Ordinary Differential Equations-Initial Value Problem

- Taylor's series method
- Euler method
- Runge Kutta method
- Computer-based solutions
  - o IVPAG subroutine (Fortran IMSL subroutine that Solves an initial-value problem for ordinary differential equations using either Adams-Moulton's or Gear's BDF method)
  - o MATLAB solution of initial value problems for ordinary differential equations (ode23, ode45, ode113, ode15s, ode23s, ode23t, ode23tb))

Differential Equations

Differential Equations

Ordinary Differential Equations

$$\frac{d^2 v}{dt^2} + 6tv = 1$$

involve one or more Ordinary derivatives of unknown functions

Partial Differential Equations

$$\frac{\partial^2 u}{\partial y^2} - \frac{\partial^2 u}{\partial x^2} = 0$$

involve one or more partial derivatives of unknown functions

Auxiliary conditions

auxiliary conditions

Initial Conditions

- all conditions are at one point of the independent variable

Boundary Conditions

- The conditions are not at one point of the independent variable

Initial value and Boundary-Value Problems

Initial-Value Problems

- The auxiliary conditions are at one point of the independent variable

$$\ddot{x} + 2\dot{x} + x = e^{-2t}$$

$$x(0) = 1, \quad \dot{x}(0) = 2.5$$

same

Boundary-Value Problems

- The auxiliary conditions are not at one point of the independent variable
- More difficult to solve than initial value problem

$$\ddot{x} + 2\dot{x} + x = e^{-2t}$$

$$x(0) = 1, \quad x(2) = 1.5$$

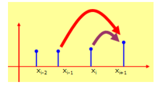
different

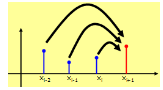
Classification of ODE

ODE can be classified in different ways

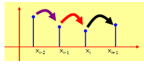
- Order
  - First order ODE
  - Second order ODE
  - N^{th} order ODE
- Linearity
  - Linear ODE
  - Nonlinear ODE
- Auxiliary conditions
  - Initial value problems
  - Boundary value problems

Classification of the Methods





Numerical Methods
for solving ODE



Single-Step Methods

Estimates of the solution at a particular step are entirely based on information on the previous step

Multiple-Step Methods

Estimates of the solution at a particular step are based on information on more than one step

Euler, Runge-Kutta: single step methods & Adam-Moulton method: multi-step method

Taylor Series Method

The problem to be solved is a first order ODE

$$\boxed{\frac{dy(x)}{dx} = f(x,y), \quad y(x_0) = y_0}$$

Estimates of the solution at different base points

$$y(x_0+h), \quad y(x_0+2h), \quad y(x_0+3h), \quad ....$$

are computed using truncated Taylor series expansions

Taylor Series Expansion

Truncated Taylor Series Expansion

$$y(x_0+h) \approx \sum_{k=0}^{n} \frac{h^k}{k!}\left(\left.\frac{d^k y}{dx^k}\right|_{x=x_0,\, y=y_0}\right)$$

$$\approx y(x_0) + h\left.\frac{dy}{dx}\right|_{\substack{x=x_0,\\ y=y_0}} + \frac{h^2}{2!}\left.\frac{d^2 y}{dx^2}\right|_{\substack{x=x_0,\\ y=y_0}} + \ldots + \frac{h^n}{n!}\left.\frac{d^n y}{dx^n}\right|_{\substack{x=x_0,\\ y=y_0}}$$

n^{th} order Taylor series method uses n^{th} order Truncated Taylor series expansion

Euler Method

- First order Taylor series method is known as Euler Method
- Only the constant term and linear term are used in Euler method.
- The error due to the use of the truncated Taylor series is of order $O(h^2)$.

Leonhard Euler
(Basel, April 15, 1707 – September 18, 1783)

- $y = f(x)$

$$\frac{\partial \rho}{\partial t} + \nabla(\rho u) = 0$$

First Order Taylor Series Method ... (Euler Method)

$$y(x_0 + h) = y(x_0) + h \left.\frac{dy}{dx}\right|_{\substack{x=x_0,\\ y=y_0}} + o(h^2)$$

Notation:

$$x_n = x_0 + nh, \qquad y_n = y(x_n),$$

$$\left.\frac{dy}{dx}\right|_{\substack{x=x_i,\\ y=y_i}} = f(x_i, y_i)$$

Euler Method

$$y_{i+1} = y_i + h\ f(x_i, y_i)$$

Euler Method

Problem:

Given the first order ODE $\quad \dot{y}(x) = f(x, y)$

with the initial condition $\quad y_0 = y(x_0)$

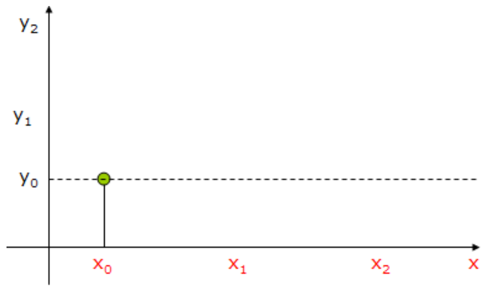
Determine $\quad y_i = y(x_0 + ih) \qquad for\ i = 1,2,...$

