微分方程数值解实验报告

班级：　　　应数拔尖 1501

姓名：　　　冯洲

学号：　　　U201510104

指导老师：　　　张诚坚

1 实验一

1.1 问题描述及要求

设有初值问题：

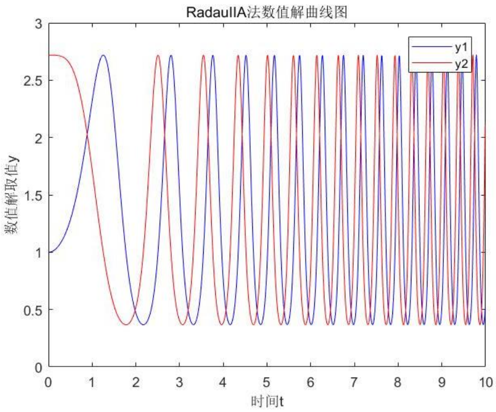
$$\begin{cases} y_1'(t) = 2ty_1(t)\ln[\max(y_2(t), 10^{-3})], \quad t \in [0, 10] \\ y_2'(t) = -2ty_2(t)\ln[\max(y_1(t), 10^{-3})], \quad t \in [0, 10] \\ y_1(0) = 1, \quad y_2(0) = e \end{cases}$$

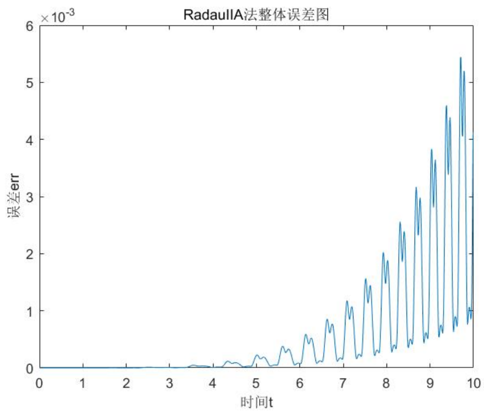
其精确解为$y_1(t) = \exp[\sin(t^2)], y_2(t) = \exp[\cos(t^2)]$. 试取步长 h=0.01，分别应用三阶 Adams-Monlton 方法和三阶 RadauIIA 方法计算该初值问题，做出其数值解的曲线图及整体误差曲线图。

1.2 问题求解

1.2.1RadauIIA 方法

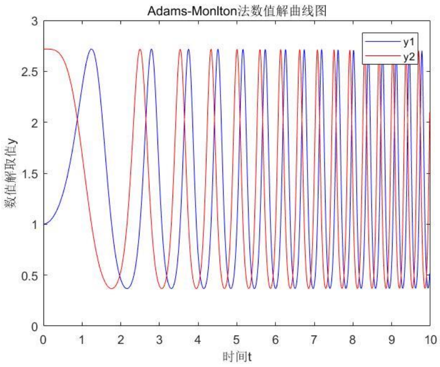
首先，根据三阶 RadauIIA 方法，并利用题设所给条件，画出数值解的曲线图及整体误差曲线图如下：

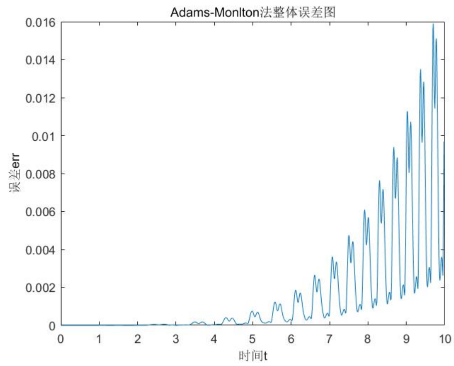




1.2.2Adams-Monlton 方法

然后，利用 Adams-Monlton 方法求出方程的数值解的曲线图及整体误差曲线图如下：





比较两种方法的整体误差图可以看出，两者的误差分布的变化趋势相似，但是两者的误差虽然都很小，但是 RadauIIA 方法的误差最大值要比 Adams-Monlton 方法的小。

1.3 实验一代码

1.3.1 主程序代码

```
clc,clear
%%  输入参数准备
a=0;b=10;y0=[1 exp(1)]';d=2;h=0.01;
%% Radau2
Radau2a(a,b,y0,d,h);
%% adams
adams(a,b,y0,d,h);
```

1.3.2 算法及函数代码附后

2 实验二

2.1 问题描述及要求

试取步长 $h=h_1=h_2=1/64$ ，应用五点差分格式及 Jacobi 迭代法，G-S 迭代法，最佳 JOR 方法，最佳 SOR 方法，最佳单参数 ADI 方法和共轭梯度法，求解边值问题：

$$\begin{cases} \Delta u = 2\pi^2 e^{\pi(x+y)}(\sin\pi x\cos\pi y + \cos\pi x\sin\pi y),\ (x,y)\in G=(0,1)\times(0,1) \\ u=0,\ (x,y)\in\partial G \end{cases}$$

