1. 代码：

1、代码

clear;clc;

h = 0.1;

x = 0 : h : 20;

N = length(x);

x1p=zeros(1,N);

x2p=zeros(1,N);

x1=zeros(1,N);

x2=zeros(1,N);

x1(1)=0;

x2(1)=1;

func\_1=@(x2)x2;

func\_2=@(x\_1,x\_2) -2\*x\_1-x\_2+5;

for i =2:N

x1\_p=x1(i-1)+h\*func\_1(x2(i-1));

x2\_p=x2(i-1)+h\*func\_2(x1(i-1),x2(i-1));

string\_ola=['欧拉法',num2str(x1\_p),' ',num2str(x2\_p)];

disp(string\_ola);

x1(i)=x1(i-1)+h/2\*(func\_1(x2(i-1))+func\_1(x2\_p));

x2(i)=x2(i-1)+h/2\*(func\_2(x1(i-1),x2(i-1))+func\_2(x1\_p,x2\_p));

string=['梯形法',num2str(x1(i)),' ',num2str(x2(i))];

disp(string);

end

plot(x, x1, 'o', x, x2, '\*');

legend('位移', '速度');

2、每步输出：

图形用户界面, 应用程序

描述已自动生成

3、代码结果

图形用户界面, 图表, 折线图

描述已自动生成

1. Simulink
2. 模型图

图示, 示意图

描述已自动生成

1. 绘图结果

位移

图形用户界面

中度可信度描述已自动生成

速度

电脑萤幕画面

描述已自动生成

1. 结果对比分析

代码中，梯形法选择的步长为0.01，总共的时间为20s，但是并没有控制精度。Simulink法的步长与时间设置与代码中设置相同，但是将精度控制1e-6之内，所以看起来更为平滑。但是二者的结果基本相同。