Euler Method:

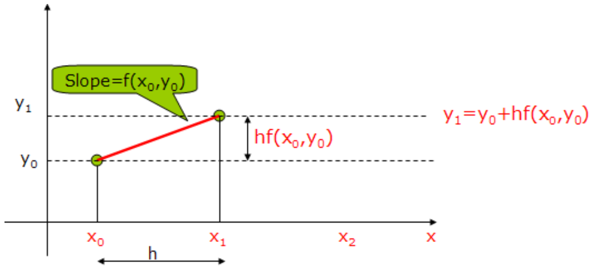
$$y_0 = y(x_0)$$

$$y_{i+1} = y_i + h\ f(x_i, y_i) \qquad for\ i = 1,2,...$$

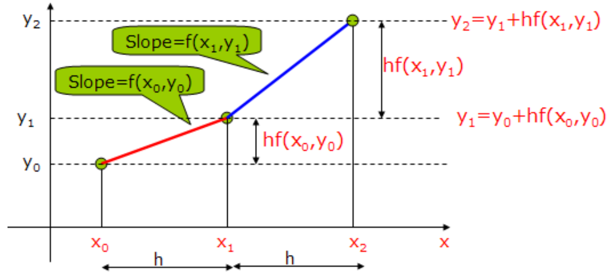
Interpretation of Euler Method



Interpretation of Euler Method



Interpretation of Euler Method



Example 1

Use Euler method to solve the ODE

$$\frac{dy}{dx} = 1 + x^2, \qquad y(1) = -4$$

to determine y(1.01), y(1.02) and y(1.03)

Example 1

$$\boxed{f(x,y) = 1 + x^2, \quad x_0 = 1, \quad y_0 = -4, \quad h = 0.01}$$

Euler Method

$$y_{i+1} = y_i + h\, f(x_i, y_i)$$

$$Step\ 1: \quad y_1 = y_0 + h\, f(x_0, y_0) = -4 + 0.01\,(1 + (1)^2) = -3.98$$

$$Step\ 2: \quad y_2 = y_1 + h\, f(x_1, y_1) = -3.98 + 0.01\left(1 + (1.01)^2\right) = -3.9598$$

$$Step\ 3: \quad y_3 = y_2 + h\, f(x_2, y_2) = -3.9598 + 0.01\left(1 + (1.02)^2\right) = -3.9394$$

Example 1

$$f(x,y) = 1 + x^2, \quad x_0 = 1, \quad y_0 = -4, \quad h = 0.01$$

Summary of the result

i	xi	yi
0	1.00	-4.00
1	1.01	-3.98
2	1.02	-3.9595
3	1.03	-3.9394

Example 1

$$f(x,y) = 1 + x^2, \quad x_0 = 1, \quad y_0 = -4, \quad h = 0.01$$

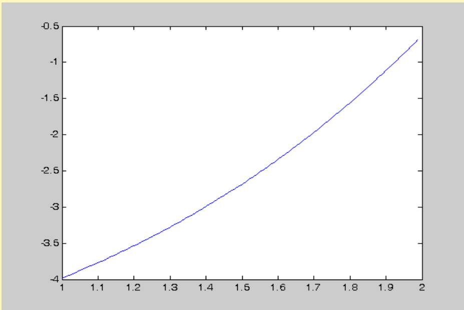
Comparison with true value

i	xi	yi	True value of yi
0	1.00	-4.00	-4.00
1	1.01	-3.98	-3.97996
2	1.02	-3.9595	-3.95959
3	1.03	-3.9394	-3.93091

Example 1

$$f(x,y) = 1 + x^2, \quad x_0 = 1, \quad y_0 = -4, \quad h = 0.01$$

A graph of the solution of the ODE for 1<x<2



Types of Errors

- Local truncation error:

  error due to the use of truncated Taylor series to compute x(t+h) in one step.
- Global Truncation error

  accumulated truncation over many steps
- Round off error:

  error due to finite number of bits used in representation of numbers. This error could be accumulated and magnified in succeeding steps.

Second Order Taylor Series methods

$$\boxed{Given\ \ \frac{dy(x)}{dx} = f(y,x), \quad y(x_0) = y_0}$$

Second order Taylor Series method

$$y_{i+1} = y_i + h\frac{dy}{dx} + \frac{h^2}{2!}\frac{d^2y}{dx^2} + O(h^3)$$

$\frac{d^2y}{dx^2}$ needs to be derived analatically.

Third Order Taylor Series methods

$$\boxed{Given\ \ \frac{dy(x)}{dx} = f(y,x), \quad y(x_0) = y_0}$$

Third order Taylor Series method

$$y_{i+1} = y_i + h\frac{dy}{dx} + \frac{h^2}{2!}\frac{d^2y}{dx^2} + \frac{h^3}{3!}\frac{d^3y}{dx^3} + O(h^4)$$

$\frac{d^2y}{dx^2}$ and $\frac{d^3y}{dx^3}$ need to be derived analatically.

High Order Taylor Series methods

$$\boxed{Given\ \ \frac{dy(x)}{dx} = f(y,x), \quad y(x_0) = y_0}$$

nth order Taylor Series method

$$y_{i+1} = y_i + h\frac{dy}{dx} + \frac{h^2}{2!}\frac{d^2y}{dx^2} + \ldots + \frac{h^n}{n!}\frac{d^ny}{dx^n} + O(h^{n+1})$$

$\frac{d^2y}{dx^2}, \frac{d^3y}{dx^3}, \ldots\ldots, \frac{d^ny}{dx^n}$ need to be derived analatically.

High Order Taylor Series methods

- High order Taylor series methods are more accurate than Euler method
- The 2nd, 3rd and higher order derivatives need to be derived analytically which may not be easy.

Example 2 -Second order Taylor Series Method

Use Second order Taylor Series method to solve

$$\frac{dx}{dt} + 2x^2 + t = 1,\ x(0) = 1, \qquad use \quad h = 0.01$$

$$\boxed{what\ is \ \frac{d^2x(t)}{dt^2}\ ?}$$

Example 2 -Second order Taylor Series Method

Use Second order Taylor Series method to solve

$$\frac{dx}{dt} + 2x^2 + t = 1,\ x(0) = 1, \qquad use \quad h = 0.01$$

$$\frac{dx}{dt} = 1 - 2x^2 - t$$

$$\frac{d^2x(t)}{dt^2} = 0 - 4x\frac{dx}{dt} - 1 = -4x(1 - 2x^2 - t) - 1$$

$$x_{i+1} = x_i + h(1 - 2x_i^{\ 2} - t_i) + \frac{h^2}{2}(-1 - 4x_i(1 - 2x_i^{\ 2} - t_i))$$

Example 2 -Second order Taylor Series Method

$$f(t,x) = 1 - 2x^2 - t, \quad t_0 = 0, \quad x_0 = 1, \quad h = 0.01$$

$$x_{i+1} = x_i + h(1 - 2x_i^{\ 2} - t_i) + \frac{h^2}{2}(-1 - 4x_i(1 - 2x_i^{\ 2} - t_i))$$

Step 1 :

$$x_1 = 1 + .01(1 - 0 - 2(1)^2) + \frac{(0.01)^2}{2}(-1 - 4(1)(-1)) = 0.9901$$

Step 2 :

$$x_2 = 1 + .01(1 - 0.01 - 2(0.9901)^2) + \frac{(0.01)^2}{2}(-1 - 4(0.9901)(-1)) = 0.9807$$

Step 3 :

$x_3 = 0.9716$

Example 2 -Second order Taylor Series Method

$$f(t,x) = 1 - 2x^2 - t, \quad t_0 = 0, \quad x_0 = 1, \quad h = 0.01$$

Summary of the results

i	t_i	x_i
0	0.00	1
1	0.01	0.9901
2	0.02	0.9807
3	0.03	0.9716

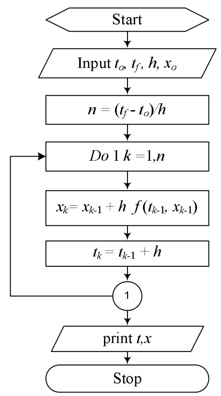
Programming Euler Method

Write a MATLAB program to implement Euler method to solve

$$\frac{dv}{dt} = 1 - 2v^2 - t. \qquad v(0) = 1$$

$$for \quad t_i = 0.01\,i, \qquad i = 1,2,\ldots,100$$

Algorithm_ Euler Method



Programming Euler Method

```
f=inline('1-2*v^2-t','t','v')
h=0.01
t=0
v=1
T(1)=t;
V(1)=v;
for i=1:100
    v=v+h*f(t,v)
    t=t+h;
    T(i+1)=t;
    V(i+1)=v;
end
```

```
f =
     Inline function:
     f(t,v) = 1-2*v^2-t
h =    0.0100
t =    0
v =    1
v =    0.9900
..............
......
v =    0.4725
```

$$\frac{dv}{dt} = 1 - 2v^2 - t. \qquad v(0) = 1$$

$$for \quad t_i = 0.01\,i, \qquad i = 1,2,\ldots,100$$

Programming Euler Method

```
f=inline('1-2*v^2-t','t','v')
h=0.01
t=0
v=1
T(1)=t;
V(1)=v;
for i=1:100
    v=v+h*f(t,v)
    t=t+h;
    T(i+1)=t;
    V(i+1)=v;
end
```

Definition of the ODE

Initial condition

Main loop

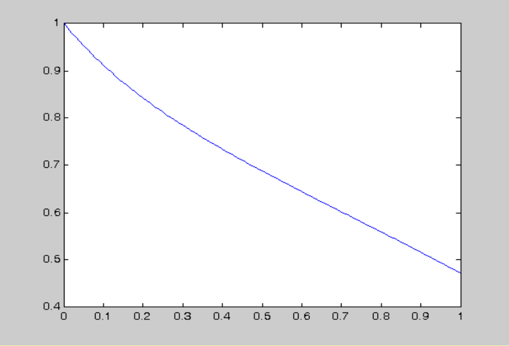
Euler method

Storing information

Programming Euler Method

Plot of the solution

plot(T,V)



Runge-Kutta Method .... Motivation

- Find accurate methods to solve ODE that does not require calculating high order derivatives.
- The approach is to a formula involving unknown coefficients then determine these coefficients to match as many terms of the Taylor series expansion

Runge-Kutta Method

Second Order Runge Kutta

$$K_1 = f(x_i, y_i)$$

$$K_2 = f(x_i + \alpha h, y_i + \beta K_1 h)$$

$$y_{i+1} = y_i + w_1 K_1 + w_2 K_2$$

Problem :

Find α, β, w_1, w_2

such that y_{i+1} is as accurate as possible.

