Algorithm 3: Runge-Kutta methods

|  |
| --- |
| Method introduction: |
| 简单叙述这个方法的计算思想，计算公式，适用对象，优缺点等内容。这部分属于偏数学的东西。  （英文书写不要超过1页）  We have the equation , then we integrate from 0 to x on both side of the equation, we can obtain , the Runge-Kutta method’s idea is using the more points to approximate the right integrate, so we can get high order method. Such as fourth order Runge-Kutta method: |
| Algorithm Design |
| 算法设计和实施部分，这部分叙述计算机编程，怎么去实现。  可以是流程图，也可以伪代码，也可以用  step 1 make a partition of the domain , h=(b-x0)/n  step 2 from i=0 to n-1, do  step 3 you will get y(b), and return |
| Matlab code |
| function [ x, y ] = Runge\_Kutta( fun, x0, xt, y0, h )  %UNTITLED 此处显示有关此函数的摘要  % 此处显示详细说明  x = (x0: h: xt);  y = zeros(size(x));  y(1) = y0;  for k = 1: size(x,2)-1  k1 = feval(fun, x(k), y(k));  k2 = feval(fun, x(k)+0.5\*h, y(k)+0.5\*h\*k1);  k3 = feval(fun, x(k)+0.5\*h, y(k)+0.5\*h\*k2);  k4 = feval(fun, x(k)+h, y(k)+h\*k3);  y(k+1) = y(k)+h/6\*(k1+2\*k2+2\*k3+k4);  end  end  clear;  % f = @(x,y) sin(x) + y;  f = @(x,y) -y+1;  x0 = 0;  y0 = 0;  h = 0.01;  xt = 2;  [x, y] = Runge\_Kutta(f, x0, xt, y0, h );  % yy = dsolve('Dy = y + sin(t)','y(0) = 0'); %符号解  % y1 = subs(yy, 't', x);  yt = 1-exp(-x);  disp([x', y', yt']);  plot(x, y, '\*b', x, yt, 'og') |
| Examples and Result |
| 此处需要展示你的程序对课本中的例题习题的计算结果  最好是书上的例题，这样结果正确能保证程序正确，也可以和其它的算法进行结果比较。  要求：迭代列出初值，前5次结果和最后3次结果，中间用省略号。一个方法可以展示一个例子，最多展示3个同类例子。  The equation is 'Dy = -y + 1','y(0) = 0'  X 数值解 精确解  0 0 0  0.1000 0.0952 0.0952  0.2000 0.1813 0.1813  0.3000 0.2592 0.2592  0.4000 0.3297 0.3297  0.5000 0.3935 0.3935  0.6000 0.4512 0.4512  0.7000 0.5034 0.5034  0.8000 0.5507 0.5507  0.9000 0.5934 0.5934  1.0000 0.6321 0.6321  Remarks |
| 此处写该方法程序设计的一些注意事项，也可以空白 |
|  |