§5.2 Euler's Method: Initial value ODE to solve

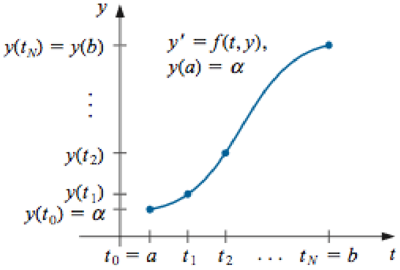
$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha.$$

§5.2 Euler's Method: Initial value ODE to solve

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

- Choose positive integer N, and select $\boxed{\textbf{mesh points}}$

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

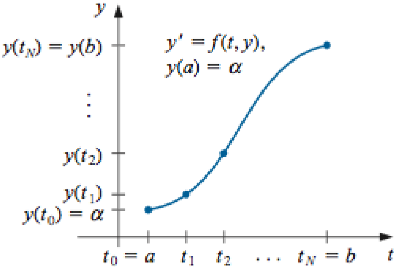


§5.2 Euler's Method: Initial value ODE to solve

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

- Choose positive integer N, and select $\boxed{\textbf{mesh points}}$

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$



https://en.wikipedia.org/wiki/Euler_method

Euler's Method for: $\frac{dy}{dt} = f(t, y), \quad a \leq t \leq b, \;\; y(a) = \alpha.$

Euler's Method for: $\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \;\; y(a) = \alpha.$

- Choose positive integer N, and select mesh points

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

Euler's Method for: $\frac{dy}{dt} = f(t, y), \quad a \leq t \leq b, \; y(a) = \alpha.$

- Choose positive integer N, and select mesh points

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

- For each j, do 2-term Taylor expansion

$$y(t_{j+1}) = y(t_j) + h\,\underline{y'(t_j)} + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1.$$

Euler's Method for: $\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \ \ y(a) = \alpha.$

- Choose positive integer N, and select mesh points

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

- For each j, do 2-term Taylor expansion

$$y(t_{j+1}) = y(t_j) + h\,\underline{y'(t_j)} + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1.$$

- Because $y(t)$ satisfies ODE,

$$y(t_{j+1}) = y(t_j) + h\,\underline{f(t_j, y(t_j))} + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1.$$

Euler's Method for: $\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \;\; y(a) = \alpha.$

- Choose positive integer N, and select mesh points

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

- For each j, do 2-term Taylor expansion

$$y(t_{j+1}) = y(t_j) + h\,\underline{y'(t_j)} + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1.$$

- Because $y(t)$ satisfies ODE,

$$y(t_{j+1}) = y(t_j) + h\,\underline{f(t_j, y(t_j))} + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1.$$

- Ignore error term, set $w_0 = \alpha$,

$$w_{j+1} = w_j + h\,f(t_j, w_j), \quad j = 0, 1, \cdots, N-1.$$

Euler's Method for: $\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \;\; y(a) = \alpha.$

- Choose positive integer N, and select mesh points

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

- For each j, do 2-term Taylor expansion

$$y(t_{j+1}) = y(t_j) + h\,\underline{y'(t_j)} + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1.$$

- Because $y(t)$ satisfies ODE,

$$y(t_{j+1}) = y(t_j) + h\,\underline{f(t_j, y(t_j))} + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1.$$

- Ignore error term, set $w_0 = \alpha$,

$$w_{j+1} = w_j + h\,f(t_j, w_j), \quad j = 0, 1, \cdots, N-1.$$

- Does it work?

Euler's Method for: $\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \;\; y(a) = \alpha.$

- Choose positive integer N, and select mesh points

  $$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

- For each j, do 2-term Taylor expansion

  $$y(t_{j+1}) = y(t_j) + h\,\underline{y'(t_j)} + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1.$$

- Because $y(t)$ satisfies ODE,

  $$y(t_{j+1}) = y(t_j) + h\,\underline{f(t_j, y(t_j))} + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1.$$

- Ignore error term, set $w_0 = \alpha$,

  $$w_{j+1} = w_j + h\,f(t_j, w_j), \quad j = 0, 1, \cdots, N-1.$$

- Does it work?

With the blessing of math god $\cdots$

Euler, 1707–1783

- Euler's number (base of natural log)

$$e = 2.71828\cdots$$

- Euler's identity

$$e^{i\,\pi} + 1 = 0$$

- Euler's formula

$$e^{i\,\phi} = \cos\,\phi + i\,\sin\,\phi.$$

Example ODE: $\frac{dy}{dt} = y - t^2 + 1, \ \ 0 \le t \le 2, \ \ y(0) = 0.5$

- Exact solution:

$$y(t) = (1+t)^2 - 0.5\, e^t$$

- Choose positive integer $N = 10$, so

$$h = 0.2, \quad t_j = 0.2\, j, \quad \text{for} \quad j = 0, 1, 2, \cdots 10.$$

Example ODE: $\frac{dy}{dt} = y - t^2 + 1, \;\; 0 \le t \le 2, \;\; y(0) = 0.5$

- Exact solution:

$$y(t) = (1+t)^2 - 0.5\, e^t$$

- Choose positive integer $N = 10$, so

$$h = 0.2, \quad t_j = 0.2\, j, \quad \text{for} \quad j = 0, 1, 2, \cdots 10.$$

- Set $w_0 = 0.5$. For $j = 0, 1, \cdots, 9$,

$$\begin{aligned} w_{j+1} &= w_j + h\,(w_j - t_j^2 + 1) \\ &= w_j + 0.2(w_i - 0.04j^2 + 1) \\ &= 1.2w_j - 0.008j^2 + 0.2. \end{aligned}$$

Example ODE: $\frac{dy}{dt} = y - t^2 + 1, \;\; 0 \le t \le 2, \;\; y(0) = 0.5$

- Exact solution:

$$y(t) = (1+t)^2 - 0.5\, e^t$$

- Choose positive integer $N = 10$, so

$$h = 0.2, \quad t_j = 0.2\, j, \quad \text{for} \quad j = 0, 1, 2, \cdots 10.$$

- Set $w_0 = 0.5$. For $j = 0, 1, \cdots, 9$,

$$\begin{aligned} w_{j+1} &= w_j + h\,(w_j - t_j^2 + 1) \\ &= w_j + 0.2(w_i - 0.04j^2 + 1) \\ &= 1.2w_j - 0.008j^2 + 0.2. \end{aligned}$$

t_i	w_i	$y_i = y(t_i)$	$\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0.0000000
0.2	0.8000000	0.8292986	0.0292986
0.4	1.1520000	1.2140877	0.0620877
0.6	1.5504000	1.6489406	0.0985406
0.8	1.9884800	2.1272295	0.1387495
1.0	2.4581760	2.6408591	0.1826831
1.2	2.9498112	3.1799415	0.2301303
1.4	3.4517734	3.7324000	0.2806266
1.6	3.9501281	4.2834838	0.3333557
1.8	4.4281538	4.8151763	0.3870225
2.0	4.8657845	5.3054720	0.4396874

Example ODE: $\frac{dy}{dt} = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5$

- Exact solution:

$$y(t) = (1+t)^2 - 0.5\, e^t$$

- Choose positive integer $N = 10$, so

$$h = 0.2, \quad t_j = 0.2\, j, \quad \text{for} \quad j = 0, 1, 2, \cdots 10.$$

- Set $w_0 = 0.5$. For $j = 0, 1, \cdots, 9$,

$$\begin{aligned} w_{j+1} &= w_j + h\,(w_j - t_j^2 + 1) \\ &= w_j + 0.2(w_i - 0.04j^2 + 1) \\ &= 1.2w_j - 0.008j^2 + 0.2. \end{aligned}$$

t_i	w_i	$y_i = y(t_i)$	$\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0.0000000
0.2	0.8000000	0.8292986	0.0292986
0.4	1.1520000	1.2140877	0.0620877
0.6	1.5504000	1.6489406	0.0985406
0.8	1.9884800	2.1272295	0.1387495
1.0	2.4581760	2.6408591	0.1826831
1.2	2.9498112	3.1799415	0.2301303
1.4	3.4517734	3.7324000	0.2806266
1.6	3.9501281	4.2834838	0.3333557
1.8	4.4281538	4.8151763	0.3870225
2.0	4.8657845	5.3054720	0.4396874

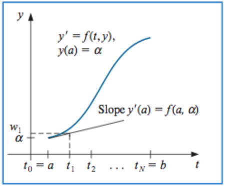
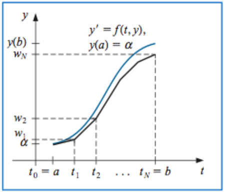


Figure 5.4



Theorem: Suppose that in the initial value ODE,

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

- $f(t, y)$ is continuous,
- $f(t, y)$ satisfies Lipschitz condition

$$\begin{array}{rcl} |f(t, y_1) - f(t, y_2)| & \le & L\,|y_1 - y_2| \quad \text{on domain} \\ D & = & \{(t, y) \quad | \quad a \le t \le b, \quad -\infty < y < \infty\}. \end{array}$$

Let $w_0, w_1, \cdots, w_N$ be the approximations generated by Euler's method for some positive integer N. Then for each $j = 0, 1, \cdots, N$,

$$|y(t_j) - w_j| \le \frac{hM}{2L}\left(e^{L(t_j - a)} - 1\right),$$

where $h = (b-a)/N, \quad t_j = a + j\,h$, $M = \mathbf{max}_{t \in [a,b]}\,|y''(t)|$.

Theorem is a mixed bag

Theorem:

$$|y(t_j) - w_j| \leq \frac{hM}{2L}\left(e^{L(t_j-a)} - 1\right), \tag{1}$$

where $h = (b-a)/N, \quad t_j = a + j\,h$, $M = \mathbf{max}_{t\in[a,b]}\,|y''(t)|$.

Theorem is a mixed bag

Theorem:

$$|y(t_j) - w_j| \leq \frac{hM}{2L}\left(e^{L(t_j - a)} - 1\right), \tag{1}$$

where $h = (b-a)/N, \quad t_j = a + j\,h,\ M = \mathbf{max}_{t\in[a,b]}\,|y''(t)|.$

$\boxed{\text{How good is Theorem?}}$

Theorem is a mixed bag

Theorem:

$$|y(t_j) - w_j| \leq \frac{hM}{2L}\left(e^{L(t_j-a)} - 1\right), \qquad (1)$$

where $h = (b-a)/N, \quad t_j = a + j\,h,\ M = \mathbf{max}_{t\in[a,b]}\,|y''(t)|.$

$\boxed{\text{How good is Theorem?}}$

- **Good News**: Euler's method indeed converges:

$$\lim_{N\to\infty} \frac{hM}{2L}\left(e^{L(t_j-a)} - 1\right) = 0. \qquad (2)$$

Theorem is a mixed bag

Theorem:

$$|y(t_j) - w_j| \le \frac{hM}{2L}\left(e^{L(t_j-a)} - 1\right), \qquad (1)$$

where $h = (b-a)/N, \quad t_j = a + j\,h,\ M = \mathbf{max}_{t\in[a,b]}\,|y''(t)|$.