Runge-Kutta Method

Problem: *Find* α, β, w_1, w_2

to match as many terms as possible.

$$\boxed{\begin{aligned} x(t+h) &= x(t) + hx'(t) + \frac{h^2}{2}x''(t) + \frac{h^3}{6}x'''(t) + \ldots \\ x(t+h) &= x(t) + w_1 h\ f(t,x) + w_2 f(t+\alpha h, x+\beta h\ f(t,x)) \end{aligned}}$$

$$f(t+\alpha h, x+\beta h\ f) = f + \alpha h\ f_t + \beta h\ f_x + \frac{1}{2}\left(\alpha h\frac{\partial}{\partial t} + \beta h\frac{\partial}{\partial x}\right)^2 f(\bar{t},\bar{x})$$

$$x(t+h) = x(t) + (w_1 + w_2)h\ f(t,x) + \alpha w_2 h^2 f_t + \beta w_2 h^2 f\ f_t + O(h^3)$$

Runge-Kutta Method

$$x(t+h) = \boxed{x(t)} + \boxed{hx'(t)} + \boxed{\frac{h^2}{2}x''(t)} + \frac{h^3}{6}x'''(t) + \ldots$$

$$x(t+h) = \boxed{x(t)} + \boxed{(w_1 + w_2)h\ f(t,x)} + \boxed{\alpha\, w_2 h^2 f_t + \beta\, w_2 h^2 f\, f_t} + O(h^3)$$

$\Rightarrow w_1 + w_2 = 1, \quad \alpha\, w_2 = 0.5, \quad \beta\, w_2 = 0.5$

One possible solution

$w_1 = 0.5, \quad w_2 = 0.5, \quad \alpha = 1, \quad \beta = 1$

Runge-Kutta Method

Second Order Runge Kutta

$K_1 = h\ f(t,x)$

$K_2 = h\ f(t+h, x+K_1)$

$$x(t+h) = x(t) + \frac{1}{2}(K_1 + K_2)$$

Alternative Formula

Second Order Runge Kutta

$F_1 = f(t,x)$

$F_2 = f(t+h, x+hF_1)$

$$x(t+h) = x(t) + \frac{h}{2}(F_1 + F_2)$$

Runge-Kutta Method … Alternative Formulas

$\Rightarrow w_1 + w_2 = 1, \quad \alpha\, w_2 = 0.5, \quad \beta\, w_2 = 0.5$

another solution

Pick α any non-zero number

$$\beta = \alpha, \quad w_1 = 1 - \frac{1}{2\alpha}, \quad w_2 = \frac{1}{2\alpha}$$

Second Order Runge Kutta Formulas (select $\alpha \neq 0$)

$K_1 = h\, f(t,x)$

$K_2 = h\ f(t+\alpha h, x+\alpha\ K_1)$

$$x(t+h) = x(t) + \left(1 - \frac{1}{2\alpha}\right)F_1 + \frac{1}{2\alpha}F_2$$

Runge-Kutta Method

Third Order Runge Kutta (RK3)

$K_1 = f(x_i, y_i)$

$$K_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1 h)$$

$$K_3 = f(x_i + \frac{1}{2}h, y_i - K_1 h + 2K_2 h)$$

$$y(x+h) = y(x) + \frac{1}{6}(K_1 + 4K_2 + K_3)$$

Runge-Kutta Method

Fourth Order Runge Kutta

$$K_1 = h\ f(t,x)$$

$$K_2 = h\ f(t+\frac{1}{2}h, x+\frac{1}{2}K_1)$$

$$K_3 = h\ f(t+\frac{1}{2}h, x+\frac{1}{2}K_2)$$

$$K_4 = h\ f(t+h, x+K_3)$$

$$x(t+h) = x(t)+\frac{1}{6}\left(K_1+2K_2+2K_3+K_4\right)$$

Runge-Kutta Method

Fifth Order Runge-Kutta

$$K_1 = f(x_i, y_i)$$

$$K_2 = f(x_i+\frac{1}{4}h, y_i+\frac{1}{4}K_1 h)$$

$$K_3 = f(x_i+\frac{1}{4}h, y_i+\frac{1}{8}K_1 h+\frac{1}{8}K_2 h)$$

$$K_4 = f(x_i+\frac{1}{2}h, y_i-\frac{1}{2}K_2 h+K_3 h)$$

$$K_5 = f(x_i+\frac{3}{4}h, y_i+\frac{3}{16}K_1 h+\frac{9}{16}K_4 h)$$

$$K_6 = f(x_i+h, y_i-\frac{3}{7}K_1 h+\frac{2}{7}K_2 h+\frac{12}{7}K_3 h-\frac{12}{7}K_4 h+\frac{8}{7}K_5 h)$$

