SIMULATION OF DYNAMICAL SYSTEMS

Robert Bosch Centre for Cyber-Physical Systems

Table of Contents

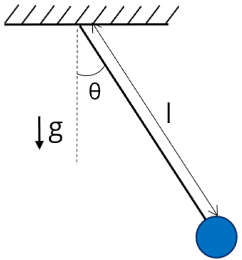
Background

- First order ODE:
$$\frac{dx}{dt} = f(x); x(t_0) = x_0$$
- Higher order ODE $\rightarrow$ System of linear/nonlinear equations $\rightarrow$ x is a vector. e.g.,
$$\frac{d^2x}{dt^2} + A(x)\frac{dx}{dt} + B(x)x = C(x)$$
- Analytical Solutions: In many situations an analytical solution is not possible!
- Numerical Solutions: A set of discrete points that approximate the function $x(t)$

Example

Pendulum Dynamics: Non-Linear (approximation-free) ODE

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} sin(\theta) = 0$$



Overview of Numerical Methods

- Start with an initial value
- Then, estimate the value at a 2nd nearby point $\rightarrow$ 3rd point $\rightarrow \ldots$
- Single-step and multistep approach
  - Single step approach: $x_i \rightarrow x_{i+1}$
  - Multi-step approach: $\ldots, x_{i-2}, x_{i-1}, x_i \rightarrow x_{i+1}$
- Explicit and implicit approach
  - Right hand side in explicit method: known values

$$x_{i+1} = F(t_i, t_{i+1}, x_i)$$

  - Right hand side in implicit method: unknown value

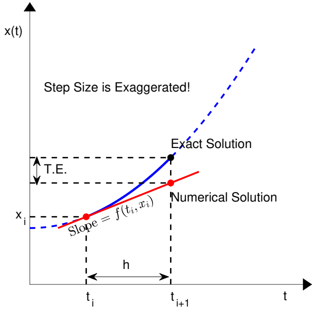
$$x_{i+1} = F(t_i, t_{i+1}, x_{i+1})$$

Euler's Explicit Method

- Euler's Explicit Method

$$x_{i+1} = x_i + f(x_i)h, h = t_{i+1} - t_i$$

- Truncation Error $T.E. = \frac{h^2}{2}\frac{d^2x}{dt^2} = O(h^2)$



Newton's Method

Algorithm 1 Newton's method

for $k = 0$ to $N - 1$ **do**

$\Delta y \leftarrow -\frac{g(y=y_k)}{g'(y=y_k)}$

$y_{k+1} \leftarrow y_k + \Delta y$

$y_{\text{NEW}} \leftarrow y_{k+1}$

if $\Delta y < 0.001$ **then**

End Loop

end if

end for

$x_{i+1} = y_{\text{NEW}}$

Runge-Kutta Methods

The basic idea of Runge-Kutta methods is to approximate the integral by a weighted average of slopes and approximate slopes at a number of points in the interval $[t_i, \ t_{i+1}]$.

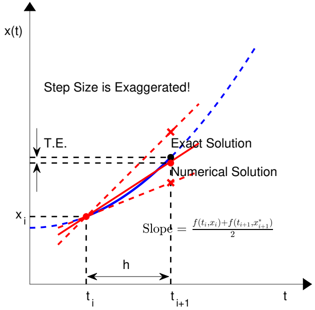
- Runge-Kutta method of second order: Uses two slopes to arrive at second order

$$x_{i+1} = x_i + \frac{1}{2}(k_1 + k_2)$$
$$k_1 = hf(t_i, x_i), \ k_2 = hf(t_i + h, x_i + k_1)$$

- The Trunction Error in this method $T.E. = O(h^3)$

Runge-Kutta method of second order

- Truncation Error $T.E. = \frac{h^2}{2}\frac{d^2x}{dt^2} = O(h^3)$



Runge-Kutta Methods

- Runge-Kutta method of fourth order: Uses four slopes to arrive at fourth order

$$\begin{aligned} x_{i+1} &= x_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= hf(t_i, x_i) \\ k_2 &= hf(t_i + h/2, x_i + k_1/2) \\ k_3 &= hf(t_i + h/2, x_i + k_2/2) \\ k_4 &= hf(t_i + h, x_i + k_3) \end{aligned}$$

  - The Truncation Error of this method $T.E. = O(h^5)$

Example

- Solve the ODE:

$$\frac{dx}{dt} = -1.2x + 7e^{-0.3t}$$

from $t = 0$ to $t = 8$, with $x(0) = 3$, using

1. Euler's explicit method
2. RK 2nd order
3. RK 4th order

using $h = 0.5$.

Higher order ODE

- Pendulum Dynamics: Non-Linear (approximation-free) ODE

$$\frac{d^2\theta}{dt^2} + \frac{b}{l}\frac{d\theta}{dt} + \frac{g}{l} sin(\theta) = 0$$

- State space representation:

$$x_1 = \theta, x_2 = \dot{\theta}$$
$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{l} sin(x_1) - \frac{b}{l} x_2 \end{bmatrix} = F(x)$$

Higher order ODE

- Pendulum Dynamics: Non-Linear (approximation-free) ODE

$$\frac{d^2\theta}{dt^2} + \frac{b}{l}\frac{d\theta}{dt} + \frac{g}{l} sin(\theta) = 0$$

- $g = 9.81, b = 1, l = 1$
- $\theta_0 = \pi/3, \dot{\theta}_0 = 0$
- $t_{span} = [0, 8]$

ode45

- $[T_{OUT}, Y_{OUT}] = \text{ode45}(\text{ODEFUN}, T_{SPAN}, Y_0)$
- Integrates the system of differential equations $y' = f(t, y)$ from time $T_{SPAN}(1)$ to $T_{SPAN}(\text{end})$ with initial conditions Y_0.
- Each row in the solution array Y_{OUT} corresponds to a time in the column vector T_{OUT}.
- $[T_{OUT}, Y_{OUT}] = \text{ode45}(\text{ODEFUN}, T_{SPAN}, Y_0, \text{OPTIONS})$ specifies integration option values in the fields of a structure, OPTIONS. Create the options structure with odeset.

ode45

- $[T_{OUT}, Y_{OUT}] = \text{ode45}(\text{ODEFUN}, T_{SPAN}, Y_0)$
- ODEFUN is a function handle. For a scalar, T and a vector, Y, ODEFUN(T, Y) must return a column vector corresponding to $f(t, y)$.
- T_{SPAN} is a two-element vector $[T_0\, T_{FINAL}]$ or a vector with several time points $[T_0\, T_1 \ldots T_{FINAL}]$. If you specify more than two time points, ode45 returns interpolated solutions at the requested times.
- Y_O is a column vector of initial conditions, one for each equation.

ode45

- Solve $y' = 2t$, for the time interval $t_{span} = [0, 5]$ and initial condition $y_0 = 0$;
- Sample code:

$$tspan = [0, 5];$$
$$y0 = 0;$$
$$[t, y] = ode45(@(t, y)\ 2 * t, tspan, y0);$$
plot(t,y,'-o')

Reference

- Numerical Methods by S.R.K Iyengar and R.K. Jain