$\boxed{\text{How good is Theorem?}}$

- **Good News**: Euler's method indeed converges:

$$\lim_{N\to\infty} \frac{hM}{2L}\left(e^{L(t_j-a)} - 1\right) = 0. \qquad (2)$$

- **Bad News**: N in (2) likely too big for numerical convergence:

$$N \gg e^{L(t_N-a)} = e^{L(b-a)} > 4\times 10^{15} \approx \frac{1}{\text{eps}} \quad \text{if} \quad L > 6, \quad b-a > 6.$$

Theorem is a mixed bag

Theorem:

$$|y(t_j) - w_j| \leq \frac{hM}{2L}\left(e^{L(t_j - a)} - 1\right), \qquad (1)$$

where $h = (b-a)/N, \quad t_j = a + j\,h,\ M = \mathbf{max}_{t \in [a,b]} |y''(t)|.$

$\boxed{\text{How good is Theorem?}}$

- **Good News**: Euler's method indeed converges:

$$\lim_{N \to \infty} \frac{hM}{2L}\left(e^{L(t_j - a)} - 1\right) = 0. \qquad (2)$$

- **Bad News**: N in (2) likely too big for numerical convergence:

$$N \gg e^{L(t_N - a)} = e^{L(b-a)} > 4 \times 10^{15} \approx \frac{1}{\mathrm{eps}} \quad \text{if} \quad L > 6, \quad b - a > 6.$$

- **Worse**: ANY numerical solution can be hopeless for chaotic solutions (**butterfly effect**)

Theorem: Suppose that in the initial value ODE,

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

- $f(t, y)$ is continuous,
- $f(t, y)$ satisfies Lipschitz condition

$$\begin{array}{rcl} |f(t, y_1) - f(t, y_2)| & \le & L\,|y_1 - y_2| \quad \text{on domain} \\ D & = & \{(t, y) \quad | \quad a \le t \le b, \quad -\infty < y < \infty\}\,. \end{array}$$

Let $w_0, w_1, \cdots, w_N$ be the approximations generated by Euler's method for some positive integer N. Then for each $j = 0, 1, \cdots, N$,

$$|y(t_j) - w_j| \le \frac{hM}{2L}\left(e^{L(t_j - a)} - 1\right),$$

where $h = (b - a)/N, \quad t_j = a + j\,h$, $M = \mathbf{max}_{t \in [a,b]}\,|y''(t)|$.

Proof of **Theorem** (I)

- Mesh points

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

Proof of **Theorem** (I)

- Mesh points

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

- For each j, we have 2-term Taylor expansion

$$\begin{aligned} y(t_{j+1}) &= y(t_j) + h\,y'(t_j) + \frac{h^2}{2} y''(\xi_j) \\ &= y(t_j) + h\,f(t_j, y(t_j)) + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1. \end{aligned}$$

Proof of **Theorem** (I)

- Mesh points

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

- For each j, we have 2-term Taylor expansion

$$\begin{aligned} y(t_{j+1}) &= y(t_j) + h\,y'(t_j) + \frac{h^2}{2} y''(\xi_j) \\ &= y(t_j) + h\,f(t_j, y(t_j)) + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1. \end{aligned}$$

- Euler's method for each j

$$w_{j+1} = w_j + h\,f(t_j, w_j).$$

Proof of **Theorem** (I)

- Mesh points

$$t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$$

- For each j, we have 2-term Taylor expansion

$$\begin{aligned} y(t_{j+1}) &= y(t_j) + h\,y'(t_j) + \frac{h^2}{2} y''(\xi_j) \\ &= y(t_j) + h\,f(t_j, y(t_j)) + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1. \end{aligned}$$

- Euler's method for each j

$$w_{j+1} = w_j + h\,f(t_j, w_j).$$

- Subtraction of two equations:

$$y(t_{j+1}) - w_{j+1} = y(t_j) - w_j + h\left(f(t_j, y(t_j)) - f(t_j, w_j)\right) + \frac{h^2}{2} y''(\xi_j).$$

Proof of **Theorem** (II)

- Difference of two equations:

$$y(t_{j+1})-w_{j+1} = y(t_j)-w_j+h\left(f(t_j, y(t_j)) - f(t_j, w_j)\right)+\frac{h^2}{2}y''(\xi_j).$$

Proof of **Theorem** (II)

- Difference of two equations:

$$y(t_{j+1})-w_{j+1} = y(t_j)-w_j+h\left(f(t_j, y(t_j)) - f(t_j, w_j)\right)+\frac{h^2}{2}y''(\xi_j).$$

- This implies a linear recursion:

$$\begin{aligned} |y(t_{j+1}) - w_{j+1}| &\le |y(t_j) - w_j| + h\,|f(t_j, y(t_j)) - f(t_j, w_j)| + \frac{h^2}{2}\left|y''(\xi_j)\right| \\ &\le |y(t_j) - w_j| + hL\,|y(t_j) - w_j| + \frac{h^2}{2}\,M \\ &= (1 + hL)\,|y(t_j) - w_j| + \frac{h^2}{2}\,M. \end{aligned}$$

Proof of **Theorem** (II)

- Difference of two equations:

$$y(t_{j+1})-w_{j+1} = y(t_j)-w_j+h\left(f(t_j,y(t_j)) - f(t_j,w_j)\right)+\frac{h^2}{2}y''(\xi_j).$$

- This implies a linear recursion:

$$\begin{aligned}
|y(t_{j+1}) - w_{j+1}| &\leq |y(t_j) - w_j| + h\,|f(t_j,y(t_j)) - f(t_j,w_j)| + \frac{h^2}{2}\left|y''(\xi_j)\right| \\
&\leq |y(t_j) - w_j| + hL\,|y(t_j) - w_j| + \frac{h^2}{2}\,M \\
&= (1+hL)\,|y(t_j) - w_j| + \frac{h^2}{2}\,M.
\end{aligned}$$

- Further simplification for $j = 0, \cdots, N-1$,

$$\begin{aligned}
\underline{|y(t_{j+1}) - w_{j+1}| + \frac{hM}{2L}} &\leq (1+hL)\,|y(t_j) - w_j| + \frac{h^2}{2}\,M + \frac{hM}{2L} \\
&= (1+hL)\left(\underline{|y(t_j) - w_j| + \frac{hM}{2L}}\right).
\end{aligned}$$

Proof of **Theorem** (III)

- Solving linear recursion:

$$
\begin{aligned}
\underline{|y(t_{j+1}) - w_{j+1}| + \frac{hM}{2L}} &\leq (1+hL)\left(\underline{|y(t_j) - w_j| + \frac{hM}{2L}}\right) \\
&\leq (1+hL)^2\left(\underline{|y(t_{j-1}) - w_{j-1}| + \frac{hM}{2L}}\right) \\
&\vdots \\
&\leq (1+hL)^{j+1}\left(\underline{|y(t_0) - w_0| + \frac{hM}{2L}}\right).
\end{aligned}
$$

- Since $y(t_0) = w_0 = \alpha$,

$$|y(t_{j+1}) - w_{j+1}| + \frac{hM}{2L} \leq (1+hL)^{j+1}\,\frac{hM}{2L},$$

it follows that

$$
\begin{aligned}
|y(t_{j+1}) - w_{j+1}| &\leq \frac{hM}{2L}\left((1+hL)^{j+1} - 1\right) \\
&\leq \frac{hM}{2L}\left(e^{L(t_{j+1}-a)} - 1\right).
\end{aligned}
$$

Theorem: Suppose that in the initial value ODE,

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

- $f(t, y)$ is continuous,
- $f(t, y)$ satisfies Lipschitz condition

$$\begin{aligned} |f(t, y_1) - f(t, y_2)| &\leq L\,|y_1 - y_2| \quad \text{on domain} \\ D &= \{(t, y) \quad | \quad a \leq t \leq b, \quad -\infty < y < \infty\}. \end{aligned}$$

Let $w_0, w_1, \cdots, w_N$ be the approximations generated by Euler's method for some positive integer N. Then for each $j = 0, 1, \cdots, N$,

$$|y(t_j) - w_j| \leq \frac{hM}{2L}\left(e^{L(t_j - a)} - 1\right),$$

where $h = (b - a)/N, \quad t_j = a + j\,h$, $M = \mathbf{max}_{t \in [a,b]} |y''(t)|$.

Example ODE: $\frac{dy}{dt} = y - t^2 + 1, \;\; 0 \le t \le 2, \;\; y(0) = 0.5$

- Exact solution:

$$y(t) = (1+t)^2 - 0.5\, e^t$$

- Choose positive integer $N = 10$, so

$$h = 0.2, \quad t_j = 0.2\, j, \quad \text{for} \quad j = 0, 1, 2, \cdots 10.$$

Example ODE: $\frac{dy}{dt} = y - t^2 + 1, \ \ 0 \leq t \leq 2, \ \ y(0) = 0.5$

- Exact solution:

$$y(t) = (1+t)^2 - 0.5\, e^t$$

- Choose positive integer $N = 10$, so

$$h = 0.2, \quad t_j = 0.2\, j, \quad \text{for} \quad j = 0, 1, 2, \cdots 10.$$

- Set $w_0 = 0.5$. For $j = 0, 1, \cdots, 9$,

$$\begin{aligned} w_{j+1} &= w_j + h\,(w_j - t_j^2 + 1) \\ &= w_j + 0.2(w_i - 0.04 j^2 + 1) \\ &= 1.2 w_j - 0.008 j^2 + 0.2. \end{aligned}$$

Example ODE: $\frac{dy}{dt} = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5$

- Exact solution:

$$y(t) = (1+t)^2 - 0.5\, e^t$$

- Choose positive integer $N = 10$, so

$$h = 0.2, \quad t_j = 0.2\, j, \quad \text{for} \quad j = 0, 1, 2, \cdots 10.$$

- Set $w_0 = 0.5$. For $j = 0, 1, \cdots, 9$,

$$\begin{aligned} w_{j+1} &= w_j + h\,(w_j - t_j^2 + 1) \\ &= w_j + 0.2(w_i - 0.04 j^2 + 1) \\ &= 1.2 w_j - 0.008 j^2 + 0.2. \end{aligned}$$

- $y''(t) = 2 - 0.5\, e^t, \quad \frac{\partial f}{\partial y}(t, y) = 1,$

it follows that $\left|y''(t)\right| \le 0.5\, e^2 - 2 \overset{def}{=} M, \quad L = 1.$

$$|y(t_i) - w_i| \le \frac{hM}{2L}\left(e^{L(t_i - a)} - 1\right) = 0.1\,\left(0.5\, e^2 - 2\right)\,\left(e^{t_i} - 1\right).$$

Example ODE: $\frac{dy}{dt} = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5$

- Exact solution:

$$y(t) = (1+t)^2 - 0.5\, e^t$$

- Choose positive integer $N = 10$, so

$$h = 0.2, \quad t_j = 0.2\, j, \quad \text{for} \quad j = 0, 1, 2, \cdots 10.$$

- Set $w_0 = 0.5$. For $j = 0, 1, \cdots, 9$,

$$\begin{aligned} w_{j+1} &= w_j + h\,(w_j - t_j^2 + 1) \\ &= w_j + 0.2(w_i - 0.04j^2 + 1) \\ &= 1.2w_j - 0.008j^2 + 0.2. \end{aligned}$$

- $y''(t) = 2 - 0.5\, e^t, \quad \frac{\partial f}{\partial y}(t, y) = 1,$

it follows that $\left|y''(t)\right| \le 0.5\, e^2 - 2 \overset{def}{=} M, \quad L = 1.$

$$|y(t_i) - w_i| \le \frac{hM}{2L}\left(e^{L(t_i - a)} - 1\right) = 0.1\left(0.5\, e^2 - 2\right)\left(e^{t_i} - 1\right).$$

t_i	$\lvert y(t_i) - w_i \rvert$	Error Bound
0.200000	0.029300	0.037520
0.400000	0.062090	0.083340
0.600000	0.098540	0.139310
0.800000	0.138750	0.207670
1.000000	0.182680	0.291170
1.200000	0.230130	0.393150
1.400000	0.280630	0.517710
1.600000	0.333360	0.669850
1.800000	0.387020	0.855680
2.000000	0.439690	1.082640

Example ODE: $\frac{dy}{dt} = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5$