$$y_{i+1} = y_i+\frac{h}{90}\left(7K_1+32K_3+12K_4+32K_5+7K_6\right)$$

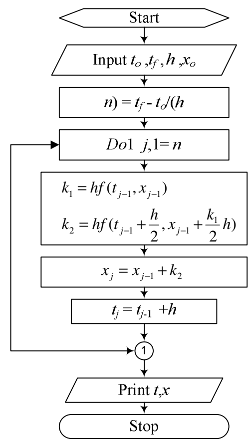
Second order Runge-Kutta Method Example

Solve the following system to find x(1.02) using RK2

$$\dot{x}(t) = 1+x^2(t)+t^3, \qquad x(1) = -4, h = 0.01$$

Runge-Kutta Method

2nd Order algorithm



Second order Runge-Kutta Method
Example

Solve the following system to find x(1.02) using RK2

$$\dot{x}(t) = 1 + x^2(t) + t^3, \qquad x(1) = -4, h = 0.01$$

STEP 1:

$$K_1 = h\ f(t, x) = 0.01(1 + x^2 + t^3) = 0.18$$

$$K_2 = h\ f(t + h, x + K_1) = 0.01(1 + (x + 0.18)^2 + (t + .01)^3) = 0.1692$$

$$x(1 + 0.01) = x(1) + \frac{1}{2}(K_1 + K_2) = -4 + \frac{1}{2}(0.18 + 0.1692) = -3.8254$$

Second order Runge-Kutta Method
Example

STEP 2

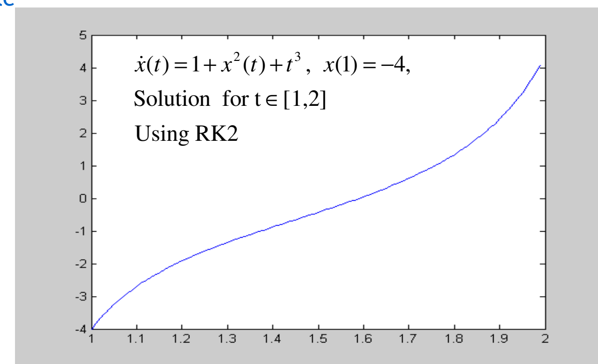
$$K_1 = h\ f(t, x) = 0.01(1 + x^2 + t^3) = 0.1666$$

$$K_2 = h\ f(t + h, x + K_1) = 0.01(1 + (x + 0.1666)^2 + (t + .01)^3) = 0.1545$$

$$x(1.01 + 0.01) = x(1.01) + \frac{1}{2}(K_1 + K_2)$$

$$= -3.8254 + \frac{1}{2}(0.1666 + 0.1545) = -3.6648$$

Second order Runge-Kutta Method
Example



Second order Runge-Kutta Method

Given:

$$\frac{dy}{dx} = f(x, y)$$

$$y(x_0) = y_0$$

h

Determine # of steps needed

Second order Runge-Kutta Method

Given :

$$\frac{dy}{dx} = f(x, y)$$

$$y(x_0) = y_0$$

$$h$$

Determine # of steps needed

RK 2 formula

$$K_1 = f(x_i, y_i)$$

$$K_2 = f(x_i + h, y_i + K_1 h)$$

$$y_{i+1} = y_i + \frac{h}{2}(K_1 + K_2)$$

Second order Runge-Kutta Method

Given :

$$\frac{dy}{dx} = f(x, y)$$

$$y(x_0) = y_0$$

$$h$$

Determine # of steps needed

RK 2 formula

$$K_1 = f(x_i, y_i)$$

$$K_2 = f(x_i + h, y_i + K_1 h)$$

$$y_{i+1} = y_i + \frac{h}{2}(K_1 + K_2)$$

Step 1 :

$$K_1 = f(x_0, y_0)$$

$$K_2 = f(x_0 + h, y_0 + K_1 h)$$

$$y_1 = y_0 + \frac{h}{2}(K_1 + K_2)$$

Step 2 : $x_1 = x_0 + h$

$$K_1 = f(x_1, y_1)$$

$$K_2 = f(x_1 + h, y_1 + K_1 h)$$

$$y_2 = y_1 + \frac{h}{2}(K_1 + K_2)$$

Second order Runge-Kutta Method … example

Use the second order Runge Kutta method to solve the differential equation

$$\frac{dy}{dx} = 1 + y^2 + x^3, \qquad y(1) = -4$$

to find $y(1.01), y(1.02)$

Second order Runge-Kutta Method … example

Problem :

$$\frac{dy}{dx} = 1 + y^2 + x^3, \qquad y(1) = -4$$

Use RK 2 to find $y(1.01), y(1.02)$

$$h = 0.01$$

$$f(x, y) = 1 + y^2 + x^3$$

$$x_0 = 1, \quad y_0 = -4$$

Second order Runge-Kutta Method … example

Problem :

$$\frac{dy}{dx} = 1 + y^2 + x^3, \qquad y(1) = -4$$

$Use \;\; RK2 \;\; to \;\; find \;\; y(1.01), y(1.02)$

h = 0.01

$$f(x, y) = 1 + y^2 + x^3$$

$$x_0 = 1, \quad y_0 = -4$$

Step 1 :

$$K_1 = f(x_0, y_0) = (1 + y_0^2 + x_0^3) = 18.0$$

$$K_2 = f(x_0 + h, y_0 + K_1 h) = (1 + (y_0 + 0.18)^2 + (x_0 + .01)^3) = 16.92$$

$$y_1 = y_0 + \frac{h}{2}(K_1 + K_2) = -4 + \frac{0.01}{2}(18 + 16.92) = -3.8254$$

Second order Runge-Kutta Method … example

Problem :

$$\frac{dy}{dx} = 1 + y^2 + x^3, \qquad y(1) = -4$$

$Use \;\; RK2 \;\; to \;\; find \;\; y(1.01), y(1.02)$

h = 0.01

$$f(x, y) = 1 + y^2 + x^3$$

$$x_1 = 1.01, \quad y_1 = -3.8254$$

Second order Runge-Kutta Method … example

Problem :

$$\frac{dy}{dx} = 1 + y^2 + x^3, \qquad y(1) = -4$$

$Use \;\; RK2 \;\; to \;\; find \;\; y(1.01), y(1.02)$

h = 0.01

$$f(x, y) = 1 + y^2 + x^3$$

$$x_1 = 1.01, \quad y_1 = -3.8254$$

Step 2 :

$$K_1 = f(x_1, y_1) = (1 + y_1^2 + x_1^3) = 16.66$$

$$K_2 = f(x_1 + h, y_1 + K_1 h) = (1 + (y_1 + 0.1666)^2 + (x_1 + .01)^3) = 15.45$$

$$y_2 = y_1 + \frac{h}{2}(K_1 + K_2) = -3.8254 + \frac{0.01}{2}(16.66 + 15.45) = -3.6648$$

Second order Runge-Kutta Method … example

Problem :

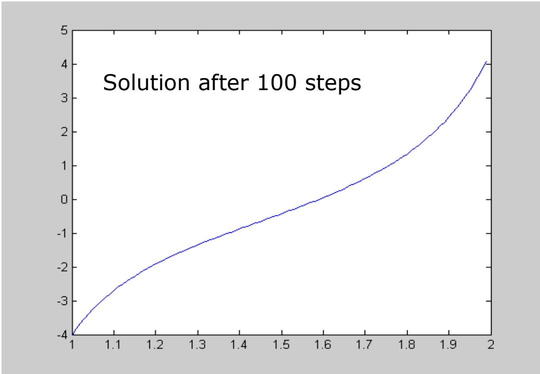
$$\frac{dy}{dx} = 1 + y^2 + x^3, \qquad y(1) = -4$$

$Use \;\; RK2 \;\; to \;\; find \;\; y(1.01), y(1.02)$

Summary of the solution

i	x_i	y_i
0	1.00	−4.0000
1	1.01	−3.8254
2	1.02	−3.6648

Second order Runge-Kutta Method … example



Fourth order Runge-Kutta Method … example

$$\frac{dy}{dx} = 1 + y + x^2$$

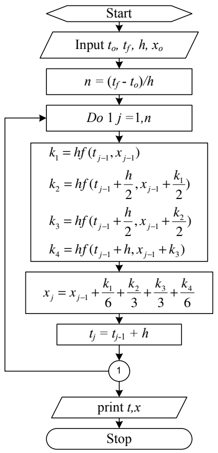
$$y(0) = 0.5$$

$$h = 0.2$$

$$Use \quad RK\ 4\ to\ compute \quad y(0.2)\ \ and \quad y(0.4)$$

Runge-Kutta Method

4th Order algorithm



Fourth order Runge-Kutta Method … example

Problem :

$$\frac{dy}{dx} = 1 + y + x^2, \qquad y(0) = 0.5$$

$$Use\ \ RK\,4\ \ to\ \ find\ \ y(0.2),\ y(0.4)$$

Remember - RK4

Fourth Order Runge Kutta (RK4)

$$K_1 = f(x_i, y_i)$$

$$K_2 = f(x_i + \frac{1}{2}h,\ y_i + \frac{1}{2}K_1 h)$$

$$K_3 = f(x_i + \frac{1}{2}h,\ y_i + \frac{1}{2}K_2 h)$$

$$K_4 = f(x_i + h,\ y_i + K_3 h)$$

$$y_{i+1} = y_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

Fourth order Runge-Kutta Method … example

Problem :

$$\frac{dy}{dx} = 1 + y + x^2, \qquad y(0) = 0.5$$

$$Use\ \ RK4\ \ to\ \ find\ \ y(0.2),\ y(0.4)$$

h = 0.2

$$f(x, y) = 1 + y + x^2$$

$$x_0 = 0, \quad y_0 = 0.5$$

Step 1 :

$$K_1 = f(x_0, y_0) = 1 + y_0 + x_0^{\,2} = 1.5$$

$$K_2 = f(x_0 + \frac{1}{2}h,\ y_0 + \frac{1}{2}K_1 h) = 1 + (y_0 + 0.15) + (x_0 + 0.1)^2 = 1.64$$

$$K_3 = f(x_0 + \frac{1}{2}h,\ y_0 + \frac{1}{2}K_2 h) = 1 + (y_0 + 0.164) + (x_0 + 0.1)^2 = 1.654$$

$$K_4 = f(x_0 + h,\ y_0 + K_3 h) = 1 + (y_0 + 0.16545) + (x_0 + 0.2)^2 = 1.7908$$

$$y_1 = y_0 + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) = 0.8293$$

Fourth order Runge-Kutta Method … example

Problem :

$$\frac{dy}{dx} = 1 + y + x^2, \qquad y(0) = 0.5$$

$$Use\ \ RK4\ \ to\ \ find\ \ y(0.2),\ y(0.4)$$

h = 0.2

$$f(x, y) = 1 + y + x^2$$

$$x_0 = 0.2, \quad y_0 = 0.8293$$

Step 2 :

$$K_1 = f(x_1, y_1) =$$

$$K_2 = f(x_1 + \frac{1}{2}h,\ y_1 + \frac{1}{2}K_1 h) =$$

$$K_3 = f(x_1 + \frac{1}{2}h,\ y_1 + \frac{1}{2}K_2 h) =$$

$$K_4 = f(x_1 + h,\ y_1 + K_3 h) =$$

$$y_2 = y_1 + \frac{0.2}{6}(K_1 + 2K_2 + 2K_3 + K_4) =$$

Fourth order Runge-Kutta Method … example

Problem :

$$\frac{dy}{dx} = 1 + y + x^2, \qquad y(0) = 0.5$$

$$Use\ \ RK4\ \ to\ \ find\ \ y(0.2),\ y(0.4)$$

h = 0.2

$$f(x, y) = 1 + y + x^2$$

$$x_0 = 0.2, \quad y_0 = 0.8293$$

Step 2 :

$$K_1 = f(x_1, y_1) = 1.7893$$

$$K_2 = f(x_1 + \frac{1}{2}h,\ y_1 + \frac{1}{2}K_1 h) = 1.9182$$

$$K_3 = f(x_1 + \frac{1}{2}h,\ y_1 + \frac{1}{2}K_2 h) = 1.9311$$

$$K_4 = f(x_1 + h,\ y_1 + K_3 h) = 2.0555$$

$$y_2 = y_1 + \frac{0.2}{6}(K_1 + 2K_2 + 2K_3 + K_4) = 1.2141$$

Fourth order Runge-Kutta Method … example

Problem :

$$\frac{dy}{dx} = 1 + y + x^2, \qquad y(0) = 0.5$$

$$Use\ \ RK4\ \ to\ \ find\ \ y(0.2),\ y(0.4)$$

Summary of the solution

i	x_i	y_i
0	0.0	0.5
1	0.2	0.8293
2	0.4	1.2141

Runge-Kutta Method … Summary

- Runge Kutta methods generate accurate solution without the need to calculate high order derivatives.
- Second order RK have local truncation error of order $O(h^3)$
- Fourth order RK have local truncation error of order $O(h^5)$
- N function evaluations are needed in N th order RK method.

Computer-based solutions

- IVPAG subroutine (Fortran IMSL subroutine that Solves an initial-value problem for ordinary differential equations using either Adams-Moulton's or Gear's BDF method)
- MATLAB solution of initial value problems for ordinary differential equations (ode23, ode45, ode113, ode15s, ode23s, ode23t, ode23tb))

Computer-based solutions

Solve an initial-value problem for ordinary differential equations using either Adams-Moulton's or Gear's BDF method.

Usage

CALL IVPAG (IDO, N, FCN, FCNJ, A, T, TEND, TOL, PARAM, Y)

Arguments

IDO — Flag indicating the state of the computation. (Input/Output)

IDO	State
1	Initial entry
2	Normal re-entry
3	Final call to release workspace
4	Return because of interrupt 1
5	Return because of interrupt 2 with step accepted
6	Return because of interrupt 2 with step rejected
7	Return for new value of matrix A.

Computer-based solutions

N — Number of differential equations. (Input)

FCN — User-supplied SUBROUTINE to evaluate functions. The usage is CALL FCN (N, T, Y, YPRIME), where

N — Number of equations. (Input)
T — Independent variable, *t*. (Input)
Y — Array of size N containing the dependent variable values, *y*. (Input)
YPRIME — Array of size N containing the values of the vector y' evaluated at (t, y). (Output)
See Comment 3.

FCN must be declared EXTERNAL in the calling program.

FCNJ — User-supplied SUBROUTINE to compute the Jacobian. The usage is CALL FCNJ (N, T, Y, DYPDY) where

N — Number of equations. (Input)
T — Independent variable, *t*. (Input)
Y — Array of size N containing the dependent variable values, $y(t)$. (Input)

DYPDY — An array, with data structure and type determined by PARAM(14) = MTYPE, containing the required partial derivatives $\partial fi/\partial yj$. (Output)

Computer-based solutions

Example 1

Euler's equation for the motion of a rigid body not subject to external forces is

$$y_1' = y_2 y_3 \qquad y_1(0) = 0$$
$$y_2' = -y_1 y_3 \qquad y_2(0) = 1$$
$$y_3' = -0.51 y_1 y_2 \qquad y_3(0) = 1$$

```
      INTEGER    N, NPARAM
      PARAMETER  (N=3, NPARAM=50)
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER    IDO, IEND, NOUT
      REAL       A(1,1), PARAM(NPARAM), T, TEND, TOL, Y(N)
C                                  SPECIFICATIONS FOR SUBROUTINES
      EXTERNAL   IVPAG, SSET, UMACH
C                                  SPECIFICATIONS FOR FUNCTIONS
      EXTERNAL   FCN, FCNJ
C                                  Initialize
      CALL SSET (NPARAM, 0.0, PARAM, 1)
C
```


Computer-based solutions

```
      IDO  = 1
      T    = 0.0
      Y(1) = 0.0
      Y(2) = 1.0
      Y(3) = 1.0
      TOL  = 1.0E-6
C                                  Write title
      CALL UMACH (2, NOUT)
      WRITE (NOUT,99998)
C                                  Integrate ODE
      IEND = 0
   10 CONTINUE
      IEND = IEND + 1
      TEND = IEND
C                                  The array a(*,*) is not used.
      CALL IVPAG (IDO, N, FCN, FCNJ, A, T, TEND, TOL, PARAM, Y)
      IF (IEND .LE. 10) THEN
         WRITE (NOUT,99999) T, Y
C                                  Finish up
         IF (IEND .EQ. 10) IDO = 3
         GO TO 10
      END IF
99998 FORMAT (11X, 'T', 14X, 'Y(1)', 11X, 'Y(2)', 11X, 'Y(3)')
99999 FORMAT (4F15.5)
      END
C
```


Computer-based solutions

```
      SUBROUTINE FCN (N, X, Y, YPRIME)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER    N
      REAL       X, Y(N), YPRIME(N)
      YPRIME(1) = Y(2)*Y(3)
      YPRIME(2) = -Y(1)*Y(3)
      YPRIME(3) = -0.51*Y(1)*Y(2)
      RETURN
      END
C
      SUBROUTINE FCNJ (N, X, Y, DYPDY)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER    N
      REAL       X, Y(N), DYPDY(N,*)
C                                  This subroutine is never called
      RETURN
      END
```


Computer-based solutions

Output

T	Y(1)	Y(2)	Y(3)
1.00000	0.80220	0.59705	0.81963
2.00000	0.99537	-0.09615	0.70336
3.00000	0.64141	-0.76720	0.88892
4.00000	-0.26961	-0.96296	0.98129
5.00000	-0.91173	-0.41079	0.75899
6.00000	-0.95751	0.28841	0.72967
7.00000	-0.42877	0.90342	0.95197
8.00000	0.51092	0.85963	0.93106
9.00000	0.97567	0.21926	0.71730
10.00000	0.87790	-0.47884	0.77906

Look for IMSL document (IVPAG.pdf) for detailed description and other examples

See "Example of using UVPAG subroutine"

Computer-based solutions

- MATLAB solution of initial value problems for ordinary differential equations (ode23, ode45, ode113, ode15s, ode23s, ode23t, ode23tb))

ode23, ode45, ode113, ode15s, ode23s, ode23t, ode23tb - Solve initial value problems for ordinary differential equations

Syntax

[T,Y] = *solver*(odefun,tspan,y0)
[T,Y] = *solver*(odefun,tspan,y0,options)
[T,Y,TE,YE,IE] = *solver*(odefun,tspan,y0,options)
sol = *solver*(odefun,[t0 tf],y0...)

Arguments …..

For detailed description, see the site:

http://www.mathworks.com/access/helpdesk/help/techdoc/ref/ode23.html

Computer-based solutions

Example 1

An example of a nonstiff system is the system of equations describing the m forces.

$$y'_1 = y_2\, y_3 \qquad y_1(0) = 0$$
$$y'_2 = -y_1\, y_3 \qquad y_2(0) = 1$$
$$y'_3 = -0.51\, y_1\, y_2 \qquad y_3(0) = 1$$

To simulate this system, create a function rigid containing the equations

```
function dy = rigid(t,y)
dy = zeros(3,1);    % a column vector
dy(1) = y(2) * y(3);
dy(2) = -y(1) * y(3);
dy(3) = -0.51 * y(1) * y(2);
```

In this example we change the error tolerances using the odeset command initial condition vector [0 1 1] at time 0.

```
options = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5]);
[T,Y] = ode45(@rigid,[0 12],[0 1 1],options);
```

Plotting the columns of the returned array Y versus T shows the solution

```
plot(T,Y(:,1),'-',T,Y(:,2),'-.',T,Y(:,3),'.')
```