Numerical Solution of Differential Equations

IIIT, Allahabad

Instructor: Srijit Bhattacharjee

Solving Ordinary Differential Equations

- An ODE is in general has the following form

$$f((y(x), y'(x), y"(x), \cdots, x) = 0$$

- For first order ODE Euler method provides a numerical or approximate solution.
- Solve $\dfrac{dy}{dx} = f(x, y)$ with initial condition: $y(x_0) = y_0$.
- Taylor expanding y wrt x_0 and keeping only first order terms in step size h:

$$y(x_0 + h) = y(x_1) = y_1 = y_0 + hf(x_0, y_0)$$

- Next

$$y_2 = y_1 + hf(x_1 y_1)$$
$$y_3 = y_2 + hf(x_2, y_2) \cdots$$

Euler Method

- General: $y_{n+1} = y_n + hf(x_n, y_n),\ n = 0, 1, 2, \cdots$
- Problems: Too slow to converge. Error is $O(h^2)$.
  No scope to improve the value of y.
- In modified Euler's method one uses

$$y_1 = y_0 + \int_{x_i}^{x_{i+1}} f(x, y)dx$$

  and the integration is done using Trapezoidal method.

$$y_1 = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_1, y(x_1))]$$

- $y(x_1)$ is obtained using Euler's method. $y_1 = y_0 + hf(x_0, y_0)$, and the first approximation of y_1 becomes

$$y_1^{(1)} = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_1, y_1^{(0)}))]$$

Euler Method

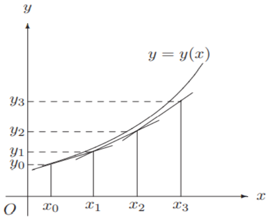
- In a generic step

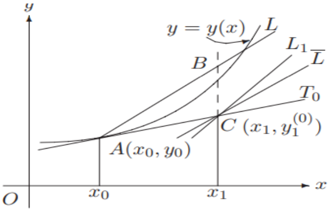
$$y_{i+1}^{(k+1)} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_i, y_i^{(k)}))], k = 0, 1, 2, \cdots \; i = 0, 1, 2, \cdots$$

- The integration $\int_{x_i}^{x_{i+1}} f(x, y)$ can also be done by Monte Carlo method when f is independent of y.
- One can obtain the iteration formula using MC estimator

$$y(x_i) = y(x_{i-1}) + \frac{x_i - x_{i-1}}{N} \sum_{j=1}^{N} f(x_j)$$

EM and Modified EM





Examples

- Find the solution of the equation for $x = 0.5$,

$$y' = x^2 - y,\ y(0) = 1$$

- ODEs are very common in Engineering and Science. Example bacterial growth equation:

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right),\ N(0) = N_0$$

System of ODEs

- Evaluation of values of dependent variables y, for more than one ODEs can also be obtained by the methods discussed.
- Enzyme reactions:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} SE \xrightarrow{k_2} P + E$$

$$\frac{ds}{dt} = -k_1 es + k_{-1} c, \quad \frac{de}{dt} = -k_1 es + (k_{-1} + k_2)c$$
$$\frac{dc}{dt} = k_1 es - (k_{-1} + k_2)c, \quad \frac{dp}{dt} = k_2 c.$$

- One can write the system of equations symbolically as

$$y_i' = f_i(x, y_1, y_2, \cdots), \; y_i(x_0) = y_{i0}$$

- The Euler's method can be written as

$$y_{i,j+1} = y_{i,j} + hf_i(x, y_{1,j}, y_{2,j}, \cdots y_{m,j}); m = 1, 2, \cdots 4; j = 0, 1, 2, \cdots N$$

Performance of Euler Methods

- Local truncation error of modified Euler method is $O(h^3)$ whereas for Euler method, it is $O(h^2)$.
- Compute $y(x)$ solving $y' + 2y = x^2 e^{-2x}$, $y(0) = 1$ by EM and MEM with with $h = 0.1$ and $h = 0.05$ respectively.
- Exact solution $y(x) = \frac{1}{3}(e^{-2x} + 3)$

E = Euler Method
M E = Modifed Euler Method

x	E h = 0.1	E h = 0.05	M E h = 0.1	M E h = 0.05	Exact
0.000	1.000000000	1.000000000	1.000000000	1.000000000	1.000000000
0.100	0.800000000	0.810113105	0.820409365	0.819331760	0.819003663
0.200	0.640818731	0.657393464	0.674403812	0.672642391	0.672107566
0.300	0.515336265	0.535590690	0.556553290	0.554402342	0.553750941
0.400	0.417208317	0.439092731	0.461944052	0.459617490	0.458914649
0.500	0.340955917	0.363016799	0.386268321	0.383916533	0.383207751
0.600	0.281961719	0.303216926	0.325840313	0.323564517	0.322880195
0.700	0.236412367	0.256242290	0.277567879	0.275432049	0.274791217
0.800	0.201213145	0.219269254	0.238899650	0.236940201	0.236353524
0.900	0.173891893	0.190022158	0.207760869	0.205994363	0.205466518
1.000	0.152502724	0.166692376	0.182486361	0.180915522	0.180447044

Stability of Euler Method

- An iteration scheme to solve IVP is stable if small perturbations in the initial conditions do not cause the numerical approximation to diverge away from the true solution provided the true solution of the IVP is bounded
- Consider: $y'(t) = \lambda y(t)$; $y(0) = y_0$. $t \in [0, T]$
- Solution: $y(t) = y_0 e^{\lambda t}$
  Take a slightly perturbed initial condition:
  $y(0) = y_0 + \epsilon \implies y_\epsilon(t) = (y_0 + \epsilon)e^{\lambda t}$
- For $\lambda < 0$, the solution may become large and unstable even for small ϵ.
- How?

