

Fourth Order Runge-Kutta Method

Computational Physics

The Runge-Kutta Method

Outline

- Review 2nd Order Runge-Kutta Method
- 4th Order Runge-Kutta Equations
- Implementation in MATLAB

Second Order Runge-Kutta

$$\frac{dy}{dt} = f(y, t)$$

**Differential
Equation**

$$y\left(t + \frac{\Delta t}{2}\right) = y(t) + f(y(t), t) \frac{\Delta t}{2}$$

**Estimate value
of y at half-step
(Euler Method)**

$$y(t + \Delta t) = y(t) + f\left(y\left(t + \frac{\Delta t}{2}\right), t + \frac{\Delta t}{2}\right) \Delta t$$

**Use value at
half-step to
find new estimate
of derivative**

Fourth Order Runge-Kutta

$$y(t + \Delta t) = y(t) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)\Delta t$$

Estimate of derivative in interval

$$k_1 = f(y(t), t) \quad \leftarrow \text{Value at beginning of interval}$$

$$k_2 = f\left(y(t) + k_1 \frac{\Delta t}{2}, t + \frac{\Delta t}{2}\right) \quad \leftarrow \text{Two estimates of value at mid-point}$$

$$k_3 = f\left(y(t) + k_2 \frac{\Delta t}{2}, t + \frac{\Delta t}{2}\right)$$

$$k_4 = f(y(t) + k_3 \Delta t, t + \Delta t) \quad \text{Estimate of value at end of interval}$$

A MATLAB Approach

Let Z be solution vector:

$$Z = [X \ V]$$

Differential Equation:

$$dZ/dt = [dX/dt \ dV/dt]$$

$$dZ/dt_i = [Z_i(2) \ -g] \quad \% \text{ falling ball (Euler)}$$

Algorithm to Advance Calculation:

$$Z_{i+1} = Z_i + dZ/dt_i * \Delta t$$

```

% Example: Falling Ball with air resistance
% 4th Order Runge Kutta Method
% NOTE: Z = [X V]
clear
g = 9.8;      % decay constant
b = 0.01;    % drag coefficient
X0 = 0;      % initial values
V0 = 0;

dt = .5;     % step size
t = 0:dt:10; % array of times for calculation
Z = zeros(length(t),2); % define Z array
Z(1,:) = [X0 V0]; % initialize

% loop
for i=1:length(t)-1
    Z(i+1,:) = Z(i,:) + dZdt(g,b,Z(i,:),dt)*dt;
end

% make a nice graph of Velocity
plot(t,Z(:,2),'md','MarkerSize',6)
xlabel('t (s)');
ylabel('V (m/s)');

```

Implementation in MATLAB

Z is a 2xn array with
BOTH X & V

X1	V1	first time
X2	V2	second time
X3	V3	third time
.....		
Xn	Vn	last time

Z(i,:) is Xi Vi from the ith row

Function called to
compute derivative by
4th order Runge-Kutta

Z(:,2) is all V values from
the second column of Z

FILE: dZdt.m

```
function dZdt = falling_ball_function(g,b,Z,dt);  
% Computes derivative by RK4  NOTE: Z = [X V]  
V0 = Z(2);  
X0 = Z(1);
```

```
dVdt1 = -g - sign(V0)*b*V0^2;  
dXdt1 = V0;
```

```
Vmid1 = V0 + dVdt1*dt/2;  
Xmid1 = X0 + dXdt1*dt/2;
```

```
dVdt2 = -g - sign(Vmid1)*b*Vmid1^2;  
dXdt2 = Vmid1;
```

```
Vmid2 = V0 + dVdt2*dt/2;  
Xmid2 = X0 + dXdt2*dt/2;
```

```
dVdt3 = -g - sign(Vmid2)*b*Vmid2^2;  
dXdt3 = Vmid2;
```

```
Vfinal = V0 + dVdt3*dt;  
Xfinal = X0 + dXdt3*dt;
```

```
dVdt4 = -g - sign(Vfinal)*b*Vfinal^2;  
dXdt4 = Vfinal;
```

```
dZdt = [(dXdt1+2*dXdt2+2*dXdt3+dXdt4)/6 (dVdt1+2*dVdt2+2*dVdt3+dVdt4)/6];
```

The 4 derivative estimates



Compute Weighted Averages:

[dXdt dVdt]

