

算法结构

**1.欧拉法:**

对自变量取值范围进行离散化

微分方程等式左边用一阶差商代替导数

对微分方程做等价变换

**2.改进欧拉法**

对自变量取值范围进行离散化

微分方程等式左边用一阶差商代替导数等式右边直接进行积分

分别利用左、右端点得到加权平均

对微分方程做等价变换

**3.** **Adams外插法**

对自变量取值范围进行离散化

微分方程等式左边用一阶差商代替导数

等式右边构造f(t,u(t))的插值函数进行积分

并求解出插值函数系数

对微分方程做等价变换

**4.** **四阶R-K方法**

对自变量取值范围进行离散化

将u(t)在某一点按Taylor展开,并将u’(t)=f(t,u(t))带入展开式得到u(t+h)和u(t)的一组迭代公式

微分方程等式两边作积分,并将f(t,u(t))用一些结点函数做近似,得到含有节点函数系数的一组迭代公式

两组迭代公式对应位置系数相同,解出节点函数系数

对微分方程做等价变换

代码

**1.欧拉法**

function result = Euler(start,finish,stride,u0)%依次是 起点,终点,步长,初始值

n = (finish - start)/stride;%结点数

t = start;%t(0)

u=[];u(1) = u0;%初始值相同

for i = 1 : n

u(i+1) = u(i) + stride \* t \* u(i)^2;

t = t + stride;

end

result = u;

xlswrite('E:\学习\计算机\微分方程数值解\结课上机\实验项目一\result.xlsx',u,'Sheet1','B2');

end

**2.改进欧拉法**

function result = improved\_Euler(start,finish,stride,u0)%依次是 起点,终点,步长,初始值

%预报矫正的改进的欧拉法

n = (finish - start)/stride;%结点数

u=[];u(1) = u0;%u(1)是实际上的u(0)

t = start;%t(0)

for i = 1 : n

u(i+1) = u(i) + stride/2\*( ...

t \* u(i)^2 ... % f(t,u)= t(i)\*u(i)^2

+...

( t + stride )\*( u(i) + stride\* t \*u(i)^2 )^2 );

%f( t+h,u(i+1) ) = t(i+1)\*(u(i) + h\*f(t,u))^2,预报矫正

t = t + stride;

end

result = u;

xlswrite('E:\学习\计算机\微分方程数值解\结课上机\实验项目一\result.xlsx',u,'Sheet1','B3');

end

**3.** **Adams外插法**

function result = Adams(start,finish,stride,u0 )%依次是 起点,终点,步长.,u(0),此代码为2阶的

n = (finish - start)/stride;%结点数

%题目要求采用2阶的Adams外插法,但是只给了一个结点的初始值,因此还需要一个初始值

%利用改进欧拉法对u1进行预报,即代码中的u(2)

u=[];u(1) = u0;%u(1)是实际上的u(0)

t = start;%t(0)

u(2) = u(1) + stride/2\*( ...

t \* u(1)^2 ... % f(t,u)= t(i)\*u(i)^2

+...

( t + stride )\*( u(1) + stride\* t \*u(1)^2 )^2 );

%f( t+h,u(i+1) ) = t(i+1)\*(u(i) + h\*f(t,u))^2,预报矫正

t = t + stride;

% 从而得到u0 u1两个初始值 分别存储在u(1)u(2)中

for i = 2:n

u(i+1) = u(i) + stride/2\*( 3\*t\*u(i)^2 - (t-stride)\*u(i-1)^2 );

t = t + stride;

end

result = u;

xlswrite('E:\学习\计算机\微分方程数值解\结课上机\实验项目一\result.xlsx',u,'Sheet1','B4');

end

**4.** **四阶R-K方法**

function result = Runge\_Kuuta(start,finish,stride,u0)%依次是 起点,终点,步长.

