$

- Differentiation is inherently sensitive, as small perturbations in data can cause large changes in result.

Numerical differentiation

Problem: Given smooth function $f : \mathbb{R} \longrightarrow \mathbb{R}$, compute its first and second derivatives at a point $x \in \mathbb{R}$:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \text{ and } f''(x) = \lim_{h \to 0} \frac{f'(x+h) - f'(x)}{h}$$

- Differentiation is inherently sensitive, as small perturbations in data can cause large changes in result.
- If a function is known only at discrete set of points, then a good approach is to fit some smooth function to given data and then differentiate the approximating function.

Finite difference approximations

Let $h > 0$. Consider the expansions

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2} f''(\xi) \text{ for some } x < \xi < x+h, \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2} f''(\eta) \text{ for some } x-h < \eta < x. \end{aligned}$$

Finite difference approximations

Let $h > 0$. Consider the expansions

$$
\begin{aligned}
f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(\xi) \text{ for some } x < \xi < x+h, \\
f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(\eta) \text{ for some } x-h < \eta < x.
\end{aligned}
$$

Forward difference approximation:

$$
\frac{f(x+h) - f(x)}{h} = f'(x) + \underbrace{\frac{h}{2}f''(\xi)}_{\text{truncation error}} = f'(x) + \mathcal{O}(h).
$$

Finite difference approximations

Let $h > 0$. Consider the expansions

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2} f''(\xi) \text{ for some } x < \xi < x+h, \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2} f''(\eta) \text{ for some } x-h < \eta < x. \end{aligned}$$

Forward difference approximation:

$$\frac{f(x+h) - f(x)}{h} = f'(x) + \underbrace{\frac{h}{2} f''(\xi)}_{\text{truncation error}} = f'(x) + \mathcal{O}(h).$$

Backward difference approximation:

$$\frac{f(x) - f(x-h)}{h} = f'(x) - \underbrace{\frac{h}{2} f''(\xi)}_{\text{truncation error}} = f'(x) + \mathcal{O}(h).$$

Finite difference approximations (cont.)

Assuming that f is C^3, we have the expansions

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(\xi), \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(\eta). \end{aligned}$$

Finite difference approximations (cont.)

Assuming that f is C^3, we have the expansions

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(\xi), \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(\eta). \end{aligned}$$

Centered difference approximation:

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \underbrace{\frac{h^2}{6}f'''(\theta)}_{\text{truncation error}} = f'(x) + \mathcal{O}(h^2).$$

Finite difference approximations (cont.)

Now assume that f is C^4 and consider the expansions

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(\xi), \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(\eta). \end{aligned}$$

Finite difference approximations (cont.)

Now assume that f is C^4 and consider the expansions

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(\xi), \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(\eta). \end{aligned}$$

Then we have 3-point centered difference approximation of f'':

$$\frac{f(x+h) - 2f(x) + f(x-h)}{h^2} = f''(x) + \underbrace{\frac{h^2}{12}f^{(4)}(\theta)}_{\text{truncation error}} = f''(x) + \mathcal{O}(h^2).$$

Finite difference approximations (cont.)

Now assume that f is C^4 and consider the expansions

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(\xi), \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(\eta). \end{aligned}$$

Then we have 3-point centered difference approximation of f'':

$$\frac{f(x+h) - 2f(x) + f(x-h)}{h^2} = f''(x) + \underbrace{\frac{h^2}{12}f^{(4)}(\theta)}_{\text{truncation error}} = f''(x) + \mathcal{O}(h^2).$$

- Software packages are available implementing automatic differentiation (AD) of functions.

Example

Consider $f(x) = e^x$. Then $f'(0) = 1$. Forward difference approximation and centered difference approximation of $f'(0)$ is given in the following table.

h	$\frac{e^h - 1}{h} - 1$	$\frac{e^h - e^{-h}}{2h} - 1$
10^{-1}	5.17×10^{-2}	1.6×10^{-3}
10^{-2}	5.02×10^{-3}	1.67×10^{-5}
10^{-3}	5.0×10^{-4}	1.6×10^{-7}
10^{-4}	$5.0 \times 1o^{-5}$	1.67×10^{-9}
10^{-5}	5.0×10^{-6}	1.21×10^{-11}

Effect of rounding error

Define $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ and $E(x, h) := |f'(x) - D_h f(x)|$.