- Exact solution:

$$y(t) = (1+t)^2 - 0.5\,e^t$$

- Choose positive integer $N = 10$, so

$$h = 0.2, \quad t_j = 0.2\,j, \quad \text{for} \quad j = 0, 1, 2, \cdots 10.$$

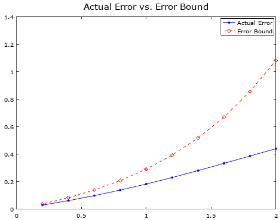
- Set $w_0 = 0.5$. For $j = 0, 1, \cdots, 9$,

$$\begin{aligned} w_{j+1} &= w_j + h\,(w_j - t_j^2 + 1) \\ &= w_j + 0.2(w_i - 0.04j^2 + 1) \\ &= 1.2w_j - 0.008j^2 + 0.2. \end{aligned}$$

- $y''(t) = 2 - 0.5\,e^t, \quad \frac{\partial f}{\partial y}(t, y) = 1,$

it follows that $\left|y''(t)\right| \le 0.5\,e^2 - 2 \stackrel{def}{=} M, \quad L = 1.$

t_i	$\lvert y(t_i) - w_i \rvert$	Error Bound
0.200000	0.029300	0.037520
0.400000	0.062090	0.083340
0.600000	0.098540	0.139310
0.800000	0.138750	0.207670
1.000000	0.182680	0.291170
1.200000	0.230130	0.393150
1.400000	0.280630	0.517710
1.600000	0.333360	0.669850
1.800000	0.387020	0.855680
2.000000	0.439690	1.082640



$$|y(t_i) - w_i| \le \frac{hM}{2L}\left(e^{L(t_i - a)} - 1\right) = 0.1\left(0.5\,e^2 - 2\right)\left(e^{t_i} - 1\right).$$

Euler's Method in Finite Precision (I)

- Mesh points $t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b - a)/N.$

Euler's Method in Finite Precision (I)

- Mesh points $t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$
- For each j, we have 2-term Taylor expansion

$$y(t_{j+1}) = y(t_j) + h\,y'(t_j) + \frac{h^2}{2}y''(\xi_j) == y(t_j) + h\,f(t_j, y(t_j)) + \frac{h^2}{2}y''(\xi_j), \quad j = 0, 1, \cdots, N-1$$

Euler's Method in Finite Precision (I)

- Mesh points $t_j = a + j\,h, \quad \text{for} \quad j = 0, 1, 2, \cdots N, \quad \text{where } h = (b-a)/N.$
- For each j, we have 2-term Taylor expansion

$$y(t_{j+1}) = y(t_j) + h\,y'(t_j) + \frac{h^2}{2} y''(\xi_j) == y(t_j) + h\,f(t_j, y(t_j)) + \frac{h^2}{2} y''(\xi_j), \quad j = 0, 1, \cdots, N-1$$

- Euler's method for each j, in finite precision

$$u_{j+1} = u_j + h\,f(t_j, u_j) + \delta_{j+1}, \quad \text{where } |\delta_{j+1}| \leq \delta$$

Euler's Method in Finite Precision (I)

- Mesh points $t_j = a + j\,h$, for $j = 0, 1, 2, \cdots N$, where $h = (b-a)/N$.
- For each j, we have 2-term Taylor expansion

$$y(t_{j+1}) = y(t_j) + h\,y'(t_j) + \frac{h^2}{2}y''(\xi_j) == y(t_j) + h\,f(t_j, y(t_j)) + \frac{h^2}{2}y''(\xi_j), \quad j = 0, 1, \cdots, N-1$$

- Euler's method for each j, in finite precision

$$u_{j+1} = u_j + h\,f(t_j, u_j) + \delta_{j+1}, \quad \text{where } |\delta_{j+1}| \le \delta$$

- Subtraction of two equations:

$$y(t_{j+1}) - u_{j+1} = y(t_j) - u_j + h\left(f(t_j, y(t_j)) - f(t_j, u_j)\right) + \frac{h^2}{2}y''(\xi_j) - \delta_{j+1}.$$

- This implies a linear recursion:

$$\begin{aligned} \left|y(t_{j+1}) - u_{j+1}\right| &\le \left|y(t_j) - u_j\right| + h\left|f(t_j, y(t_j)) - f(t_j, u_j)\right| + \frac{h^2}{2}\left|y''(\xi_j)\right| + |\delta_{j+1}| \\ &\le \left|y(t_j) - u_j\right| + hL\left|y(t_j) - u_j\right| + \frac{h^2}{2}M + \delta == (1 + hL)\left|y(t_j) - u_j\right| + \frac{h^2}{2}M + \delta. \end{aligned}$$

Euler's Method in Finite Precision (II)

- Further simplification for $j = 0, \cdots, N-1$,

$$\underline{\left|y(t_{j+1}) - u_{j+1}\right| + \frac{hM}{2L} + \frac{\delta}{hL}} \leq (1+hL)\left|y(t_j) - u_j\right| + \frac{h^2}{2}M + \boxed{\frac{hM}{2L}} + \delta + \boxed{\frac{\delta}{hL}}$$

$$= (1+hL)\left(\underline{\left|y(t_j) - u_j\right| + \frac{hM}{2L} + \frac{\delta}{hL}}\right) \leq (1+hL)^2\left(\underline{\left|y(t_{j-1}) - u_{j-1}\right| + \frac{hM}{2L} + \frac{\delta}{hL}}\right)$$

$$\vdots$$

$$\leq \quad (1+hL)^{j+1}\left(\underline{\left|y(t_0) - u_0\right| + \frac{hM}{2L} + \frac{\delta}{hL}}\right)$$

- Assume $|y(t_0) - u_0| = |\alpha - u_0| \leq \delta$,

$$\left|y(t_{j+1}) - u_{j+1}\right| + \frac{hM}{2L} + \frac{\delta}{hL} \leq (1+hL)^{j+1}\left(\delta + \frac{hM}{2L} + \frac{\delta}{hL}\right),$$

leading to error bound in finite precision

$$\begin{aligned}
\left|y(t_{j+1}) - u_{j+1}\right| \quad &\leq \quad \frac{1}{L}\left(\frac{hM}{2} + \frac{\delta}{h}\right)\left((1+hL)^{j+1} - 1\right) + \delta\,(1+hL)^{j+1} \\
&\leq \quad \frac{1}{L}\left(\frac{hM}{2} + \frac{\delta}{h}\right)\left(e^{L(t_{j+1}-a)} - 1\right) + \delta\, e^{L(t_{j+1}-a)}.
\end{aligned}$$

Euler's Method in Finite Precision (II)

- Further simplification for $j = 0, \cdots, N-1$,

$$\underline{\left|y(t_{j+1}) - u_{j+1}\right| + \frac{hM}{2L} + \frac{\delta}{hL}} \leq (1+hL)\left|y(t_j) - u_j\right| + \frac{h^2}{2}M + \boxed{\frac{hM}{2L}} + \delta + \boxed{\frac{\delta}{hL}}$$

$$= (1+hL)\left(\underline{\left|y(t_j) - u_j\right| + \frac{hM}{2L} + \frac{\delta}{hL}}\right) \leq (1+hL)^2\left(\underline{\left|y(t_{j-1}) - u_{j-1}\right| + \frac{hM}{2L} + \frac{\delta}{hL}}\right)$$

$$\vdots$$

$$\leq \quad (1+hL)^{j+1}\left(\underline{|y(t_0) - u_0| + \frac{hM}{2L} + \frac{\delta}{hL}}\right)$$

- Assume $|y(t_0) - u_0| = |\alpha - u_0| \leq \delta$,

$$\left|y(t_{j+1}) - u_{j+1}\right| + \frac{hM}{2L} + \frac{\delta}{hL} \leq (1+hL)^{j+1}\left(\delta + \frac{hM}{2L} + \frac{\delta}{hL}\right),$$

leading to error bound in finite precision

$$\begin{aligned}\left|y(t_{j+1}) - u_{j+1}\right| \quad &\leq \quad \frac{1}{L}\left(\frac{hM}{2} + \frac{\delta}{h}\right)\left((1+hL)^{j+1} - 1\right) + \delta\,(1+hL)^{j+1} \\ &\leq \quad \frac{1}{L}\left(\frac{hM}{2} + \frac{\delta}{h}\right)\left(e^{L(t_{j+1}-a)} - 1\right) + \delta\, e^{L(t_{j+1}-a)}.\end{aligned}$$

- "Optimal" stepsize h (for accuracy only): $h_{\mathbf{opt}} = \sqrt{\frac{2\delta}{M}} = O\left(\sqrt{\delta}\right)$, with $\left(\frac{h_{\mathbf{opt}}M}{2} + \frac{\delta}{h_{\mathbf{opt}}}\right) = \sqrt{2\delta\, M}$

Local Truncation Error for a general difference method

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

Definition: The difference method

$$\begin{aligned} w_0 &= \alpha, \\ w_{j+1} &= w_j + h\phi\left(t_j, w_j\right), \quad \text{for} \quad j = 0, 1, \cdots, N-1 \end{aligned}$$

has **local truncation error**

$$\begin{aligned} \tau_{j+1}(h) &\stackrel{def}{=} \frac{y(t_{j+1}) - \left(y(t_j) + h\,\phi\left(t_j, y(t_j)\right)\right)}{h} \\ &= \frac{y(t_{j+1}) - y(t_j)}{h} - \phi\left(t_j, y(t_j)\right). \end{aligned}$$

$\boxed{\textbf{LTE}\text{: additional error at step } j \text{ } \underline{\text{IF}} \text{ } w_j = y(t_j)}$

Local Truncation Error for Euler's method

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

Euler's method

$$\begin{aligned} w_0 &= \alpha, \\ w_{j+1} &= w_j + hf\left(t_j, w_j\right), \quad \text{for} \quad j = 0, 1, \cdots, N-1 \end{aligned}$$

has **local truncation error**

$$\begin{aligned} \tau_{j+1}(h) &= \frac{y(t_{j+1}) - y(t_j)}{h} - f\left(t_j, y(t_j)\right) \\ &= \frac{y(t_{j+1}) - y(t_j)}{h} - y'(t_j) = \frac{h}{2} y''(\xi_j), \quad \xi_j \in (t_j, tj+1). \end{aligned}$$

This implies

$$\left|\tau_{j+1}(h)\right| \le \frac{h}{2} \mathbf{max}_{t\in[a,b]} \left|y''(t)\right| \stackrel{def}{=} \frac{h\,M}{2}, \quad \text{a first order method.}$$

§5.3 2nd Order Taylor Method:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

§5.3 2nd Order Taylor Method:

$$\frac{dy}{dt} = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha. \quad (1)$$

- 3-term Taylor expansion

$$y(t_{j+1}) = y(t_j) + hy'(t_j) + \frac{h^2}{2}y''(t_j) + \frac{h^3}{3!}y^{(3)}(\xi_j) \quad (2)$$

§5.3 2nd Order Taylor Method:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- 3-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2} y''(t_j) + \frac{h^3}{3!} y^{(3)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)) \quad (3)$$

§5.3 2nd Order Taylor Method:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- 3-term Taylor expansion

$$y(t_{j+1}) = y(t_j) + hy'(t_j) + \frac{h^2}{2}y''(t_j) + \frac{h^3}{3!}y^{(3)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) = f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)) \quad (3)$$

2-nd order Taylor method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

where

$$\mathbf{T}^{(2)}(t_j, w_j) = f(t_j, w_j) + \frac{h}{2}f'(t_j, w_j) \xleftarrow{\textit{nasty!}}$$

§5.3 2nd Order Taylor Method:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- 3-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2}y''(t_j) + \frac{h^3}{3!}y^{(3)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)) \quad (3)$$