%题目要求四级四阶方法,对应 m = 4 , f = tu^2

u0 = 1 ; %给定初始值

t = start;%t(0)

n = (finish - start)/stride;%结点数

u=[];u(1) = u0;%u(1)实际的u(0)

for i = 1:n

k1 = t\*u(i)^2;

k2 = (t + stride/2)\*( u(i) + stride/2 \* k1)^2;

k3 = (t + stride/2)\*( u(i) + stride/2 \* k2)^2;

k4 = (t + stride) \*( u(i) + stride \* k3)^2;

u(i+1) = u(i) + stride/6\*( k1 + 2\*k2 + 2\*k3 + k4);

t = t + stride;

end

result = u ;

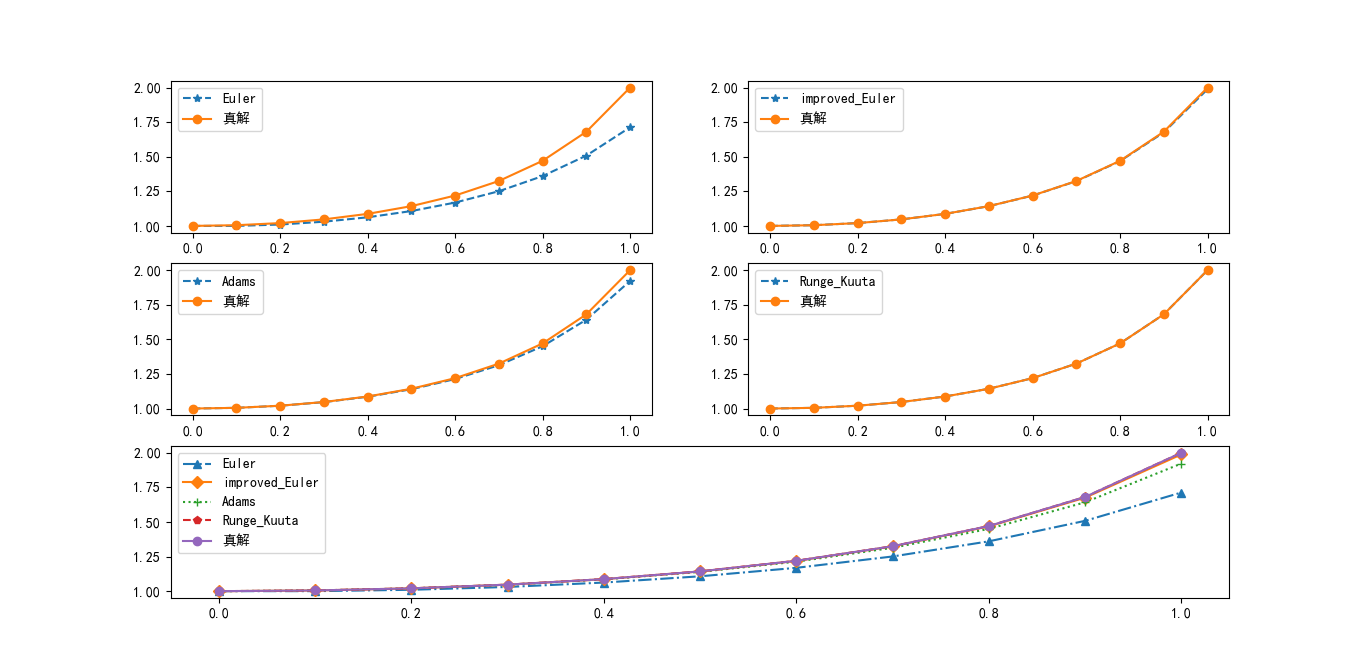
xlswrite('E:\学习\计算机\微分方程数值解\结课上机\实验项目一\result.xlsx',u,'Sheet1','B5');

end

结果

各种方法计算结果下表:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| t | Euler | improved\_Euler | Adams | Runge\_Kuuta | 真解 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 0.1 | 1 | 1.005 | 1.005 | 1.005025136 | 1.00502513 |
| 0.2 | 1.01 | 1.02035441 | 1.020150375 | 1.020408206 | 1.02040816 |
| 0.3 | 1.030402 | 1.047026381 | 1.046321454 | 1.047120522 | 1.04712042 |
| 0.4 | 1.062253848 | 1.086794642 | 1.085179872 | 1.086956728 | 1.08695652 |
| 0.5 | 1.107389178 | 1.142568237 | 1.139414965 | 1.14285752 | 1.14285714 |
| 0.6 | 1.168704718 | 1.218971213 | 1.213232642 | 1.219512844 | 1.2195122 |
| 0.7 | 1.250656961 | 1.323439592 | 1.31324999 | 1.324504352 | 1.32450331 |
| 0.8 | 1.360146959 | 1.46838356 | 1.450177669 | 1.470589653 | 1.47058824 |
| 0.9 | 1.508146939 | 1.675790842 | 1.642177607 | 1.680672908 | 1.68067227 |
| 1 | 1.712852586 | 1.98812569 | 1.922117881 | 1.999991198 | 2 |



作出各种方法与真解的图: