实验报告一.

201718026 徐泉峄

一、理论基础

1、Euler法：

考虑 $u=u(t)$, s.t $\begin{cases}\frac{du(t)}{dt}=f(t,u(t)) \\ u(0)=u_0\end{cases}$ $0\le t\le T$ ★

将连续问题离散化，差商代替导数，将 $[0,T]$ 作 N 等分，$h=T/N$，节点 $0=t_0<t_1<\cdots<t_N=T$，

节点 $t_j=j\cdot h$，设 $u(t_j)\approx u_j$，取相邻节点 t_n, t_{n+1}，$u'(t_n)\approx\frac{u(t_{n+1})-u(t_n)}{t_{n+1}-t_n}=\frac{u(t_{n+1})-u(t_n)}{h}$ ①

又 $t=t_n$，$u'(t_n)=f(t_n,u(t_n))$ ②，①②联立，等价变形有：

$u(t_{n+1})=u(t_n)+h\cdot f[t_n,u(t_n)]$，若以 $u_n\sim u(t_n)$，则有

$$u_{n+1}=u_n+h\cdot f[t_n,u_n].$$

2、改进欧拉法预报校正.

对于★式，考虑 $\int_{t_n}^{t_{n+1}}u'(t)\,dt=u(t_{n+1})-u(t_n)=\int_{t_n}^{t_{n+1}}f(t,u(t))\,dt$ ③

③式右端作数值积分近似，得到左矩形公式，$u(t_{n+1})\approx u(t_n)+hf(t_n,u(t_n))$

以 $u_n\approx u(t_n)$，有 $u_{n+1}=u_n+h\cdot f(t_n,u_n)$，同理可得 $u_{n+1}=u_n+h\cdot f(t_{n+1},u_{n+1})$

作平均：$u_{n+1}=u_n+\frac{h}{2}[f(t_n,u_n)+f(t_n+h,u_{n+1})]$，其中 $f(t_n+h,u_{n+1})$ 中 u_{n+1}

采用Euler法进行预报校正.

3、二阶Adams外插法.

对于★式，等价为 $u(t_{n+1})=u(t_n)+\int_{t_n}^{t_{n+1}}f(t,u(t))$，为提高 $\int_{t_n}^{t_{n+1}}f(t,u(t))$ 精度

在 $[t_n,t_{n+1}]$ 上作 k 次插值多项式，记作 $L_{n,k}(t)$，则 $\int_{t_n}^{t_{n+1}}f(t,u(t))\,dt\approx\int_{t_n}^{t_{n+1}}L_{n,k}(t)\,dt$

对于Adams外插法，求解 $L_{n,k}(t)$ 时，采用等距节点的Newton向后插值公式.

令 $t=t_n+\tau h$，$(t_n\le t\le t_{n+1}, 0\le\tau\le1)$ $\Delta_+f_{n-1}=f_n-f_{n-1}$，$\cdots\Delta_+^j f_{n-j}$ 表示 f_{n-j} 的

j 阶向前差分，则 $L_{n,k}(t)=f_n+\tau\Delta_+f_{n-1}+\cdots+\frac{\tau\cdots(\tau+k-1)}{k!}\Delta_+^k f_{n-k}$.

记 $\binom{s}{j}=\frac{s(s-1)\cdots(s-j+1)}{j!}$，定义 $\binom{s}{0}=1$，从而可推导：

$$u(t_{n+1})\approx u(t_n)+h\sum_{j=0}^{k}a_j\cdot\Delta_+^j f_{n-j}，其中 a_j=\int_0^1(-1)^j\binom{-\tau}{j}d\tau$$

$k=1$ 时阶数为2. $u_{n+1}=h\cdot[\frac{3}{2}f_n-\frac{1}{2}f_{n-1}]$.

Date.　　　Page.

