

```

% Vector version of RK4 for systems of 2 differential equations
% y1'=f1(t,y1,y2)  y1(a)=y10
% y2'=f2(t,y1,y2)  y2(a)=y20
% Input: [a,b] - interval of t, m-number of subintervals
% ya=(y10,y20)' - initial value
% fi(t,y1,...,yn) - right side functions (n=1,2)
% Output: y(m,2): y(m,i) - approximation for yi(t) function. (i=1,2)
% t - time steps
function [t y]=rk4vec(a,b,m,ya,f1,f2)
h=(b-a)/m;
y=zeros(m+1,2); y(1,:)=ya; t(1)=a;
for k=2:m+1;
    t(k)=t(k-1)+h; s11=0;s12=0;s21=0;s22=0;s31=0;s32=0;s41=0;s42=0;
%Iteration
    s11=f1(t(k-1),y(k-1,1),y(k-1,2));
    s12=f2(t(k-1),y(k-1,1),y(k-1,2));
    s21=f1(t(k-1)+h/2,y(k-1,1)+(h/2)*s11,y(k-1,2)+(h/2)*s12);
    s22=f2(t(k-1)+h/2,y(k-1,1)+(h/2)*s11,y(k-1,2)+(h/2)*s12);
    s31=f1(t(k-1)+h/2,y(k-1,1)+(h/2)*s21,y(k-1,2)+(h/2)*s22);
    s32=f2(t(k-1)+h/2,y(k-1,1)+(h/2)*s21,y(k-1,2)+(h/2)*s22);
    s41=f1(t(k-1)+h,y(k-1,1)+h*s31,y(k-1,2)+h*s32);
    s42=f2(t(k-1)+h,y(k-1,1)+h*s31,y(k-1,2)+h*s32);
    y(k,1)=y(k-1,1)+(h/6)*(s11+2*s21+2*s31+s41);
    y(k,2)=y(k-1,2)+(h/6)*(s12+2*s22+2*s32+s42);
end

```

```

function dydt = ex1(t,y)
dydt = [y(2)^2-2*y(1); y(1)-y(2)-t*y(2)^2 ];

```

```

%Example 6.13 Text book. Using Runge-Kutta 4 method.

```

```

clear;
a=0;b=1; m=100;
ya=[0 1];
f1=inline('z^2-2*x','t','x','z');
f2=inline('x-z-t*z^2','t','x','z');
[t y]=rk4vec(a,b,m,ya,f1,f2);
subplot(2,1,1);plot(t,y(:,1),'r-',t,y(:,2),'--')
title('Runge-Kutta of order 4 approximation')
xlabel('time t');
ylabel('solution y');
grid
legend('y_1(t)', 'y_2(t)')

```

```

%Using the MatLab ode45 command
[t1,y1] = ode45(@ex1,[0 1],[0; 1]);
subplot(2,1,2); plot(t1,y1(:,1),'r-',t1,y1(:,2),'--')
grid
title('Matlab Solution');
xlabel('time t');
ylabel('solution y');
legend('y_1(t)', 'y_2(t)')

```

