RUKASECOND METHOD

RK_4_2 Step of 4th order Runge-Kutta algorithm for second order ODE

We need to implement two linked Runge-Kutta algorithms for

$$q' = i \qquad equation\ 1$$

$$ai' + bi + cq = d \qquad equation\ 2$$

which were obtained from the second order ODE:

$$aq'' + bq' + cq = d$$

The function "func" is obtained from 2) by rearranging:

$$i' = (1/a) * (d - bi - cq)$$

and replacing a, b, c and d with L, v_in, R and C respectively.

In our code we used x = q and y = i, so that the algorithm can be used generically.

The coefficients for the classic 4th order Runge-Kutta algorithm are:

$$k_{x1}\ ,\ k_{x2}\ ,\ k_{x3}\ \text{and}\ k_{x4} \qquad for\ equation\ 1$$

$$k_{y1}\ ,\ k_{y2}\ ,\ k_{y3}\ \text{and}\ k_{y4} \qquad for\ equation\ 2$$

Note that for optimizing the code, instead of calling the function "func" for 1), we call, since we are solving a second order equation and not any set of two simultaneous first order ODE's. Therefore the function for x' will always be y.

$$x_i = y + h * k_{y(i-1)}$$

```
function [ x_it,y_it ] = rukasecond( x, y, t, h, func)
    k_y1=func(x, y, t);
    k_x1=y;
    k_y2=func(x+0.5*h*k_x1, y+0.5*h*k_y1, t+0.5*h);
    k_x2=y+0.5*h*k_y1;
    k_y3=func(x+0.5*h*k_x2, y+0.5*h*k_y2, t+0.5*h);
    k_x3=y+0.5*h*k_y2;
    k_y4=func(x+h*k_x3, y+h*k_y3, t+h);
    k_x4=y+h*k_y3;

    x_it=x+h*(k_x1+2*(k_x2+k_x3)+k_x4)/6;
    y_it=y+h*(k_y1+2*(k_y2+k_y3)+k_y4)/6;


end
```

Published with MATLAB® R2015b