ME242 – MECHANICAL ENGINEERING SYSTEMS

LECTURE 23:

- Numerical Methods

INTEGRATION OF ODE's

Given a scalar, first-order differential equation of the form

$$\frac{dy}{dt} = f(y,t) \qquad \text{with} \qquad y(t_o) = y_o$$

Goals:

- Advancement of the system in time by integration of ODE
- The quantity being integrated, f, is itself a function of the result of the integration, y.

Method:

- We seek to approximate the solution $y(t)$ at timestep $t_{n+1}=t_n+h$ given t_o and the solution at the previously-computed timesteps t_1 to t_n

Note:

- ODE's of higher order are generalization of this first order case

INTEGRATION OF ODE's – RUNGE-KUTTA

The Runge-Kutta methods can be written in the general form

$$k_1 = f(y_n, t_n)$$
$$k_2 = f(y_n + \beta_1 h k_1, t_n + \alpha_1 h)$$
$$k_3 = f(y_n + \beta_2 h k_1 + \beta_3 h k_2, t_n + \alpha_2 h)$$
$$\vdots$$
$$y_{n+1} = y_n + \gamma_1 h k_1 + \gamma_2 h k_2 + \gamma_3 h k_3 + \cdots$$

The constants α_i, β_i, and γ_i are selected to match as many terms as possible of the exact solution

$$y(t_{n+1}) = y(t_n) + h y'(t_n) + \frac{h^2}{2} y''(t_n) + \frac{h^3}{6} y'''(t_n) + \cdots$$

- These are explicit and "self starting" methods
- As the number of intermediate steps k_i increases, the accuracy also increases

INTEGRATION OF ODE's – RUNGE-KUTTA

Given our ODE

$$\frac{dy}{dt} = f(y,t) \qquad \text{with} \qquad y(t_o) = y_o$$

We can compute the derivatives of the Taylor series expansion

$$y(t_{n+1}) = y(t_n) + h y'(t_n) + \frac{h^2}{2} y''(t_n) + \frac{h^3}{6} y'''(t_n) + \cdots$$

as

$$y' = \frac{dy}{dt} = f(y,t)$$
$$y'' = \frac{dy'}{dt} = \frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt} = f_t + f f_y$$
$$y''' = \frac{dy''}{dt} = \frac{d}{dt}(f_t + f f_y) = \frac{\partial}{\partial t}(f_t + f f_y) + \frac{\partial}{\partial y}(f_t + f f_y)\frac{dy}{dt} = f_{tt} + f_t f_y + 2 f f_{yt} + f_y^2 f + f^2 f_{yy}$$

INTEGRATION OF ODE's – RUNGE-KUTTA

Second Order Runge Kutta (RK2)

$$k_1 = f(y_n, t_n)$$

$$\begin{aligned} k_2 &= f(y_n + \beta_1 h k_1, t_n + \alpha_1 h) \\ &\approx f(y_n, t_n) + f_y(y_n, t_n)\beta_1 h f(y_n, t_n) + f_t(y_n, t_n)\alpha_1 h \end{aligned}$$

$$\begin{aligned} y_{n+1} &= y_n + \gamma_1 h k_1 + \gamma_2 h k_2 \\ &\approx y_n + \gamma_1 h f(y_n, t_n) \\ &\quad + \gamma_2 h \left[f(y_n, t_n) + f_y(y_n, t_n)\beta_1 h f(y_n, t_n) + f_t(y_n, t_n)\alpha_1 h \right] \\ &= y_n + (\gamma_1 + \gamma_2) h f(y_n, t_n) + \gamma_2 h^2 \beta_1 f_y(y_n, t_n) f(y_n, t_n) + \gamma_2 h^2 \alpha_1 f_t(y_n, t_n) \end{aligned}$$

With this scheme, we seek to match the exact solution

$$y(t_{n+1}) = y(t_n) + h f(y_n, t_n) + \frac{h^2}{2}\left(f_t(y_n, t_n) + f(y_n, t_n) f_y(y_n, t_n) \right) + \cdots$$

INTEGRATION OF ODE's – RUNGE-KUTTA

Second Order Runge Kutta (RK2)

Matching coefficients to as high an order as possible, we require

$$\left.\begin{aligned} \gamma_1 + \gamma_2 &= 1 \\ \gamma_2 h^2 \beta_1 &= \frac{h^2}{2} \\ \gamma_2 h^2 \alpha_1 &= \frac{h^2}{2} \end{aligned}\right\} \Rightarrow \beta_1 = \alpha_1, \gamma_2 = \frac{1}{2\alpha_1}, \gamma_1 = 1 - \frac{1}{2\alpha_1}.$$

Thus, the general form of the two-step second-order RK isc

$$k_1 = f(y_n, t_n)$$

$$k_2 = f(y_n + \alpha_1 h k_1, t_n + \alpha_1 h)$$

$$y_{n+1} = y_n + \left(1 - \frac{1}{2\alpha_1}\right) h k_1 + \frac{1}{2\alpha_1} h k_2$$

α_1: Free parameter

INTEGRATION OF ODE's – RUNGE-KUTTA

Second Order Runge Kutta (RK2)

α_1=1/2 Midpoint Method

α_1=1 Predictor-Corrector Method

$$y^*_{n+1} = y_n + hf(y_n, t_n)$$
$$y_{n+1} = y_n + \frac{h}{2}\left[f(y_n, t_n) + f(y^*_{n+1}, t_{n+1})\right]$$