2-nd order Taylor method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

where

$$\mathbf{T}^{(2)}(t_j, w_j) = f(t_j, w_j) + \frac{h}{2}f'(t_j, w_j) \xLeftarrow{nasty!}$$

- $f'(t, y(t))$ is total derivative

$$f'(t, y(t)) \quad = \quad \frac{d}{dt}f(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \; y'(t)$$

§5.3 2nd Order Taylor Method:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- 3-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2} y''(t_j) + \frac{h^3}{3!} y^{(3)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)) \quad (3)$$

2-nd order Taylor method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

where

$$\mathbf{T}^{(2)}(t_j, w_j) = f(t_j, w_j) + \frac{h}{2} f'(t_j, w_j) \xLeftarrow{\textit{nasty!}}$$

- $f'(t, y(t))$ is total derivative

$$f'(t, y(t)) \quad = \quad \frac{d}{dt} f(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \; y'(t) = \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \; f(t, y(t))$$

§5.3 2nd Order Taylor Method:

$$\frac{dy}{dt} = f(t,y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- 3-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2}y''(t_j) + \frac{h^3}{3!}y^{(3)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)) \quad (3)$$

2-nd order Taylor method
$w_0 = \alpha,$ $w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$

where

$$\mathbf{T}^{(2)}(t_j, w_j) = f(t_j, w_j) + \frac{h}{2}f'(t_j, w_j) \quad \xLeftarrow{nasty!}$$

- $f'(t, y(t))$ is total derivative

$$f'(t,y(t)) \quad = \quad \frac{d}{dt}f(t,y(t)) = \frac{\partial f}{\partial t}(t,y(t)) + \frac{\partial f}{\partial y}(t,y(t))\ y'(t) == \frac{\partial f}{\partial t}(t,y(t)) + \frac{\partial f}{\partial y}(t,y(t))\ f(t,y(t))$$

$$f'(t,w) \quad == \quad \frac{\partial f}{\partial t}(t,w) + \frac{\partial f}{\partial y}(t,w)\ f(t,w), \quad \Longleftarrow \quad \text{two partial derivatives}$$

§5.3 2nd Order Taylor Method:

$$\frac{dy}{dt} = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha. \quad (1)$$

- 3-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2} y''(t_j) + \frac{h^3}{3!} y^{(3)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)) \quad (3)$$

2-nd order Taylor method	where
$w_0 = \alpha,$ $w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$	$\mathbf{T}^{(2)}(t_j, w_j) = f(t_j, w_j) + \frac{h}{2} f'(t_j, w_j) \quad \stackrel{nasty!}{\Longleftarrow}$

- $f'(t, y(t))$ is total derivative

$$f'(t, y(t)) \quad = \quad \frac{d}{dt} f(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \; y'(t) == \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \; f(t, y(t))$$

$$f'(t, w) \quad == \quad \frac{\partial f}{\partial t}(t, w) + \frac{\partial f}{\partial y}(t, w) \; f(t, w), \quad \Longleftarrow \text{ two partial derivatives}$$

- But **LTE** is 2-nd order

$$\tau_{j+1}(h) \stackrel{def}{=} \frac{y(t_{j+1}) - y(t_j)}{h} - \mathbf{T}^{(2)}(t_j, y(t_j))$$

§5.3 2nd Order Taylor Method:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- 3-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + h y'(t_j) + \frac{h^2}{2} y''(t_j) + \frac{h^3}{3!} y^{(3)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)) \quad (3)$$

2-nd order Taylor method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

where

$$\mathbf{T}^{(2)}(t_j, w_j) = f(t_j, w_j) + \frac{h}{2} f'(t_j, w_j) \quad \overset{nasty!}{\Longleftarrow}$$

- $f'(t, y(t))$ is total derivative

$$f'(t, y(t)) \quad = \quad \frac{d}{dt} f(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \; y'(t) == \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t)) \; f(t, y(t))$$

$$f'(t, w) \quad == \quad \frac{\partial f}{\partial t}(t, w) + \frac{\partial f}{\partial y}(t, w) \; f(t, w), \quad \Longleftarrow \quad \text{two partial derivatives}$$

- But **LTE** is 2-nd order

$$\tau_{j+1}(h) \overset{def}{=} \frac{y(t_{j+1}) - y(t_j)}{h} - \mathbf{T}^{(2)}(t_j, y(t_j)) \overset{(3)}{=} \frac{y(t_{j+1}) - y(t_j)}{h} - \left(y'(t_j) + \frac{h}{2} y''(t_j) \right)$$

§5.3 2nd Order Taylor Method:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- 3-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2} y''(t_j) + \frac{h^3}{3!} y^{(3)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)) \quad (3)$$

2-nd order Taylor method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

where

$$\mathbf{T}^{(2)}(t_j, w_j) = f(t_j, w_j) + \frac{h}{2} f'(t_j, w_j) \quad \xLeftarrow{nasty!}$$

- $f'(t, y(t))$ is total derivative

$$f'(t, y(t)) \quad = \quad \frac{d}{dt} f(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t))\; y'(t) == \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t))\; f(t, y(t))$$

$$f'(t, w) \quad == \quad \frac{\partial f}{\partial t}(t, w) + \frac{\partial f}{\partial y}(t, w)\; f(t, w), \quad \Longleftarrow \quad \text{two partial derivatives}$$

- But **LTE** is 2-nd order

$$\tau_{j+1}(h) \stackrel{def}{=} \frac{y(t_{j+1}) - y(t_j)}{h} - \mathbf{T}^{(2)}(t_j, y(t_j)) \stackrel{(3)}{=} \frac{y(t_{j+1}) - y(t_j)}{h} - \left(y'(t_j) + \frac{h}{2} y''(t_j)\right) \stackrel{(2)}{==} \frac{h^{\boxed{2}}}{3!} y^{(3)}(\xi_j)$$

Example (when derivatives are easy)

$\frac{dy}{dt} = f(t,y),\ f(t,y) = y - t^2 + 1,\ 0 \leq t \leq 2,\ y(0) = 0.5$

Example (when derivatives are easy)

$\frac{dy}{dt} = f(t, y),\ f(t, y) = y - t^2 + 1,\ 0 \leq t \leq 2,\ y(0) = 0.5$

2-nd order Taylor method

$$w_0 = 0.5,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

where

$$\mathbf{T}^{(2)}(t_j, w_j) = \frac{\partial f}{\partial t}(t, w) + \frac{\partial f}{\partial y}(t, w)\ f(t, w)$$

Example (when derivatives are easy)

$\frac{dy}{dt} = f(t, y),\ f(t, y) = y - t^2 + 1,\ 0 \le t \le 2,\ y(0) = 0.5$

2-nd order Taylor method

$$w_0 = 0.5,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

where

$$\mathbf{T}^{(2)}(t_j, w_j) = \frac{\partial f}{\partial t}(t, w) + \frac{\partial f}{\partial y}(t, w)\ f(t, w)$$

Since $\quad f(t, y(t)) = y - t^2 + 1, \quad \frac{\partial f}{\partial t}(t, y(t)) = -2t, \quad \frac{\partial f}{\partial y}(t, y(t)) = 1.$

Thus, for $N = 10,\ h = 2/N = 0.2$

$$w_{j+1} \quad = \quad w_j + h\left(w_j - t_j^2 + 1 + \frac{h}{2}\left(-2t_j + w_j - t_j^2 + 1\right)\right)$$

Example (when derivatives are easy)

$$\frac{dy}{dt} = f(t,y),\; f(t,y) = y - t^2 + 1,\; 0 \le t \le 2,\; y(0) = 0.5$$

2-nd order Taylor method

$$w_0 = 0.5,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

where

$$\mathbf{T}^{(2)}(t_j, w_j) = \frac{\partial f}{\partial t}(t, w) + \frac{\partial f}{\partial y}(t, w)\; f(t, w)$$

Since $\quad f(t, y(t)) = y - t^2 + 1, \quad \frac{\partial f}{\partial t}(t, y(t)) = -2t, \quad \frac{\partial f}{\partial y}(t, y(t)) = 1.$

Thus, for $N = 10,\; h = 2/N = 0.2$

$$\begin{aligned} w_{j+1} &= w_j + h\left(w_j - t_j^2 + 1 + \frac{h}{2}\left(-2t_j + w_j - t_j^2 + 1\right)\right) \\ &= \left(1 + h + \frac{h^2}{2}\right) w_j + \left(h + \frac{h^2}{2}\right)\left(1 - t_j^2\right) - h^2 t_j, \end{aligned}$$

t_i	Euler's Method	Taylor Method	Exact Solution
0.00000	0.50000	0.50000	0.50000
0.20000	0.80000	0.83000	0.82930
0.40000	1.15200	1.21580	1.21409
0.60000	1.55040	1.65208	1.64894
0.80000	1.98848	2.13233	2.12723
1.00000	2.45818	2.64865	2.64086
1.20000	2.94981	3.19135	3.17994
1.40000	3.45177	3.74864	3.73240
1.60000	3.95013	4.30615	4.28348
1.80000	4.42815	4.84630	4.81518
2.00000	4.86578	5.34768	5.30547

Example (when derivatives are easy)

$\frac{dy}{dt} = f(t,y),\ f(t,y) = y - t^2 + 1,\ 0 \le t \le 2,\ y(0) = 0.5$

2-nd order Taylor method

$$w_0 = 0.5,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

where

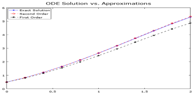
$$\mathbf{T}^{(2)}(t_j, w_j) = \frac{\partial f}{\partial t}(t,w) + \frac{\partial f}{\partial y}(t,w)\ f(t,w)$$

Since $\quad f(t, y(t)) = y - t^2 + 1, \quad \frac{\partial f}{\partial t}(t, y(t)) = -2t, \quad \frac{\partial f}{\partial y}(t, y(t)) = 1.$

Thus, for $N = 10,\ h = 2/N = 0.2$

$$\begin{aligned} w_{j+1} &= w_j + h\left(w_j - t_j^2 + 1 + \frac{h}{2}\left(-2t_j + w_j - t_j^2 + 1\right)\right) \\ &= \left(1 + h + \frac{h^2}{2}\right) w_j + \left(h + \frac{h^2}{2}\right)\left(1 - t_j^2\right) - h^2 t_j, \end{aligned}$$

t_i	Euler's Method	Taylor Method	Exact Solution
0.00000	0.50000	0.50000	0.50000
0.20000	0.80000	0.83000	0.82930
0.40000	1.15200	1.21580	1.21409
0.60000	1.55040	1.65208	1.64894
0.80000	1.98848	2.13233	2.12723
1.00000	2.45818	2.64865	2.64086
1.20000	2.94981	3.19135	3.17994
1.40000	3.45177	3.74864	3.73240
1.60000	3.95013	4.30615	4.28348
1.80000	4.42815	4.84630	4.81518
2.00000	4.86578	5.34768	5.30547



Example (when derivatives are easy)

$$\frac{dy}{dt} = f(t,y),\ f(t,y) = y - t^2 + 1,\ 0 \le t \le 2,\ y(0) = 0.5$$

2-nd order Taylor method

$$w_0 = 0.5,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

where

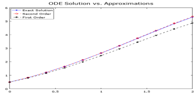
$$\mathbf{T}^{(2)}(t_j, w_j) = \frac{\partial f}{\partial t}(t, w) + \frac{\partial f}{\partial y}(t, w)\ f(t, w)$$

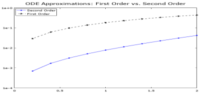
Since $\quad f(t, y(t)) = y - t^2 + 1, \quad \dfrac{\partial f}{\partial t}(t, y(t)) = -2t, \quad \dfrac{\partial f}{\partial y}(t, y(t)) = 1.$