4、Runge_kutta法，

对于方程，对$u(t+h)$在t处作泰勒展开有，

$$u(t+h)=u(t)+\sum_{j=1}^{3} h^j u^{(j)}(t)/j!+o(h^4)=u(t)+h\Phi_1(t,u(t),h)$$

其中$\Phi_1(t,u(t),h)=f+\frac{1}{2}hF+\frac{h^2}{6}(F f_u'+G)+o(h^3)$ ④

$F=f_t'+f\cdot f_u'$，$G=f_{tt}''+2f\cdot f_{tu}''+f^2 f_{uu}''$

另一方面，令$f(t,u(t))=\sum_{i=1}^{3}K_i c_i$，$c_i$、$K_i$待定，不妨取$K_1=f(t,u(t))$

$K_2=f(t_2,u(t_2))$，其中将$u(t_2)$用Euler法近似，则$K_2=f(t+a_2h,u(t)+ha_2K_1)$

再将K_2按二元函数作泰勒展开，得$K_2=f+ha_2F+\frac{h^2a_2^2}{2}G+o(h^3)$

同理K_3的解析式，$K_3=f+ha_3F+h^2(a_2b_{32}f_u'F+\frac{1}{2}a_3^2G)+o(h^3)$

令$\Phi(t,u(t),h)=\sum_{i=1}^{3}c_iK_i=(c_1+c_2+c_3)F+h(a_2c_2+a_3c_3)F+\frac{h^2}{2}[2a_2b_{32}c_3f_u'F+G(a_2^2c_2+a_3^2c_3)]$

与④式对应系数相等，当$m=4$，解得我们Runge-Kutta方法。

二、算法结构

见文档。

三、程序代码

~~源~~见文档。

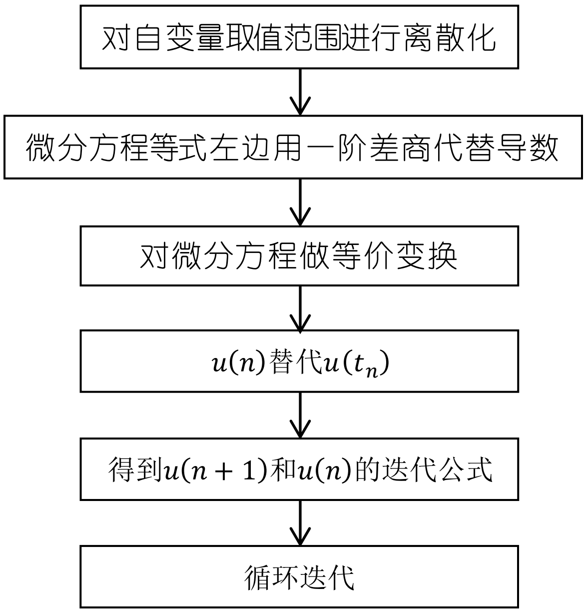
四、实验结果分析

图表见压缩包，通过编程实现与理论基础学习，总结对比各种方法如下：Euler法实现简单方便，但精度较低，可以考虑使用改进Euler法进一步提高精度。对于Adams法以Runge-kutta法，虽然精度高，但理论复杂，计算量大，几种方法应在实际使用中择优选择。

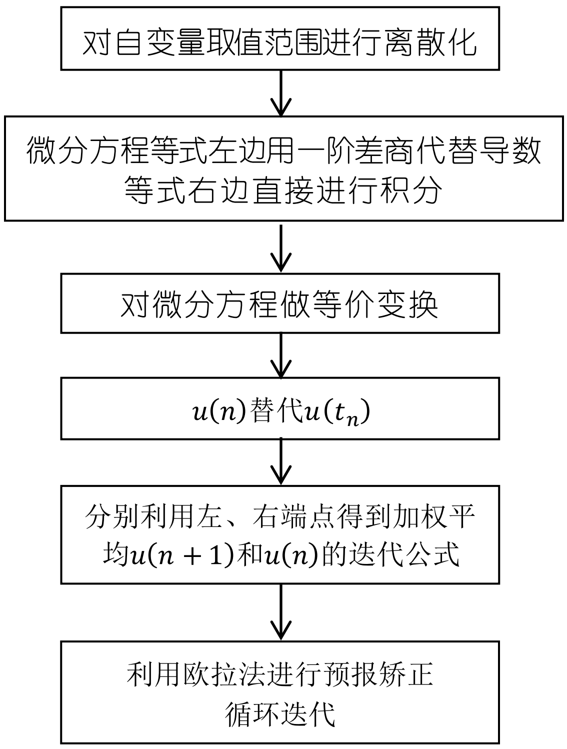
算法结构

考虑微分方程: $\begin{cases} \dfrac{du}{dt} = f(\,t, u(t)\,), 0 \le t \le T \\ u(0) = u_0 \end{cases}$

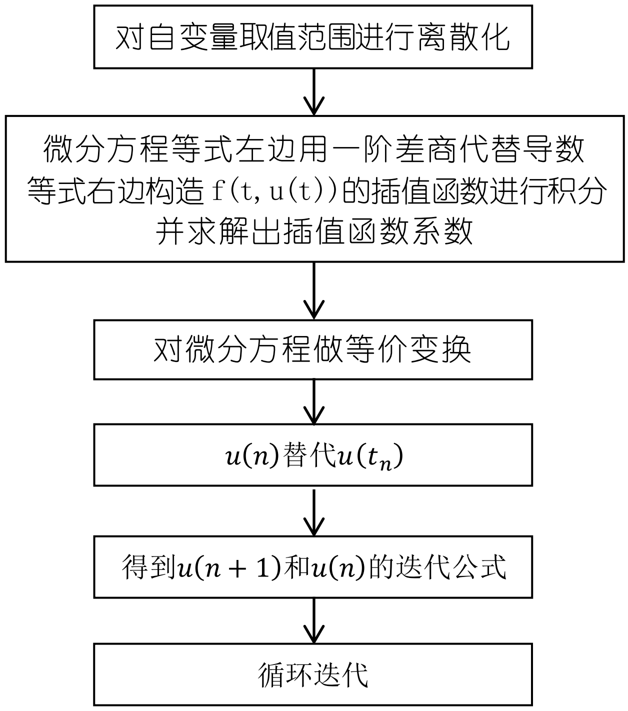
1.欧拉法:



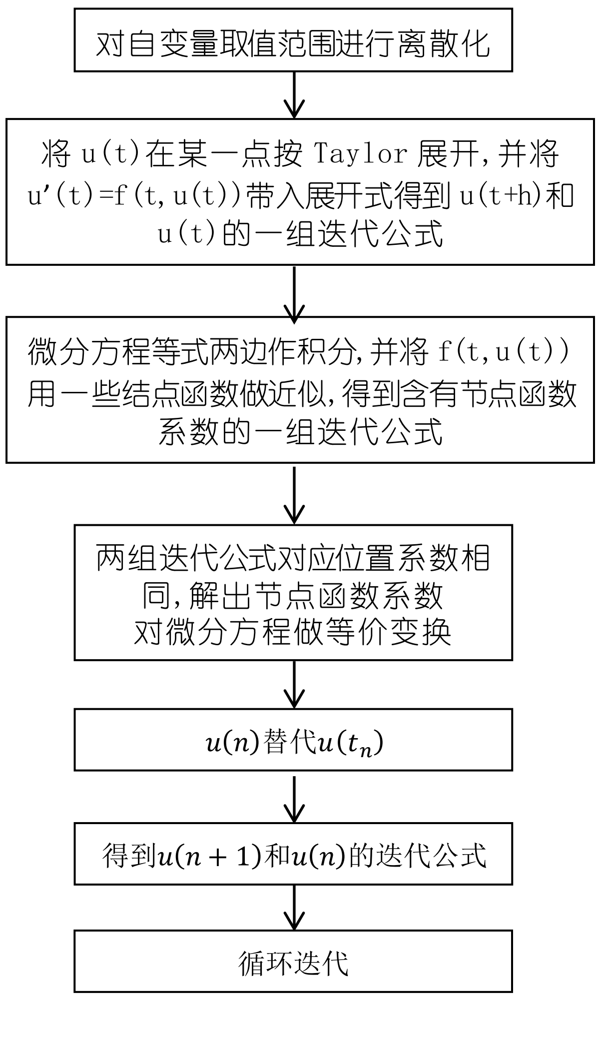
2.改进欧拉法



3. Adams 外插法



4. 四阶 R-K 方法



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: