SPATIO-TEMPORAL SIMULATION MODELLING: POINT MODELS AND DIFFERENTIAL EQUATIONS

03 Runge-Kutta

Derek Karssenberg

1

Classical Runge-Kutta method of 4th order

- calculate four auxilliarry variables
- derive new value from these
- small numerical error, easy to program

2

Classical Runge-Kutta method of 4th order

We have the initial value problem

$$\frac{dy}{dt} = f(y,t), \qquad y(t_0) = c$$

The solution is

$$k_1 = \Delta t \cdot f\big(y(t),t\big)$$

$$k_2 = \Delta t \cdot f\big(y(t) + \tfrac{1}{2}k_1, t\big)$$

$$k_3 = \Delta t \cdot f\big(y(t) + \tfrac{1}{2}k_2, t\big)$$

$$k_4 = \Delta t \cdot f\big(y(t) + k_3, t\big)$$

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

3

Classical Runge-Kutta method of 4th order

We have the initial value problem

$$\frac{dy}{dt} = f(y,t), \qquad y(t_0) = c$$

The solution is

$$k_1 = \Delta t \cdot f\big(y(t),t\big)$$

$$k_2 = \Delta t \cdot f\big(y(t) + \tfrac{1}{2}k_1, t\big)$$

$$k_3 = \Delta t \cdot f\big(y(t) + \tfrac{1}{2}k_2, t\big)$$

$$k_4 = \Delta t \cdot f\big(y(t) + k_3, t\big)$$

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

Exercise:

$$\frac{dy}{dt} = ky, \qquad y(t_0) = y_i$$

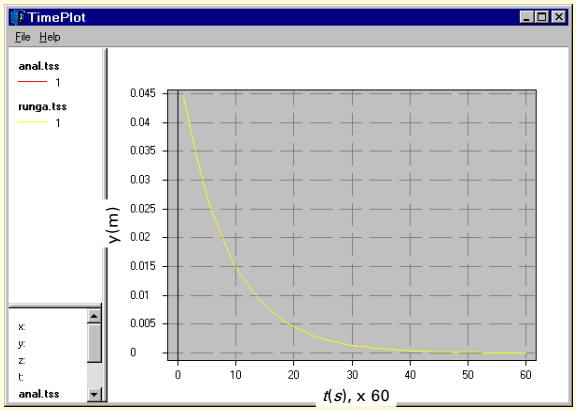
$y(t_0) = 0.02$
$k = -0.002$
Calculate $y(t + 60)$
Use a time step of 60

4

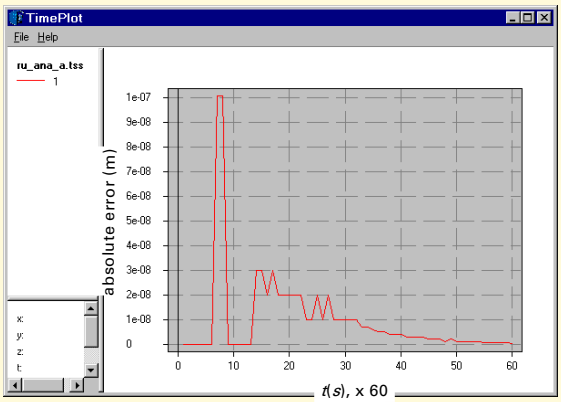
Runge-Kutta method, example

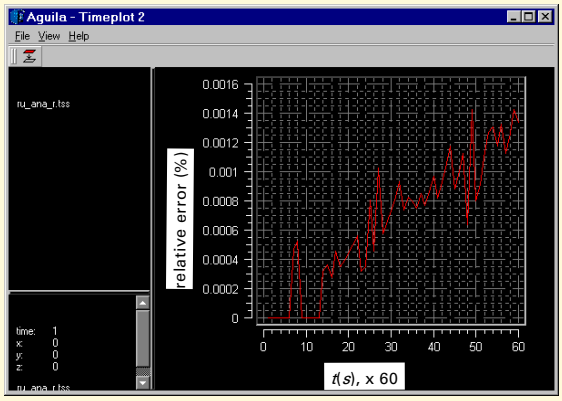
```
k = -0.002;            # fraction output from canopy (s-1)
Dt = 60;               # timestep (s)

y = 0.05
t = 0

while t < 60
    KOne = Dt * (k*y)
    KTwo = Dt * (k*(y + 0.5*KOne))
    KThree = Dt *(k*(y + 0.5*KTwo))
    KFour = Dt *(k*(y + KThree))
    y = y + (1.0/6.0)*(Kone + 2.0 * KTwo + 2.0 * KThree + KFour)
    print(y)
    t = t + 1
```

5



6





Some conclusions, final remarks

- in many cases, Euler method can be used (and is used)
- when precision is important, use Runge-Kutta

Not all diff. equations can be solved with Runge-Kutta!

- For complicated problems, use pre-programmed software
  Example: MODFLOW

Literature: Kreyszig

Kreyszig (1999)	*sheets*
h	Δt
x	t
y_{n+1}	$y(t+\Delta t)$

You could read:

- automatic step size selection (p 945)
- proof of local error (p 947)
- error and step size control (p 949)