Effect of rounding error

Define $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ and $E(x,h) := |f'(x) - D_h f(x)|.$

Recall that $D_h f(x) = f'(x) + \mathcal{O}(h).$

Effect of rounding error

Define $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ and $E(x,h) := |f'(x) - D_h f(x)|$.

Recall that $D_h f(x) = f'(x) + \mathcal{O}(h)$. Hence, in theory,

$$|h| \text{ small } \implies E(x,h) \text{ is small.}$$

But is it true?

Effect of rounding error

Define $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ and $E(x,h) := |f'(x) - D_h f(x)|.$

Recall that $D_h f(x) = f'(x) + \mathcal{O}(h)$. Hence, in theory,

$$|h| \text{ small } \implies E(x,h) \text{ is small.}$$

But is it true? For $f(x) = e^x$, we plot $E(0,h)$ as a function of h.

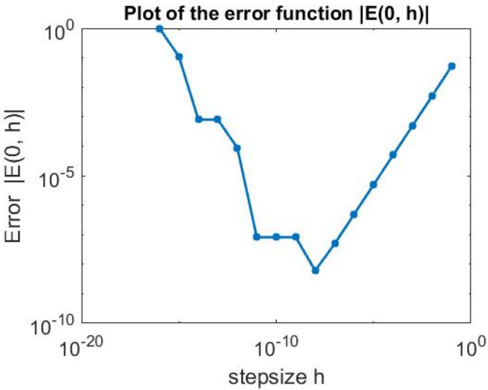
Effect of rounding error

Define $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ and $E(x,h) := |f'(x) - D_h f(x)|.$

Recall that $D_h f(x) = f'(x) + \mathcal{O}(h)$. Hence, in theory,

$$|h| \text{ small } \implies E(x,h) \text{ is small.}$$

But is it true? For $f(x) = e^x$, we plot $E(0,h)$ as a function of h.



Ordinary differential equations (ODEs)

Consider the initial value problem (IVP) associated with a first order ODE

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

where $x_0 \in [a, b]$ and $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$.

Ordinary differential equations (ODEs)

Consider the initial value problem (IVP) associated with a first order ODE

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

where $x_0 \in [a, b]$ and $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$.

The equation is in normal form. It is said to be autonomous if $f(x, y)$ is independent of x as in

$$\frac{dy}{dx} = cy(1 - y), \; y(0) = 1.$$

Ordinary differential equations (ODEs)

Consider the initial value problem (IVP) associated with a first order ODE

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

where $x_0 \in [a, b]$ and $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$.

The equation is in normal form. It is said to be autonomous if $f(x, y)$ is independent of x as in

$$\frac{dy}{dx} = cy(1 - y), \; y(0) = 1.$$

The equation specifies the slope field or direction field of the integral curve/solution curve at each point (x, y) as $f(x, y)$.

Ordinary differential equations (ODEs)

Consider the initial value problem (IVP) associated with a first order ODE

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

where $x_0 \in [a, b]$ and $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$.

The equation is in normal form. It is said to be autonomous if $f(x, y)$ is independent of x as in

$$\frac{dy}{dx} = cy(1 - y), \; y(0) = 1.$$

The equation specifies the slope field or direction field of the integral curve/solution curve at each point (x, y) as $f(x, y)$.

The direction or slope field of the ODE is obtained by affixing an arrow representing the vector $\left(\frac{1}{\sqrt{1+(f(x,y))^2}}, \frac{f(x,y)}{\sqrt{1+(f(x,y))^2}}\right)$ at each point (x, y) on a grid with $x \in [a, b]$. It is a graphical tool to analyze the solution of the IVP.

Direction fields of ODEs

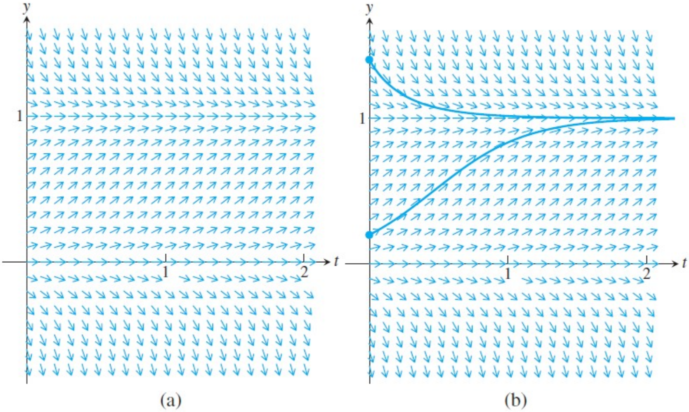


Figure : Direction field for the autonomous ODE $\frac{dy}{dt} = y(1-y)$ showing (a) no variation with t and (b) plotted with two solutions.

Direction fields of ODEs

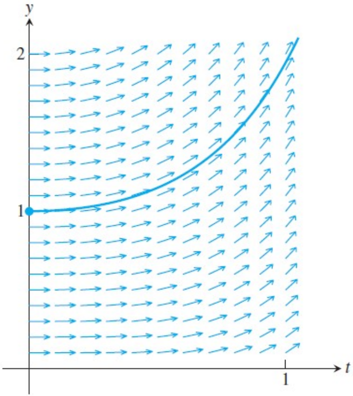
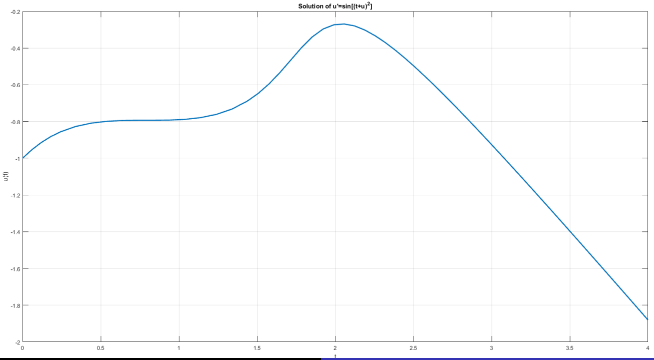


Figure : Direction field for the non-autonomous ODE $\frac{dy}{dt} = ty + y^3$ showing the solution $y(t) = 3e^{t^2/2} - t^2 - 2$ satisfying $y(0) = 1$.

Example

Consider the equation $u' = \sin((u+t)^2), u(0) = -1$. The MATLAB built-in function ode45 solves the ode numerically.

```
f = @(t,u) sin( (t+u).^2 );
[t,u] = ode45(f,[0,4],-1);
plot(t,u, 'LineWidth', 2), grid on
xlabel('t'), ylabel('u(t)')
title('Solution of u'' = sin[(t+u)^2]')
```



Ordinary differential equations (ODEs)

The initial value problem (IVP)

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

has a unique solution if $|f(x, y_1) - f(x, y_2)| \leq K|y_1 - y_2|$ for all $a \leq x \leq b$, where K is a constant independent of x.

Ordinary differential equations (ODEs)

The initial value problem (IVP)

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

has a unique solution if $|f(x,y_1) - f(x,y_2)| \leq K|y_1 - y_2|$ for all $a \leq x \leq b$, where K is a constant independent of x.

Example: Consider $f(x,y) := -y + x$. Then $y' = -y + x, \; y(0) = 1$, has a unique solution $y = x - 1 + 2e^{-x}$. ■

Ordinary differential equations (ODEs)

The initial value problem (IVP)

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

has a unique solution if $|f(x, y_1) - f(x, y_2)| \leq K|y_1 - y_2|$ for all $a \leq x \leq b$, where K is a constant independent of x.

Example: Consider $f(x, y) := -y + x$. Then $y' = -y + x, \; y(0) = 1$, has a unique solution $y = x - 1 + 2e^{-x}$. ■

- It is not expected to determine the solution y of the IVP as a function of x. Instead, we construct a table of function values

x_0	x_1	x_2	$\dots$	x_m
y_0	y_1	y_2	$\dots$	y_m

.

Ordinary differential equations (ODEs)

The initial value problem (IVP)

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

has a unique solution if $|f(x, y_1) - f(x, y_2)| \leq K|y_1 - y_2|$ for all $a \leq x \leq b$, where K is a constant independent of x.

Example: Consider $f(x, y) := -y + x$. Then $y' = -y + x, \; y(0) = 1$, has a unique solution $y = x - 1 + 2e^{-x}$. ■

- It is not expected to determine the solution y of the IVP as a function of x. Instead, we construct a table of function values

x_0	x_1	x_2	$\dots$	x_m
y_0	y_1	y_2	$\dots$	y_m

.

Most numerical methods for solving ODEs produce such a table.

Example

Compute solution of $u' = sin((u+t)^2), u(0) = -1$ at data points

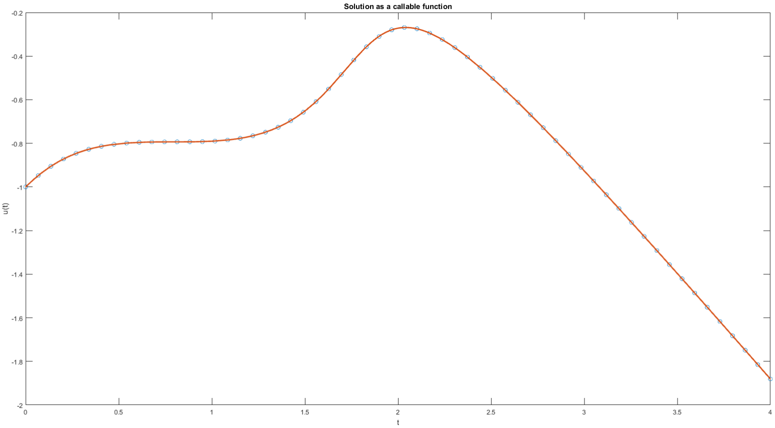
```
f = @(t,u) sin( (t+u).^2 );
t = linspace(0,4,60)';
[t,u] = ode45(f,t,-1);
plot(t,u,'o'), hold on
xlabel('t'), ylabel('u(t)')
title('Solution at 60 points')
```

Also, we can create a callable function for the solution. This is essentially a high-quality interpolant of the computed values.

```
u = @(t) deval( ode45(f,[0,4],-1), t );
fplot(u,[0,4], 'LineWidth', 2)
xlabel('t'), ylabel('u(t)')
title('Solution as a callable function')
```

Example

Computes solution of $u' = sin((u+t)^2), u(0) = -1$ at data points



Forward Euler's method

Consider the IVP $y' = f(x, y),\ y(x_0) = y_0$.

Forward Euler's method

Consider the IVP $y' = f(x, y), \; y(x_0) = y_0$.

Suppose that we want to approximate the solution $y(x)$ at the nodes $x_j = x_0 + jh$ for $j = 0 : n$, where $h > 0$ is the stepsize.

Forward Euler's method

Consider the IVP $y' = f(x, y),\; y(x_0) = y_0$.

Suppose that we want to approximate the solution $y(x)$ at the nodes $x_j = x_0 + jh$ for $j = 0 : n$, where $h > 0$ is the stepsize.

Notation: We denote by y_j the approximation of $y(x_j)$ for $j = 1 : n$.

Forward Euler's method

Consider the IVP $y' = f(x, y), \; y(x_0) = y_0$.

Suppose that we want to approximate the solution $y(x)$ at the nodes $x_j = x_0 + jh$ for $j = 0 : n$, where $h > 0$ is the stepsize.

Notation: We denote by y_j the approximation of $y(x_j)$ for $j = 1 : n$.

Forward Euler's method: Forward difference approximation of y' and the sampling of $f(x, y)$ at the node x_j

$$\frac{y(x_{j+1}) - y(x_j)}{h} = y'(x_j) + \mathcal{O}(h)$$

yield forward Euler's method

Forward Euler's method

Consider the IVP $y' = f(x, y),\ y(x_0) = y_0$.

Suppose that we want to approximate the solution $y(x)$ at the nodes $x_j = x_0 + jh$ for $j = 0 : n$, where $h > 0$ is the stepsize.

Notation: We denote by y_j the approximation of $y(x_j)$ for $j = 1 : n$.

Forward Euler's method: Forward difference approximation of y' and the sampling of $f(x, y)$ at the node x_j

$$\frac{y(x_{j+1}) - y(x_j)}{h} = y'(x_j) + \mathcal{O}(h)$$

yield forward Euler's method

$$y_{j+1} = y_j + hf(x_j, y_j), \quad j = 0 : n - 1.$$

Forward Euler's method

Consider the IVP $y' = f(x, y),\ y(x_0) = y_0$.

Suppose that we want to approximate the solution $y(x)$ at the nodes $x_j = x_0 + jh$ for $j = 0 : n$, where $h > 0$ is the stepsize.

Notation: We denote by y_j the approximation of $y(x_j)$ for $j = 1 : n$.

Forward Euler's method: Forward difference approximation of y' and the sampling of $f(x, y)$ at the node x_j

$$\frac{y(x_{j+1}) - y(x_j)}{h} = y'(x_j) + \mathcal{O}(h)$$

yield forward Euler's method

$$y_{j+1} = y_j + hf(x_j, y_j), \quad j = 0 : n - 1.$$

Forward Euler's method is an explicit method as y_{j+1} explicitly depends on $y_j,\ j = 0 : n - 1$.

Forward Euler's method

Consider the IVP $y' = f(x, y), \; y(x_0) = y_0$.

Suppose that we want to approximate the solution $y(x)$ at the nodes $x_j = x_0 + jh$ for $j = 0 : n$, where $h > 0$ is the stepsize.

Notation: We denote by y_j the approximation of $y(x_j)$ for $j = 1 : n$.

Forward Euler's method: Forward difference approximation of y' and the sampling of $f(x, y)$ at the node x_j

$$\frac{y(x_{j+1}) - y(x_j)}{h} = y'(x_j) + \mathcal{O}(h)$$

yield forward Euler's method

$$y_{j+1} = y_j + hf(x_j, y_j), \;\; j = 0 : n - 1.$$

Forward Euler's method is an explicit method as y_{j+1} explicitly depends on $y_j, \; j = 0 : n - 1$.

It uses the direction field at (x_j, y_j) to determine y_{j+1} by the formula

Change in y = step-size x slope at current position

Example

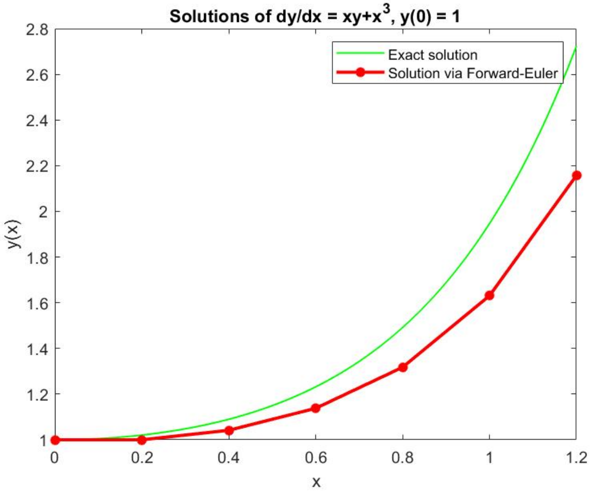


Figure : Plot of exact solution $y(x) = 3e^{x^2/2} - x^2 - 2$ of the non-autonomous ODE $\frac{dy}{dx} = xy + x^3$ satisfying $y(0) = 1$ along with solution via forward Euler method with $h = 0.2$.

Backward Euler's method

Backward difference approximation of y' and the sampling of $f(x, y)$ at the node x_{j+1}

$$\frac{y(x_{j+1}) - y(x_j)}{h} = y'(x_{j+1}) + \mathcal{O}(h)$$

yields backward Euler's method

Backward Euler's method

Backward difference approximation of y' and the sampling of $f(x, y)$ at the node x_{j+1}

$$\frac{y(x_{j+1}) - y(x_j)}{h} = y'(x_{j+1}) + \mathcal{O}(h)$$

yields backward Euler's method

$$y_{j+1} = y_j + hf(x_{j+1}, y_{j+1}) \text{ for } j = 0 : n - 1.$$

Backward Euler's method

Backward difference approximation of y' and the sampling of $f(x, y)$ at the node x_{j+1}

$$\frac{y(x_{j+1}) - y(x_j)}{h} = y'(x_{j+1}) + \mathcal{O}(h)$$

yields backward Euler's method

$$y_{j+1} = y_j + hf(x_{j+1}, y_{j+1}) \text{ for } j = 0 : n-1.$$

Backward Euler's method is an implicit method as y_{j+1} implicitly depends on y_{j+1}, $j = 0 : n-1$.

Backward Euler's method

Backward difference approximation of y' and the sampling of $f(x, y)$ at the node x_{j+1}

$$\frac{y(x_{j+1}) - y(x_j)}{h} = y'(x_{j+1}) + \mathcal{O}(h)$$

yields backward Euler's method

$$y_{j+1} = y_j + hf(x_{j+1}, y_{j+1}) \text{ for } j = 0 : n - 1.$$

Backward Euler's method is an implicit method as y_{j+1} implicitly depends on $y_{j+1}, \; j = 0 : n - 1$.It uses the direction field at (x_{j+1}, y_{j+1}) to determine y_{j+1} via

Change in y = step-size x slope at next position

In backward Euler's method, we have to solve the nonlinear equation $y_{j+1} = y_j + hf(x_{j+1}, y_{j+1})$ for y_{j+1}, which would require extra work.

Backward Euler's method

Backward difference approximation of y' and the sampling of $f(x, y)$ at the node x_{j+1}

$$\frac{y(x_{j+1}) - y(x_j)}{h} = y'(x_{j+1}) + \mathcal{O}(h)$$

yields backward Euler's method

$$y_{j+1} = y_j + hf(x_{j+1}, y_{j+1}) \text{ for } j = 0 : n-1.$$

Backward Euler's method is an implicit method as y_{j+1} implicitly depends on y_{j+1}, $j = 0 : n-1$.It uses the direction field at (x_{j+1}, y_{j+1}) to determine y_{j+1} via

Change in y = step-size x slope at next position

In backward Euler's method, we have to solve the nonlinear equation $y_{j+1} = y_j + hf(x_{j+1}, y_{j+1})$ for y_{j+1}, which would require extra work.

Example: Consider $y' = 2\sin(xy)$, $y(x_0) = y_0$. The first step in forward Euler's method gives $y_1 = y_0 + 2\sin(x_0 y_0)$.

Backward Euler's method

Backward difference approximation of y' and the sampling of $f(x, y)$ at the node x_{j+1}

$$\frac{y(x_{j+1}) - y(x_j)}{h} = y'(x_{j+1}) + \mathcal{O}(h)$$

yields backward Euler's method

$$y_{j+1} = y_j + hf(x_{j+1}, y_{j+1}) \text{ for } j = 0 : n-1.$$

Backward Euler's method is an implicit method as y_{j+1} implicitly depends on $y_{j+1},\ j = 0 : n-1$.It uses the direction field at (x_{j+1}, y_{j+1}) to determine y_{j+1} via

Change in y = step-size x slope at next position

In backward Euler's method, we have to solve the nonlinear equation $y_{j+1} = y_j + hf(x_{j+1}, y_{j+1})$ for y_{j+1}, which would require extra work.

Example: Consider $y' = 2\sin(xy),\ y(x_0) = y_0$. The first step in forward Euler's method gives $y_1 = y_0 + 2\sin(x_0 y_0)$.

The first step in backward Euler's method yields $y_1 = y_0 + 2\sin(x_1 y_1)$. To compute y_1, solve $y_0 + 2\sin(x_1 y_1) - y_1 = 0$ by a root finding method.

Example

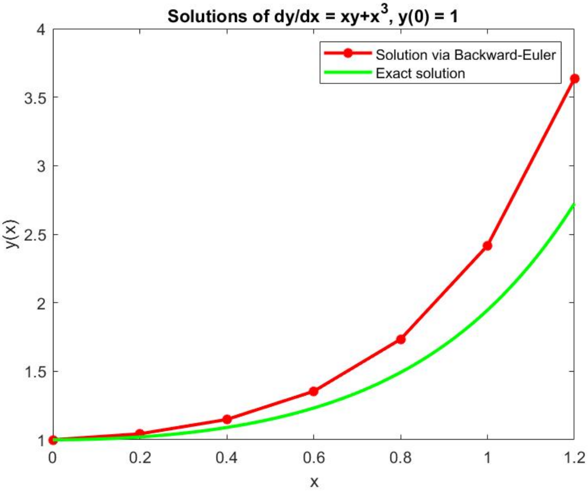


Figure : Plot of exact solution $y(x) = 3e^{x^2/2} - x^2 - 2$ of the non-autonomous ODE $\frac{dy}{dx} = xy + x^3$ satisfying $y(0) = 1$ along with solution via backward Euler method with $h = 0.2$.

Forward Euler's method via quadrature

Again, consider the IVP $y' = f(x, y), \;\; y(x_0) = y_0.$

Forward Euler's method via quadrature

Again, consider the IVP $y' = f(x, y), \;\; y(x_0) = y_0$.

Integrating $y' = f(x, y)$ in $[x_j, x_{j+1}]$, we have

$$y(x_{j+1}) - y(x_j) = \int_{x_j}^{x_{j+1}} f(t, y(t))dt.$$

Forward Euler's method via quadrature

Again, consider the IVP $y' = f(x, y), \;\; y(x_0) = y_0$.

Integrating $y' = f(x, y)$ in $[x_j, x_{j+1}]$, we have

$$y(x_{j+1}) - y(x_j) = \int_{x_j}^{x_{j+1}} f(t, y(t))dt.$$

By Newton-Cotes quadrature with single node x_j, we have

$$\int_{x_j}^{x_{j+1}} f(t, y(t))dt \approx (x_{j+1} - x_j)f(x_j, y(x_j)) = hf(x_j, y(x_j)).$$

Forward Euler's method via quadrature

Again, consider the IVP $y' = f(x, y), \;\; y(x_0) = y_0$.

Integrating $y' = f(x, y)$ in $[x_j, x_{j+1}]$, we have

$$y(x_{j+1}) - y(x_j) = \int_{x_j}^{x_{j+1}} f(t, y(t))dt.$$

By Newton-Cotes quadrature with single node x_j, we have

$$\int_{x_j}^{x_{j+1}} f(t, y(t))dt \approx (x_{j+1} - x_j) f(x_j, y(x_j)) = hf(x_j, y(x_j)).$$

This yields the forward Euler's method

$$y_{j+1} = y_j + hf(x_j, y_j), \;\; j = 0 : n-1.$$

Backward Euler's method via quadrature

Recall that
$$y(x_{j+1}) - y(x_j) = \int_{x_j}^{x_{j+1}} f(t, y(t))dt.$$

Backward Euler's method via quadrature

Recall that

$$y(x_{j+1}) - y(x_j) = \int_{x_j}^{x_{j+1}} f(t, y(t))dt.$$

Again, Newton-Cotes quadrature with single node x_{j+1} gives

$$\int_{x_j}^{x_{j+1}} f(t, y(t))dt \approx (x_{j+1} - x_j)f(x_{j+1}, y(x_{j+1})) = hf(x_{j+1}, y(x_{j+1})).$$

Backward Euler's method via quadrature

Recall that

$$y(x_{j+1}) - y(x_j) = \int_{x_j}^{x_{j+1}} f(t, y(t))dt.$$

Again, Newton-Cotes quadrature with single node x_{j+1} gives

$$\int_{x_j}^{x_{j+1}} f(t, y(t))dt \approx (x_{j+1} - x_j)f(x_{j+1}, y(x_{j+1})) = hf(x_{j+1}, y(x_{j+1})).$$

This yields the backward Euler's method

$$y_{j+1} = y_j + hf(x_{j+1}, y_{j+1}), \;\; j = 0 : n-1.$$

***