Thus, for $N = 10,\ h = 2/N = 0.2$

$$\begin{aligned} w_{j+1} &= w_j + h\left(w_j - t_j^2 + 1 + \frac{h}{2}\left(-2t_j + w_j - t_j^2 + 1\right)\right) \\ &= \left(1 + h + \frac{h^2}{2}\right) w_j + \left(h + \frac{h^2}{2}\right)\left(1 - t_j^2\right) - h^2 t_j, \end{aligned}$$

t_i	Euler's Method	Taylor Method	Exact Solution
0.00000	0.50000	0.50000	0.50000
0.20000	0.80000	0.83000	0.82930
0.40000	1.15200	1.21580	1.21409
0.60000	1.55040	1.65208	1.64894
0.80000	1.98848	2.13233	2.12723
1.00000	2.45818	2.64865	2.64086
1.20000	2.94981	3.19135	3.17994
1.40000	3.45177	3.74864	3.73240
1.60000	3.95013	4.30615	4.28348
1.80000	4.42815	4.84630	4.81518
2.00000	4.86578	5.34768	5.30547





n-th Order Taylor Methods:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

n-th Order Taylor Methods:

$$\frac{dy}{dt} = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha. \quad (1)$$

- $(n+1)$-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2} y''(t_j) + \cdots + \frac{h^n}{n!} y^{(n)}(t_j) + \frac{h^{n+1}}{(n+1)!} y^{(n+1)}(\xi_j) \quad (2)$$

n-th Order Taylor Methods:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- $(n+1)$-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2} y''(t_j) + \cdots + \frac{h^n}{n!} y^{(n)}(t_j) + \frac{h^{n+1}}{(n+1)!} y^{(n+1)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)), \quad y^{(n)}(t_j) = f^{(n-1)}(t_j, y(t_j)) \quad (3)$$

n-th Order Taylor Methods:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- $(n+1)$-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2} y''(t_j) + \cdots + \frac{h^n}{n!} y^{(n)}(t_j) + \frac{h^{n+1}}{(n+1)!} y^{(n+1)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)), \quad y^{(n)}(t_j) = f^{(n-1)}(t_j, y(t_j)) \quad (3)$$

n-th order Taylor method

$$w_0 = \alpha,$$

$$w_{j+1} = w_j + h\mathbf{T}^{(n)}(t_j, w_j), \quad j = 0, 1, \cdots,$$

where $\mathbf{T}^{(n)}(t_j, w_j) = f(t_j, w_j) + \frac{h}{2} f'(t_j, w_j) + \cdots + \frac{h^{n-1}}{n!} f^{(n-1)}(t_j, w_j)$

n-th Order Taylor Methods:

$$\frac{dy}{dt} = f(t,y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- $(n+1)$-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2} y''(t_j) + \cdots + \frac{h^n}{n!} y^{(n)}(t_j) + \frac{h^{n+1}}{(n+1)!} y^{(n+1)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)), \quad y^{(n)}(t_j) = f^{(n-1)}(t_j, y(t_j)) \quad (3)$$

n-th order Taylor method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(n)}(t_j, w_j), \quad j = 0, 1, \cdots,$$

where $\mathbf{T}^{(n)}(t_j, w_j) = f(t_j, w_j) + \frac{h}{2} f'(t_j, w_j) + \cdots + \frac{h^{n-1}}{n!} f^{(n-1)}(t_j, w_j)$

- $f'(t, y(t)), \cdots, f^{(n-1)}(t, y(t))$ are total derivatives $\Longleftarrow$ even more nasty

n-th Order Taylor Methods:

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \quad (1)$$

- $(n+1)$-term Taylor expansion

$$y(t_{j+1}) \quad = \quad y(t_j) + hy'(t_j) + \frac{h^2}{2}y''(t_j) + \cdots + \frac{h^n}{n!}y^{(n)}(t_j) + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\xi_j) \quad (2)$$

- On the other hand, (1) implies in (2)

$$y'(t_j) \quad = \quad f(t_j, y(t_j)), \quad y''(t_j) = f'(t_j, y(t_j)), \quad y^{(n)}(t_j) = f^{(n-1)}(t_j, y(t_j)) \quad (3)$$

n-th order Taylor method

$$w_0 = \alpha,$$

$$w_{j+1} = w_j + h\mathbf{T}^{(n)}(t_j, w_j), \quad j = 0, 1, \cdots,$$

where $\mathbf{T}^{(n)}(t_j, w_j) = f(t_j, w_j) + \frac{h}{2}f'(t_j, w_j) + \cdots + \frac{h^{n-1}}{n!}f^{(n-1)}(t_j, w_j)$

- $f'(t, y(t)), \cdots, f^{(n-1)}(t, y(t))$ are total derivatives $\Longleftarrow$ even more nasty
- But **LTE** is n-th order

$$\tau_{j+1}(h) \overset{def}{=} \frac{y(t_{j+1}) - y(t_j)}{h} - \mathbf{T}^{(n)}(t_j, y(t_j)) \overset{(3)}{=} \frac{y(t_{j+1}) - y(t_j)}{h} - \left(y'(t_j) + \frac{h}{2}y''(t_j) + \cdots + \frac{h^{n-1}}{n!}y^{(n)}(t_j)\right)$$

$$\overset{(2)}{=} \frac{h^{\boxed{n}}}{(n+1)!}y^{(n+1)}(\xi_j)$$

$\underline{\text{Runge-Kutta methods}} = \underline{\text{Taylor methods}} + \underline{\text{Taylor expansion in two variables}}$

- **The good**
  - Retain order of Taylor methods
  - No need for derivative calculations
- **The bad**
  - Details ugly for any order ≥ 3

First order Taylor expansion in two variables

Suppose that $f(t, y)$ and all its partial derivatives are continuous on

$$D \stackrel{def}{=} \{(t, y) \mid \quad a \leq t \leq b, \quad c \leq y \leq d.\}$$

Theorem: Let $(t_0, y_0), (t, y) \in D, \Delta_t = t - t_0, \Delta_y = y - y_0$. **Then**

$$f(t, y) = P_1(t, y) + R_1(t, y), \quad \text{where for some } (\xi, \mu) \in D$$

First order Taylor expansion in two variables

Suppose that $f(t,y)$ and all its partial derivatives are continuous on

$$D \stackrel{def}{=} \{(t,y) \,|\quad a \le t \le b, \quad c \le y \le d.\}$$

Theorem: Let $(t_0,y_0),(t,y) \in D, \Delta_t = t - t_0, \Delta_y = y - y_0$. **Then**

$$f(t,y) = P_1(t,y) + R_1(t,y), \quad \text{where for some } (\xi,\mu) \in D$$

$$\begin{aligned} P_1(t,y) &= f(t_0,y_0) + \Delta_t \frac{\partial f}{\partial t}(t_0,y_0) + \Delta_y \frac{\partial f}{\partial y}(t_0,y_0) \\ R_1(t,y) &= \frac{\Delta_t^2}{2}\frac{\partial^2 f}{\partial t^2}(\xi,\mu) + \Delta_t\,\Delta_y \frac{\partial^2 f}{\partial t \partial y}(\xi,\mu) + \frac{\Delta_y^2}{2}\frac{\partial^2 f}{\partial y^2}(\xi,\mu) \end{aligned}$$

.

First order Taylor expansion in two variables

Suppose that $f(t,y)$ and all its partial derivatives are continuous on

$$D \stackrel{def}{=} \{(t,y) \,|\quad a \le t \le b, \quad c \le y \le d.\}$$

Theorem: Let $(t_0, y_0), (t,y) \in D, \Delta_t = t - t_0, \Delta_y = y - y_0$. **Then**

$$f(t,y) = P_1(t,y) + R_1(t,y), \quad \text{where for some } (\xi, \mu) \in D$$

$$\begin{aligned} P_1(t,y) &= f(t_0,y_0) + \Delta_t \frac{\partial f}{\partial t}(t_0,y_0) + \Delta_y \frac{\partial f}{\partial y}(t_0,y_0) \\ R_1(t,y) &= \frac{\Delta_t^2}{2}\frac{\partial^2 f}{\partial t^2}(\xi,\mu) + \Delta_t\,\Delta_y \frac{\partial^2 f}{\partial t \partial y}(\xi,\mu) + \frac{\Delta_y^2}{2}\frac{\partial^2 f}{\partial y^2}(\xi,\mu) \end{aligned}$$

.

Reverse Theorem: Let $P_1(t,y) = f(t_0,y_0) + \Delta_t \frac{\partial f}{\partial t}(t_0,y_0) + \Delta_y \frac{\partial f}{\partial y}(t_0,y_0)$ for tiny Δ_t, Δ_y and for $t = t_0 + \Delta_t, y = y_0 + \Delta_y$. **Then**

$$P_1(t,y) = f(t,y) - R_1(t,y)$$

First order Taylor expansion in two variables

Suppose that $f(t, y)$ and all its partial derivatives are continuous on

$$D \stackrel{def}{=} \{(t, y) \mid \quad a \le t \le b, \quad c \le y \le d.\}$$

Theorem: Let $(t_0, y_0), (t, y) \in D, \Delta_t = t - t_0, \Delta_y = y - y_0$. **Then**

$$f(t, y) = P_1(t, y) + R_1(t, y), \quad \text{where for some } (\xi, \mu) \in D$$

$$\begin{aligned} P_1(t, y) &= f(t_0, y_0) + \Delta_t \frac{\partial f}{\partial t}(t_0, y_0) + \Delta_y \frac{\partial f}{\partial y}(t_0, y_0) \\ R_1(t, y) &= \frac{\Delta_t^2}{2} \frac{\partial^2 f}{\partial t^2}(\xi, \mu) + \Delta_t \, \Delta_y \frac{\partial^2 f}{\partial t \partial y}(\xi, \mu) + \frac{\Delta_y^2}{2} \frac{\partial^2 f}{\partial y^2}(\xi, \mu) \end{aligned}$$

.

Reverse Theorem: Let $P_1(t, y) = f(t_0, y_0) + \Delta_t \frac{\partial f}{\partial t}(t_0, y_0) + \Delta_y \frac{\partial f}{\partial y}(t_0, y_0)$ for tiny Δ_t, Δ_y and for $t = t_0 + \Delta_t, y = y_0 + \Delta_y$. **Then**

$$P_1(t, y) = f(t, y) - R_1(t, y) \approx f(t, y), \quad \text{where}$$

$$R_1(t, y) = \frac{\Delta_t^2}{2} \frac{\partial^2 f}{\partial t^2}(\xi, \mu) + \Delta_t \, \Delta_y \frac{\partial^2 f}{\partial t \partial y}(\xi, \mu) + \frac{\Delta_y^2}{2} \frac{\partial^2 f}{\partial y^2}(\xi, \mu) \quad \boxed{\text{tinier}}$$

2nd Order Runge-Kutta Method

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

2nd Order Runge-Kutta Method

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

2-nd order Taylor method

$$w_0 = \alpha,$$

$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

$$\begin{aligned}
\text{where } \mathbf{T}^{(2)}(t, y) &= f(t, y) + \frac{h}{2} f'(t, y) = f(t, y) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t, y) + \frac{\partial f}{\partial y}(t, y)\,\frac{dy}{dt}\right) \\
&= f(t, y) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t, y) + \frac{\partial f}{\partial y}(t, y)\, f(t, y)\right)
\end{aligned}$$

2nd Order Runge-Kutta Method

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

2-nd order Taylor method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

$$\begin{aligned}
\text{where } \mathbf{T}^{(2)}(t,y) &= f(t,y) + \frac{h}{2} f'(t,y) = f(t,y) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t,y) + \frac{\partial f}{\partial y}(t,y)\,\frac{dy}{dt}\right) \\
&= f(t,y) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t,y) + \frac{\partial f}{\partial y}(t,y)\, f(t,y)\right) \quad \left(\Delta_t = \frac{h}{2},\ \Delta_y = \frac{h}{2}\, f(t,y)\right) \\
&= f\left(t + \frac{h}{2}, y + \frac{h}{2} f(t,y)\right) - R_1\left(t + \frac{h}{2}, y + \frac{h}{2} f(t,y)\right)
\end{aligned}$$

- From first order Taylor expansion,

$$R_1\left(t + \frac{h}{2}, y + \frac{h}{2} f(t,y)\right) = \frac{h^2}{8}\,\frac{\partial^2 f}{\partial t^2}(\xi,\mu) + \frac{h^2 f(t,y)}{4}\,\frac{\partial^2 f}{\partial t \partial y}(\xi,\mu) + \frac{h^2 f^2(t,y)}{8}\,\frac{\partial^2 f}{\partial y^2}(\xi,\mu) = O(h^2)$$

- Method remains <u>second order</u> after dropping $R_1\left(t + \frac{h}{2}, y + \frac{h}{2} f(t,y)\right)$
- <u>Only two</u> function evaluations to approximate $\mathbf{T}^{(2)}(t_j, w_j)$

2nd Order Runge-Kutta Method

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

2-nd order Taylor method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

2-nd order Runge-Kutta method (Midpoint Method)

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + hf\left(t_j + \frac{h}{2}, w_j + \frac{h}{2} f(t_j, w_j)\right), \;\; j = 0, 1, \cdots$$

- Runge-Kutta method <u>second order, two function evaluations</u> per step
- Hooray!!! No need for derivative calculations

2nd Order Runge-Kutta Method

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

2-nd order Taylor method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\mathbf{T}^{(2)}(t_j, w_j), \quad j = 0, 1, \cdots, N-1,$$

2-nd order Runge-Kutta method (Midpoint Method)

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + hf\left(t_j + \frac{h}{2}, w_j + \frac{h}{2} f(t_j, w_j)\right), \quad j = 0, 1, \cdots$$

- Runge-Kutta method second order, two function evaluations per step
- Hooray!!! No need for derivative calculations
- **next:** general second order methods, higher order methods

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\, \left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \quad j = 0, 1, \cdots$$

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$\begin{aligned} w_0 &= \alpha, \\ w_{j+1} &= w_j + h\, \left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \;\; j = 0, 1, \cdots \end{aligned}$$

- Two function evaluations for each j,
- Want to choose $a_1, a_2, \alpha_2, \delta_2$ for highest possible order of accuracy.

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\,\left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \;\; j = 0, 1, \cdots$$

- Two function evaluations for each j,
- Want to choose $a_1, a_2, \alpha_2, \delta_2$ for highest possible order of accuracy.

local truncation error $\tau_{j+1}(h) = \dfrac{y(t_{j+1}) - y(t_j)}{h} - \left(a_1 f(t_j, y(t_j)) + a_2 f\left(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j))\right)\right)$

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\,\left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \quad j = 0, 1, \cdots$$

- Two function evaluations for each j,
- Want to choose $a_1, a_2, \alpha_2, \delta_2$ for highest possible order of accuracy.

local truncation error $\tau_{j+1}(h) = \dfrac{y(t_{j+1}) - y(t_j)}{h} - \left(a_1 f(t_j, y(t_j)) + a_2 f\left(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j))\right)\right)$

$$\begin{aligned}
\text{with } \frac{y(t_{j+1}) - y(t_j)}{h} &= y'(t_j) + \frac{h}{2} y''(t_j) + O(h^2) \\
&= f(t_j, y(t_j)) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t_j, y(t_j)) + f(t_j, y(t_j))\frac{\partial f}{\partial y}(t_j, y(t_j))\right) + O(h^2)
\end{aligned}$$

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\,\left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \quad j = 0, 1, \cdots$$

- Two function evaluations for each j,
- Want to choose $a_1, a_2, \alpha_2, \delta_2$ for highest possible order of accuracy.

local truncation error $\tau_{j+1}(h) = \dfrac{y(t_{j+1}) - y(t_j)}{h} - \left(a_1 f(t_j, y(t_j)) + a_2 f\left(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j))\right)\right)$

$$\begin{aligned}
\text{with } \frac{y(t_{j+1}) - y(t_j)}{h} &= y'(t_j) + \frac{h}{2} y''(t_j) + O(h^2) \\
&= f(t_j, y(t_j)) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t_j, y(t_j)) + f(t_j, y(t_j))\frac{\partial f}{\partial y}(t_j, y(t_j))\right) + O(h^2)
\end{aligned}$$

$$\begin{aligned}
&a_1 f(t_j, y(t_j)) + a_2 f\left(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j))\right) \\
&\quad = (a_1 + a_2)\, f(t_j, y(t_j)) + a_2\, \alpha_2\, \frac{\partial f}{\partial t}(t_j, y(t_j)) + a_2\, \delta_2 f(t_j, y(t_j))\frac{\partial f}{\partial y}(t_j, y(t_j)) + O(h^2)
\end{aligned}$$

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$\begin{aligned} w_0 &= \alpha, \\ w_{j+1} &= w_j + h \left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \quad j = 0, 1, \cdots \end{aligned}$$

- Two function evaluations for each j,
- Want to choose $a_1, a_2, \alpha_2, \delta_2$ for highest possible order of accuracy.

$$\textbf{local truncation error } \tau_{j+1}(h) = \frac{y(t_{j+1}) - y(t_j)}{h} - \left(a_1 f(t_j, y(t_j)) + a_2 f\left(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j))\right)\right)$$

$$\begin{aligned} \text{with } \frac{y(t_{j+1}) - y(t_j)}{h} &= y'(t_j) + \frac{h}{2} y''(t_j) + O(h^2) \\ &= f(t_j, y(t_j)) + \frac{h}{2}\left(\frac{\partial f}{\partial t}(t_j, y(t_j)) + f(t_j, y(t_j))\frac{\partial f}{\partial y}(t_j, y(t_j))\right) + O(h^2) \end{aligned}$$

$$\begin{aligned} & a_1 f(t_j, y(t_j)) + a_2 f\left(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j))\right) \\ &= (a_1 + a_2) f(t_j, y(t_j)) + a_2\, \alpha_2 \frac{\partial f}{\partial t}(t_j, y(t_j)) + a_2\, \delta_2 f(t_j, y(t_j))\frac{\partial f}{\partial y}(t_j, y(t_j)) + O(h^2) \end{aligned}$$

$$\Longrightarrow \tau_{j+1}(h) = \underbrace{(1 - (a_1 + a_2))}_{0} f(t_j, y(t_j)) + \underbrace{\left(\frac{h}{2} - a_2\, \alpha_2\right)}_{0} \frac{\partial f}{\partial t}(t_j, y(t_j)) + \underbrace{\left(\frac{h}{2} f(t_j, y(t_j)) - a_2\, \delta_2\right)}_{0} \frac{\partial f}{\partial y}(t_j, y(t_j)) + O(h^2)$$

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$\begin{aligned} w_0 &= \alpha, \\ w_{j+1} &= w_j + h \left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \quad j = 0, 1, \cdots \end{aligned}$$

- Two function evaluations for each j,
- Want to choose $a_1, a_2, \alpha_2, \delta_2$ for highest possible order of accuracy.

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$\begin{aligned} w_0 &= \alpha, \\ w_{j+1} &= w_j + h\,\left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \quad j = 0, 1, \cdots \end{aligned}$$

- Two function evaluations for each j,
- Want to choose $a_1, a_2, \alpha_2, \delta_2$ for highest possible order of accuracy.

local truncation error $\tau_{j+1}(h) = \dfrac{y(t_{j+1}) - y(t_j)}{h} - \left(a_1 f(t_j, y(t_j)) + a_2 f\left(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j))\right)\right) = O(h^2)$

with $1 - (a_1 + a_2) = 0, \ \dfrac{h}{2} - a_2\,\alpha_2 = 0, \ \dfrac{h}{2} f(t_j, y(t_j)) - a_2\,\delta_2 = 0.$

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$\begin{aligned} w_0 &= \alpha, \\ w_{j+1} &= w_j + h\, \left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \;\; j = 0, 1, \cdots \end{aligned}$$

- Two function evaluations for each j,
- Want to choose $a_1, a_2, \alpha_2, \delta_2$ for highest possible order of accuracy.

local truncation error $\tau_{j+1}(h) = \dfrac{y(t_{j+1}) - y(t_j)}{h} - \left(a_1 f(t_j, y(t_j)) + a_2 f\left(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j))\right)\right) = O(h^2)$

with $1 - (a_1 + a_2) = 0,\;\; \dfrac{h}{2} - a_2\,\alpha_2 = 0,\;\; \dfrac{h}{2} f(t_j, y(t_j)) - a_2\,\delta_2 = 0.$

- any valid parameter choice in $a_1, a_2, \alpha_2, \delta_2$ leads to 2nd order method
- four parameters, three equations, none leads to higher order.

Midpoint method $w_{j+1} \;=\; w_j + h\, f\left(t_j + \dfrac{h}{2}, w_j + \dfrac{h}{2} f(t_j, w_j)\right)$

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\,\left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \;\; j = 0, 1, \cdots$$

- Two function evaluations for each j,
- Want to choose $a_1, a_2, \alpha_2, \delta_2$ for highest possible order of accuracy.

local truncation error $\tau_{j+1}(h) = \dfrac{y(t_{j+1}) - y(t_j)}{h} - \left(a_1 f(t_j, y(t_j)) + a_2 f\left(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j))\right)\right) = O(h^2)$

with $1 - (a_1 + a_2) = 0, \;\; \dfrac{h}{2} - a_2\,\alpha_2 = 0, \;\; \dfrac{h}{2} f(t_j, y(t_j)) - a_2\,\delta_2 = 0.$

- any valid parameter choice in $a_1, a_2, \alpha_2, \delta_2$ leads to 2nd order method
- four parameters, three equations, none leads to higher order.

Midpoint method $w_{j+1} \;\; = \;\; w_j + h\,f\left(t_j + \dfrac{h}{2}, w_j + \dfrac{h}{2} f(t_j, w_j)\right) \quad \left(a_1 = 0, a_2 = 1, \alpha_2 = \delta_2 = \dfrac{h}{2}\right)$

Modified Euler method $w_{j+1} \;\; = \;\; w_j + \dfrac{h}{2}\left(f(t_j, w_j) + f\left(t_{j+1}, w_j + hf(t_j, w_j)\right)\right)$

General 2nd Order Runge-Kutta Methods

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

General Runge-Kutta method

$$w_0 = \alpha,$$
$$w_{j+1} = w_j + h\,\left(a_1 f(t_j, w_j) + a_2 f\left(t_j + \alpha_2, w_j + \delta_2 f(t_j, w_j)\right)\right), \quad j = 0, 1, \cdots$$

- Two function evaluations for each j,
- Want to choose $a_1, a_2, \alpha_2, \delta_2$ for highest possible order of accuracy.

local truncation error $\tau_{j+1}(h) = \dfrac{y(t_{j+1}) - y(t_j)}{h} - \left(a_1 f(t_j, y(t_j)) + a_2 f\left(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j))\right)\right) = O(h^2)$

with $1 - (a_1 + a_2) = 0, \ \dfrac{h}{2} - a_2\,\alpha_2 = 0, \ \dfrac{h}{2} f(t_j, y(t_j)) - a_2\,\delta_2 = 0.$

- any valid parameter choice in $a_1, a_2, \alpha_2, \delta_2$ leads to 2nd order method
- four parameters, three equations, none leads to higher order.