INTEGRATION OF ODE's – RUNGE-KUTTA

Consider the following scalar problem

$$\dot{y} = \lambda y \qquad \text{with} \qquad y(0) = y_o$$

We note that the analytical solution to this problem is

$$y(t) = y_o e^{\lambda t}$$

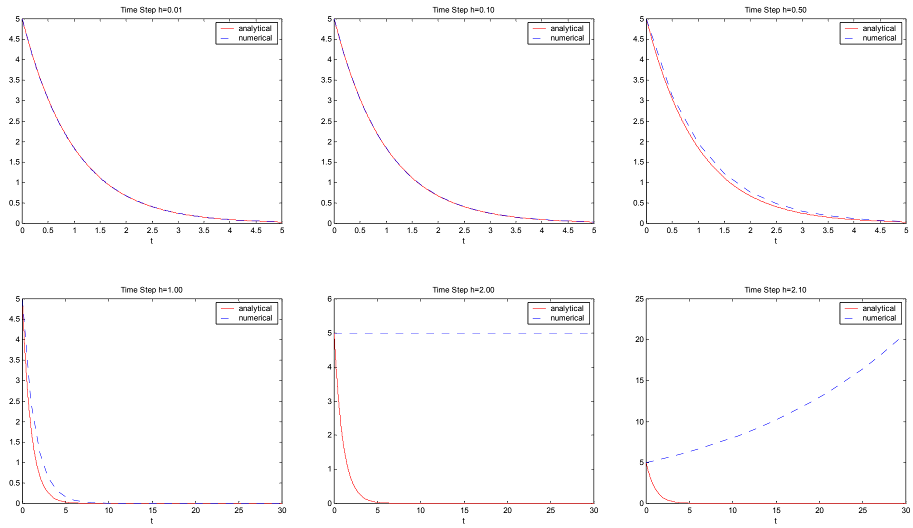
We now use the second order R-K to find a numerical solution

2nd order R-K Method:

Lecture23a.m

$$k_1 = \lambda y_n$$
$$k_2 = \lambda(y_n + \alpha_1 h \lambda y_n) = \lambda y_n (1 + \alpha_1 h \lambda)$$
$$y_{n+1} = y_n + \left(1 - \frac{1}{2\alpha_1}\right) h \lambda y_n + \frac{1}{2\alpha_1} h \lambda y_n (1 + \alpha_1 h \lambda)$$

INTEGRATION OF ODE's – RUNGE-KUTTA



INTEGRATION OF ODE's - STABILITY

Consider the following scalar problem

$$\dot{y} = \lambda y \qquad \text{with} \qquad y(0) = y_o$$

2nd order Runge-Kutta Method:

$$y_{n+1} = y_n + \left(1 - \frac{1}{2\alpha_1}\right) h\lambda y_n + \frac{1}{2\alpha_1} h\lambda y_n \left(1 + \alpha_1 h\lambda\right)$$

Thus, assuming h constant, the solution at time step n is

$$y_{n+1} = \left(1 + h\lambda + \frac{h^2\lambda^2}{2}\right) y_n \equiv \sigma y_n \Rightarrow \sigma = 1 + h\lambda + \frac{h^2\lambda^2}{2}$$

For large n, the numerical solution remains stable iff

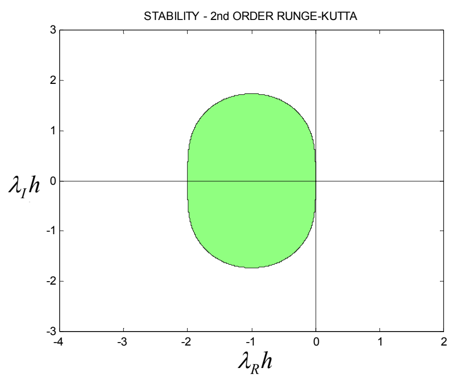
$$|\sigma| \le 1 \qquad \text{assuming } \lambda = \lambda_R + i\lambda_I$$

INTEGRATION OF ODE's - STABILITY

Consider the following scalar problem

$$\dot{y} = \lambda y \qquad \text{with} \qquad y(0) = y_o$$

2nd Order Runge-Kutta Method – Stability Region:



Stability_RK2.m

INTEGRATION OF ODE's - ACCURACY

Consider the following scalar problem

$$\dot{y} = \lambda y \qquad \text{with} \qquad y(0) = y_o$$

Exact Solution:

$$y(t_{n+1}) = y_n e^{\lambda h} = y_n \left(1 + \lambda h + \frac{\lambda^2 h^2}{2} + \frac{\lambda^3 h^3}{6} + \cdots \right)$$

Example: 2nd Order Runge-Kutta Method:

$$y_{n+1} = \left(1 + h\lambda + \frac{h^2 \lambda^2}{2} \right) y_n$$

By comparison, the leading error is proportional to h^3

2nd Order Runge-Kutta is second-order accurate

INTEGRATION OF ODE's – RUNGE-KUTTA

A popular 4th Order Runge-Kutta method can be written as

$$k_1 = f(y_n, t_n)$$

$$k_2 = f(y_n + \frac{h}{2}k_1, t_n + \frac{h}{2})$$

$$k_3 = f(y_n + \frac{h}{2}k_2, t_n + \frac{h}{2})$$

$$k_4 = f(y_n + hk_3, t_n + h)$$

$$y_{n+1} = y_n + \frac{h}{6}k_1 + \frac{h}{3}\left(k_2 + k_3\right) + \frac{h}{6}k_4$$