EM is conditionally stable

- For EM:

$$y_{n+1} = y_n + hf = y_n + \lambda h y_n \implies y_n = (1+\lambda h)^n y_0$$

- The stability criteria must be $|1+\lambda h| < 1 \implies h < \frac{-2}{\lambda}$
- Euler backward method is unconditionally stable: $y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$
- The growth factor : $y_n = (\frac{1}{1-\lambda h})^n y_0$.
- Sometimes an ODE may become *stiff* if the step size is not chosen appropriately. The stepsize would have to be very small to have a stable iteration scheme.

  Ex. Take $\lambda = -1000, -1$ and try with $h = 0.05$ for the above problem.

Runge-Kutta Method

- How to incorporate higher order derivatives in Taylor series?
- Direct computation of higher order derivatives may not be always easy to obtain.
- Computation of derivatives are not always economic in terms of numerical techniques.
- Runge-Kutta method offers a way to incorporate higher order terms in a single step algorithm.

Runge-Kutta Method 2nd order

- Recall modified Euler Method:

$$y_1^{(1)} = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_1, y_1^{(0)}))]$$

- The above formula can be recast as:

$$y_1^{(1)} = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_0 + h, y_1^{(0)}))]$$

- Putting $y_1^{(0)}) = y_0 + hf(x_0, y_0)$:

$$y_1^{(1)} = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_0 + h, y_0 + hf(x_0, y_0))]$$

- Call $k_1 = f(x_0, y_0)$ and $k_2 = f(x_0 + h, y_0 + hk_1)$
- The 2nd order R-K algorithm:

$$y_1 = y_0 + \frac{h}{2}(k_1 + k_2)$$

R-K 2 General consideration

- General method:

$$y_{i+1} = y_i + hF(x_i, y_i, h) = y_i + h(w_1 k_1 + w_2 k_2)$$

  with $k_2 = f(x_i + ah, y_i + bhk_1)$.

- The above formula can be recast as:

  $y_{i+1} = y_i + h[w_1 f(x_i, y_i) + w_2(f(x_i, y_i) + ahf'(x_i, y_i) + bhf(x_i, y_i)(f_y(x_i, y_i) + O(h^2))]$

- $y_{i+1} = y_i + h[w_1 f(x_i, y_i) + w_2(f(x_i, y_i)] + h^2 w_2[af'(x_i, y_i) + bf(x_i, y_i)(f_y(x_i, y_i)] + O(h^3)]$

- Taylor expansion of y:

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

- Comparing:

$$w_1 + w_2 = 1;\ w_2 a = 1/2;\ w_2 b = 1/2$$

R-K 2

- We have chosen $w_1 = w_2 = 1/2$ and $a = b = 1$ parametrization.
- Other parametrizations possible. $w_1 = 1/4, w_2 = 3/4$ $a = b = 2/3$
- Local truncation error $O(h^3)$.

R-K 4 Method

- $$y_{i+1} = y_i + hF(x_i, y_i, h) = y_i + h(w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4)$$
- $$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + ah, y_i + bk_1) \\ k_3 &= f(x_i + ch, y_i + dk_1 + ek_2) \\ k_4 &= f(x_i + fh, y_i + d_1 k_1 + e_1 k_2 + g_1 k_3) \end{aligned}$$
- The classical R-K 4 method is obtained by the following choices:

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + \frac{h}{2}, y_i + \frac{h}{2} k_1) \\ k_3 &= f(x_i + \frac{h}{2}, y_i + \frac{h}{2} k_2) \\ k_4 &= f(x_i + h, y_i + hk_3) \end{aligned}$$

R-K 4 Method Algorithm

- $$y_{i+1} = y_i + \frac{h}{3}[k_1 + 2k_2 + 2k_3 + k_4]$$

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1) \\ k_3 &= f(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2) \\ k_4 &= f(x_i + h, y_i + hk_3) \end{aligned}$$

- Truncation error: $O(h^5)$.

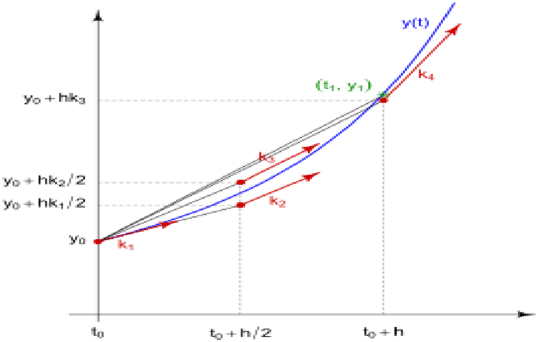


Figure: Source: Wikipedia