Midpoint method $w_{j+1} = w_j + h\,f\left(t_j + \dfrac{h}{2}, w_j + \dfrac{h}{2} f(t_j, w_j)\right) \quad \left(a_1 = 0, a_2 = 1, \alpha_2 = \delta_2 = \dfrac{h}{2}\right)$

Modified Euler method $w_{j+1} = w_j + \dfrac{h}{2}\left(f(t_j, w_j) + f\left(t_{j+1}, w_j + hf(t_j, w_j)\right)\right) \quad \left(a_1 = a_2 = \dfrac{1}{2}, \alpha_2 = \delta_2 = h\right)$

Higher Order Runge-Kutta Methods

3rd order Runge-Kutta Method

$w_0 = \alpha;$
for $\quad j = 0, 1, \cdots, N-1,$

$$w_{j+1} = w_j + \frac{h}{4}\left(f(t_j, w_j) + 3f\left(t_j + \frac{2h}{3}, w_j + \frac{2h}{3} f(t_j + \frac{h}{3}, w_j + \frac{h}{3} f(t_j, w_j))\right)\right)$$
$$\stackrel{def}{=} w_j + h\,\phi(t_j, w_j).$$

$\boxed{\text{3 function evaluations per step}}$

Higher Order Runge-Kutta Methods

3rd order Runge-Kutta Method

$w_0 = \alpha;$
for $\quad j = 0, 1, \cdots, N-1,$

$$
\begin{aligned}
w_{j+1} &= w_j + \frac{h}{4}\Big(f(t_j, w_j) + 3f\Big(t_j + \frac{2h}{3}, w_j + \frac{2h}{3} f(t_j + \frac{h}{3}, w_j + \frac{h}{3} f(t_j, w_j))\Big)\Big) \\
&\stackrel{def}{=} w_j + h\, \phi(t_j, w_j).
\end{aligned}
$$

$\boxed{\text{3 function evaluations per step}}$

4th order Runge-Kutta Method

$w_0 = \alpha;$
for $\quad j = 0, 1, \cdots, N-1,$

$$
\begin{aligned}
k_1 &= h\, f(t_j, w_j), \\
k_2 &= h\, f\left(t_j + \frac{h}{2}, w_j + \frac{1}{2} k_1\right), \\
k_3 &= h\, f\left(t_j + \frac{h}{2}, w_j + \frac{1}{2} k_2\right), \\
k_4 &= h\, f\left(t_{j+1}, w_j + k_3\right), \\
w_{j+1} &= w_j + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right).
\end{aligned}
$$

$\boxed{\text{4 function evaluations per step}}$

Example

$$\text{Initial Value ODE} \qquad \frac{dy}{dt} = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5,$$

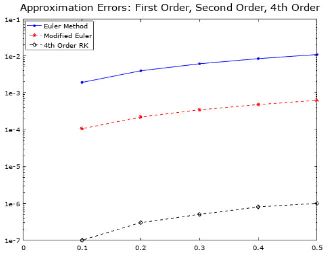
$$\text{exact solution} \qquad y(t) = (1+t)^2 - 0.5\, e^t.$$

Example

Initial Value ODE $\quad \dfrac{dy}{dt} = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5,$

exact solution $\quad y(t) = (1+t)^2 - 0.5\, e^t.$

t_i	Exact	Euler $h = 0.025$	Modified Euler $h = 0.05$	Runge-Kutta Order Four $h = 0.1$
0.0	0.5000000	0.5000000	0.5000000	0.5000000
0.1	0.6574145	0.6554982	0.6573085	0.6574144
0.2	0.8292986	0.8253385	0.8290778	0.8292983
0.3	1.0150706	1.0089334	1.0147254	1.0150701
0.4	1.2140877	1.2056345	1.2136079	1.2140869
0.5	1.4256394	1.4147264	1.4250141	1.4256384



$\boxed{\text{Crying baby Principle}}$ in adaptive algorithms

MORE POINTS IN REGIONS OF INADEQUATE ACCURACY

Crying baby Principle in adaptive algorithms

MORE POINTS IN REGIONS OF INADEQUATE ACCURACY

crying babies get more candies

Crying baby Principle in adaptive algorithms

MORE POINTS IN REGIONS OF INADEQUATE ACCURACY

crying babies get more candies

Numerical accuracy adequate unless detected otherwise

Crying baby Principle in adaptive algorithms

MORE POINTS IN REGIONS OF INADEQUATE ACCURACY

crying babies get more candies

Numerical accuracy adequate unless detected otherwise

Lots of numerical inaccuracies can go undetected.

Adaptive Error Control (I)

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

Consider a variable-step method with a well-chosen function $\phi(t, w, h)$:

- $w_0 = \alpha$.
- for $j = 0, 1, \cdots,$
  - **choose** step-size $h_j = t_{j+1} - t_j$,
  - **set** $w_{j+1} = w_j + h_j \, \phi(t_j, w_j, h_j)$.

Adaptive Error Control (I)

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

Consider a variable-step method with a well-chosen function $\phi(t, w, h)$:

- $w_0 = \alpha$.
- for $j = 0, 1, \cdots,$
  - **choose** step-size $h_j = t_{j+1} - t_j$,
  - **set** $w_{j+1} = w_j + h_j \, \phi(t_j, w_j, h_j)$.

$\boxed{\text{Adaptively choose step-size to satisfy given tolerance}}$

Adaptive Error Control (II)

- Given an **order-n** method
  - $w_0 = \alpha$.
  - for $j = 0, 1, \cdots$,

$$w_{j+1} \quad = \quad w_j + h\,\phi\left(t_j, w_j, h\right), \quad h = t_{j+1} - t_j,$$

  - **local truncation error** (LTE)

$$\tau_{j+1}(h) \quad = \quad \frac{y(t_{j+1}) - y(t_j)}{h} - \phi\left(t_j, y(t_j), h\right) = O\left(h^n\right).$$

- Given tolerance $\tau > 0$, we would like to estimate **largest** step-size h for which

$$|\tau_{j+1}(h)| \lesssim \tau.$$

Adaptive Error Control (II)

- Given an **order-n** method
  - $w_0 = \alpha$.
  - for $j = 0, 1, \cdots$,

$$w_{j+1} \quad = \quad w_j + h\,\phi\left(t_j, w_j, h\right), \quad h = t_{j+1} - t_j,$$

  - **local truncation error** (LTE)

$$\tau_{j+1}(h) \quad = \quad \frac{y(t_{j+1}) - y(t_j)}{h} - \phi\left(t_j, y(t_j), h\right) = O\left(h^n\right).$$

- Given tolerance $\tau > 0$, we would like to estimate **largest** step-size h for which

$$|\tau_{j+1}(h)| \lesssim \tau.$$

$$\boxed{\text{Approach: Estimate } \tau_{j+1}(h) \text{ with } \textbf{order-}(n+1) \text{ method}}$$

$$\widetilde{w}_{j+1} \quad = \quad \widetilde{w}_j + h\,\widetilde{\phi}\left(t_j, \widetilde{w}_j, h\right), \quad \text{for} \quad j \geq 0.$$

LTE Estimation (I)

- Assume $\quad w_j \approx y(t_j), \quad \widetilde{w}_j \approx y(t_j)$ (only estimating **LTE**).

LTE Estimation (I)

- Assume $w_j \approx y(t_j), \quad \widetilde{w}_j \approx y(t_j)$ (only estimating **LTE**).
- $\phi(t, w, h)$ is order-n method

$$
\begin{aligned}
\tau_{j+1}(h) \quad & \stackrel{def}{=} \quad \frac{y(t_{j+1}) - y(t_j)}{h} - \phi\left(t_j, y(t_j), h\right) \\
& \stackrel{w_j \approx y(t_j)}{\approx} \quad \frac{y(t_{j+1}) - \left(w_j + h\,\phi\left(t_j, w_j, h\right)\right)}{h} \\
& = \quad \frac{y(t_{j+1}) - w_{j+1}}{h} = O\left(h^n\right).
\end{aligned}
$$

LTE Estimation (I)

- Assume $\quad w_j \approx y(t_j), \quad \widetilde{w}_j \approx y(t_j) \quad$ (only estimating **LTE**).
- $\phi(t, w, h)$ is order-n method

$$\begin{aligned}
\tau_{j+1}(h) \quad & \stackrel{def}{=} \quad \frac{y(t_{j+1}) - y(t_j)}{h} - \phi\left(t_j, y(t_j), h\right) \\
& \stackrel{w_j \approx y(t_j)}{\approx} \quad \frac{y(t_{j+1}) - \left(w_j + h\,\phi\left(t_j, w_j, h\right)\right)}{h} \\
& = \quad \frac{y(t_{j+1}) - w_{j+1}}{h} = O\left(h^n\right).
\end{aligned}$$

- $\widetilde{\phi}(t, w, h)$ is order-$(n+1)$ method,

$$\begin{aligned}
\widetilde{\tau}_{j+1}(h) \quad & \stackrel{def}{=} \quad \frac{y(t_{j+1}) - y(t_j)}{h} - \widetilde{\phi}\left(t_j, y(t_j), h\right) \\
& \stackrel{\widetilde{w}_j \approx y(t_j)}{\approx} \quad \frac{y(t_{j+1}) - \left(\widetilde{w}_j + h\,\widetilde{\phi}\left(t_j, \widetilde{w}_j, h\right)\right)}{h} \\
& = \quad \frac{y(t_{j+1}) - \widetilde{w}_{j+1}}{h} = O\left(h^{n+1}\right).
\end{aligned}$$

LTE Estimation (III)

- Assume $\quad w_j \approx y(t_j), \quad \widetilde{w}_j \approx y(t_j) \quad$ (only estimating **LTE**).

LTE Estimation (III)

- Assume $\quad w_j \approx y(t_j), \quad \widetilde{w}_j \approx y(t_j) \quad$ (only estimating **LTE**).
- $\phi(t, w, h)$ is order-n method

$$\tau_{j+1}(h) \quad \approx \quad \frac{y(t_{j+1}) - w_{j+1}}{h} = O\left(h^n\right).$$

LTE Estimation (III)

- Assume $\quad w_j \approx y(t_j), \quad \widetilde{w}_j \approx y(t_j)$ (only estimating **LTE**).
- $\phi(t, w, h)$ is order-n method

$$\tau_{j+1}(h) \;\approx\; \frac{y(t_{j+1}) - w_{j+1}}{h} = O\left(h^n\right).$$

- $\widetilde{\phi}(t, w, h)$ is order-$(n+1)$ method,

$$\widetilde{\tau}_{j+1}(h) \;\approx\; \frac{y(t_{j+1}) - \widetilde{w}_{j+1}}{h} = O\left(h^{n+1}\right).$$

LTE Estimation (III)

- Assume $\quad w_j \approx y(t_j), \quad \widetilde{w}_j \approx y(t_j) \quad$ (only estimating **LTE**).
- $\phi(t, w, h)$ is order-n method

$$\tau_{j+1}(h) \quad \approx \quad \frac{y(t_{j+1}) - w_{j+1}}{h} = O\left(h^n\right).$$

- $\widetilde{\phi}(t, w, h)$ is order-$(n+1)$ method,

$$\widetilde{\tau}_{j+1}(h) \quad \approx \quad \frac{y(t_{j+1}) - \widetilde{w}_{j+1}}{h} = O\left(h^{n+1}\right).$$

- therefore

$$\begin{aligned}\tau_{j+1}(h) \quad &\approx \quad \frac{y(t_{j+1}) - \widetilde{w}_{j+1}}{h} + \frac{\widetilde{w}_{j+1} - w_{j+1}}{h} \\ &= \quad O\left(h^{n+1}\right) + \frac{\widetilde{w}_{j+1} - w_{j+1}}{h} = O\left(h^n\right).\end{aligned}$$

$$\boxed{\text{LTE estimate: } \tau_{j+1}(h) \approx \frac{\widetilde{w}_{j+1} - w_{j+1}}{h}}$$

step-size selection (I)

$$\boxed{\text{LTE estimate: } \tau_{j+1}(h) \approx \frac{\widetilde{w}_{j+1} - w_{j+1}}{h}}$$

step-size selection (I)

$$\boxed{\text{LTE estimate: } \tau_{j+1}(h) \approx \frac{\widetilde{w}_{j+1} - w_{j+1}}{h}}$$

- Since $\tau_{j+1}(h) = O\left(h^n\right)$, **assume**

$$\tau_{j+1}(h) \approx K\, h^n \quad \text{where } K \text{ is independent of } h.$$

step-size selection (I)

$$\boxed{\text{LTE estimate: } \tau_{j+1}(h) \approx \frac{\widetilde{w}_{j+1} - w_{j+1}}{h}}$$

- Since $\quad \tau_{j+1}(h) = O\left(h^n\right)$, **assume**

$$\tau_{j+1}(h) \approx K\, h^n \quad \text{where } K \text{ is independent of } h.$$

- K should satisfy

$$K\, h^n \approx \frac{\widetilde{w}_{j+1} - w_{j+1}}{h}.$$

- **Assume** LTE for new step-size $q\,h$ satisfies given tolerance ϵ:

$$|\tau_{j+1}(q\,h)| \le \epsilon, \quad \text{need to estimate } q.$$

step-size selection (II)

$$\boxed{\text{LTE estimate: } K\,h^n \approx \tau_{j+1}(h) \approx \frac{\widetilde{w}_{j+1} - w_{j+1}}{h}}$$

step-size selection (II)

$$\boxed{\text{LTE estimate: } K\,h^n \approx \tau_{j+1}(h) \approx \frac{\widetilde{w}_{j+1} - w_{j+1}}{h}}$$

- **Assume** LTE for $q\,h$ satisfies given tolerance ϵ:

$$\begin{aligned} |\tau_{j+1}(q\,h)| &\approx |K\,(q\,h)^n| = q^n\,|K\,h^n| \\ &\approx q^n \left|\frac{\widetilde{w}_{j+1} - w_{j+1}}{h}\right| \leq \epsilon. \end{aligned}$$

step-size selection (II)

$$\boxed{\text{LTE estimate: } K\, h^n \approx \tau_{j+1}(h) \approx \frac{\widetilde{w}_{j+1} - w_{j+1}}{h}}$$

- **Assume** LTE for $q\,h$ satisfies given tolerance ϵ:

$$\begin{aligned} |\tau_{j+1}(q\,h)| &\approx |K\ (q\,h)^n| = q^n\ |K\ h^n| \\ &\approx q^n \left| \frac{\widetilde{w}_{j+1} - w_{j+1}}{h} \right| \leq \epsilon. \end{aligned}$$

$$\boxed{\text{new step-size estimate: } q\,h \lesssim \left| \frac{\epsilon\, h}{\widetilde{w}_{j+1} - w_{j+1}} \right|^{\frac{1}{n}} h}$$

Summary

- Given **order-n** method

$$w_{j+1} \;=\; w_j + h\,\phi\left(t_j, w_j, h\right), \quad h = t_{j+1} - t_j, \quad j \geq 0,$$

- and given **order-$(n+1)$** method

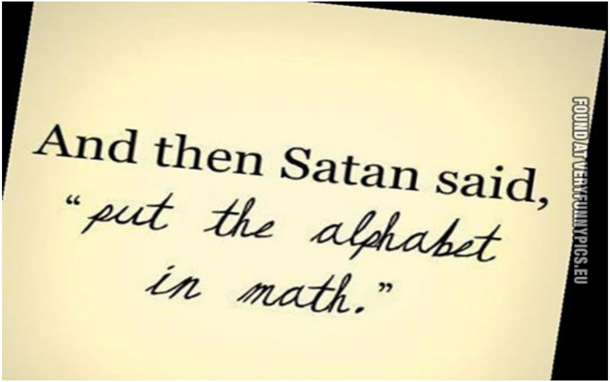
$$\widetilde{w}_{j+1} \;=\; \widetilde{w}_j + h\,\widetilde{\phi}\left(t_j, \widetilde{w}_j, h\right), \quad \text{for} \quad j \geq 0.$$

- for each j, compute

$$\begin{aligned} w_{j+1} &= w_j + h\,\phi\left(t_j, w_j, h\right), \\ \widetilde{w}_{j+1} &= w_j + h\,\widetilde{\phi}\left(t_j, w_j, h\right), \end{aligned}$$

- new step-size $q\,h$ should satisfy

$$q\,h \lesssim \left|\frac{\epsilon\,h}{\widetilde{w}_{j+1} - w_{j+1}}\right|^{\frac{1}{n}} h = \left|\frac{\epsilon}{\widetilde{\phi}\left(t_j, w_j, h\right) - \phi\left(t_j, w_j, h\right)}\right|^{\frac{1}{n}} h.$$


And then Satan said,
"put the alphabet
in math."
FOUND AT VERYFUNNYPICS.EU

Runge-Kutta-Fehlberg: 4th order method, 5th order estimate

$$
\begin{aligned}
w_{j+1} &= w_j + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5, \\
\widetilde{w}_{j+1} &= w_j + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6, \quad \text{where}
\end{aligned}
$$

Runge-Kutta-Fehlberg: 4^{th} order method, 5^{th} order estimate

$$
\begin{aligned}
w_{j+1} &= w_j + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5, \\
\widetilde{w}_{j+1} &= w_j + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6, \quad \text{where} \\
k_1 &= h\, f\left(t_j, w_j\right),
\end{aligned}
$$

Runge-Kutta-Fehlberg: 4^{th} order method, 5^{th} order estimate

$$
\begin{aligned}
w_{j+1} &= w_j + \frac{25}{216} k_1 + \frac{1408}{2565} k_3 + \frac{2197}{4104} k_4 - \frac{1}{5} k_5, \\
\widetilde{w}_{j+1} &= w_j + \frac{16}{135} k_1 + \frac{6656}{12825} k_3 + \frac{28561}{56430} k_4 - \frac{9}{50} k_5 + \frac{2}{55} k_6, \quad \text{where} \\
k_1 &= h\, f\left(t_j, w_j\right), \\
k_2 &= h\, f\left(t_j + \frac{h}{4}, w_j + \frac{1}{4}\, k_1\right), \\
k_3 &= h\, f\left(t_j + \frac{3h}{8}, w_j + \frac{3}{32}\, k_1 + \frac{9}{32}\, k_2\right), \\
k_4 &= h\, f\left(t_j + \frac{12h}{13}, w_j + \frac{1932}{2197}\, k_1 - \frac{7200}{2197}\, k_2 + \frac{7296}{2197}\, k_3\right), \\
k_5 &= h\, f\left(t_j + h, w_j + \frac{439}{216}\, k_1 - 8\, k_2 + \frac{3680}{513}\, k_3 - \frac{845}{4104}\, k_4\right), \\
k_6 &= h\, f\left(t_j + \frac{h}{2}, w_j - \frac{8}{27}\, k_1 + 2\, k_2 - \frac{3544}{2565}\, k_3 + \frac{1859}{4104}\, k_4 - \frac{11}{40}\, k_5\right).
\end{aligned}
$$

Dormand-Prince (ode45): 4^{th} order method, 5^{th} order estimate

$$\begin{aligned}
k_1 &= h\,f\left(t_j, w_j\right), \\
k_2 &= h\,f\left(t_j + \frac{h}{5}, w_j + \frac{1}{5}\,k_1\right), \\
k_3 &= h\,f\left(t_j + \frac{3h}{10}, w_j + \frac{3}{40}\,k_1 + \frac{9}{40}\,k_2\right), \\
k_4 &= h\,f\left(t_j + \frac{4h}{5}, w_j + \frac{44}{45}\,k_1 - \frac{56}{15}\,k_2 + \frac{32}{9}\,k_3\right), \\
k_5 &= h\,f\left(t_j + \frac{8h}{9}, w_j + \frac{19372}{6561}\,k_1 - \frac{25360}{2187}\,k_2 + \frac{64448}{6561}\,k_3 - \frac{212}{729}\,k_4\right), \\
k_6 &= h\,f\left(t_j + h, w_j - \frac{9017}{3168}\,k_1 - \frac{355}{33}\,k_2 + \frac{46732}{5247}\,k_3 + \frac{49}{176}\,k_4 - \frac{5103}{18656}\,k_5\right), \\
w_{j+1} &= w_j + \frac{35}{384}k_1 + \frac{500}{1113}k_3 + \frac{125}{192}k_4 - \frac{2187}{6784}k_5 + \frac{11}{84}k_6, \\
k_7 &= h\,f\left(t_{j+1}, w_{j+1}\right).
\end{aligned}$$

$$\widetilde{w}_{j+1} = w_j + \frac{5179}{57600}k_1 + \frac{7571}{16695}k_3 + \frac{393}{640}k_4 - \frac{92097}{339200}k_5 + \frac{187}{2100}k_6 + \frac{1}{40}k_7.$$

Runge-Kutta-Fehlberg vs. Dormand-Prince

- Both methods require 6 function evaluations per step.
- Runge-Kutta-Fehlberg ensures a small **LTE** of fourth order method.
- Dormand-Prince chooses coefficients to minimize **LTE** of fifth order method.
- Dormand-Prince better allows an extrapolation step for better integration accuracy.
- Dormand-Prince is basis for `ode45` in matlab.

Runge-Kutta-Fehlberg: step-size selection procedure

- compute a conservative value for q:

$$q = \left| \frac{\epsilon\, h}{\mathbf{2}\,(\widetilde{w}_{j+1} - w_{j+1})} \right|^{\frac{1}{4}}.$$

Runge-Kutta-Fehlberg: step-size selection procedure

- compute a conservative value for q:

$$q = \left| \frac{\epsilon\, h}{\mathbf{2}\left(\widetilde{w}_{j+1} - w_{j+1}\right)} \right|^{\frac{1}{4}}.$$

- make restricted step-size change:

$$h = \begin{cases} 0.1\, h, & \text{if } q \le 0.1, \\ 4\, h, & \text{if } q \ge 4. \end{cases}$$

Runge-Kutta-Fehlberg: step-size selection procedure

- compute a conservative value for q:

$$q = \left| \frac{\epsilon\, h}{\mathbf{2}\left(\widetilde{w}_{j+1} - w_{j+1}\right)} \right|^{\frac{1}{4}}.$$

- make restricted step-size change:

$$h = \begin{cases} 0.1\, h, & \text{if } q \le 0.1, \\ 4\, h, & \text{if } q \ge 4. \end{cases}$$

- step-size can't be too big:

$$h = \mathbf{min}\left(h, h_{\mathbf{max}}\right).$$

Runge-Kutta-Fehlberg: step-size selection procedure

- compute a conservative value for q:

$$q = \left| \frac{\epsilon\, h}{\mathbf{2}\left(\widetilde{w}_{j+1} - w_{j+1}\right)} \right|^{\frac{1}{4}}.$$

- make restricted step-size change:

$$h = \begin{cases} 0.1\, h, & \text{if } q \leq 0.1, \\ 4\, h, & \text{if } q \geq 4. \end{cases}$$

- step-size can't be too big:

$$h = \mathbf{min}\left(h, h_{\mathbf{max}}\right).$$

- step-size can't be too small:

$$\textbf{if} \quad h < h_{\mathbf{min}} \quad \textbf{then} \quad \textit{declare failure.}$$