要求：

①作出用共轭梯度法的解曲面图

②作出用共轭梯度法的误差曲面图

③列出诸迭代法的迭代步数

2.2 问题求解

首先，对求解区域进行网格划分，步长$h=h_1=h_2=1/64$。

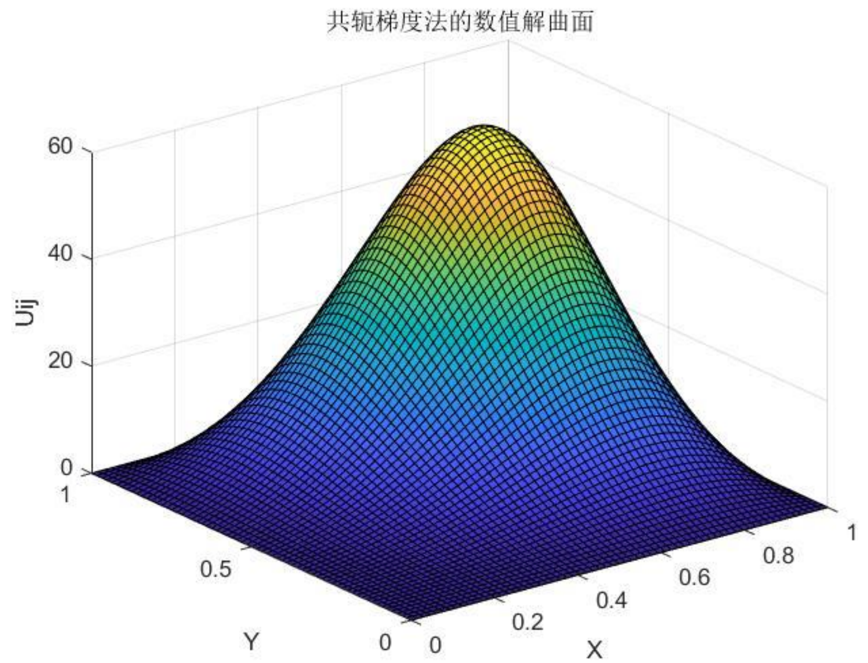
然后，根据方程构造五点差分格式如下：

$$\frac{u_{i+1,j}-2u_{ij}+u_{i-1,j}}{h_1^2}+\frac{u_{i,j+1}-2u_{ij}+u_{i,j-1}}{h_2^2}=f_{ij}$$

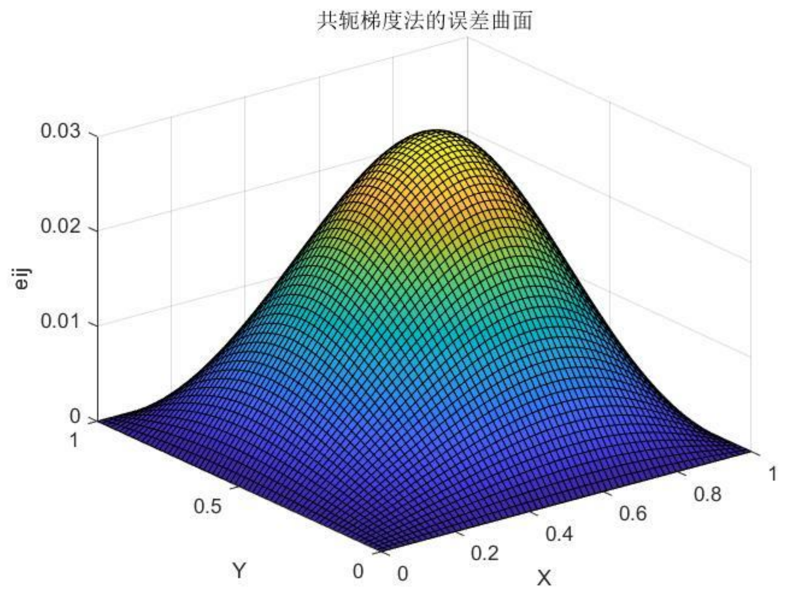
接着，由差分格式导出方程式，加上边值条件，构造问题的线性方程组

最后，由题设要求的诸线性方程组的迭代解法求解该线性方程组，得出相应的数值解。

其中，对于共轭梯度法求出的数值解，画出其数值解曲面如下：



并且通过与精确解进行比较的得到的误差曲面图如下：



最后，在通过诸迭代法求解线性方程组的过程中，取步长 h=1/64，迭代精度要求达到 10^-12，从零初值开始迭代达到要求所需的迭代次数如下表：

迭代次数表

	Jacobi 法	Gauss-Seidel 法	最佳 JOR 法	最佳 SOR 法	最佳 ADI 法	共轭梯度法
迭代次数 k	23116	11856	23116	389	331	257

从表中可以看出，Jacobi 法，Gauss-Seidel 法以及最佳 JOR 法迭代次数很多，而最佳 SOR 法，最佳 ADI 法，共轭梯度法的迭代次数明显较少，其中属共轭梯度法最优。

2.3 实验二代码

2.3.1 主程序代码

```
%主程序,
%% 迭代格式的标准化
clc,clear
N=64;
%给划分排序
p=zeros(N-1,N-1);
for i=1:(N-1)
    for j=1:(N-1)
        if i+j<=N
```

```
            p(N-i,j)=(i+j-1)*(i+j-2)/2+i;
        else
            p(N-i,j)=(N-1)^2-(N-1-i+(2*N-(i+j)-1)*(2*N-(i+j)-2)/2);
        end
    end
end
p2=zeros(N-1,N-1);
for i=1:N-1
    p2(i,:)=p(N-i,:);
end
A=zeros((N-1)^2,(N-1)^2);
b=zeros((N-1)^2,1);
%构造 A 与 b
for i=1:N-1
    for j=1:N-1
        b(p2(i,j))=f(j/N,i/N)/(N^2);
        A(p2(i,j),p2(i,j))=-4;
        if i>1
        A(p2(i,j),p2(i-1,j))=1;
        end
        if j>1
        A(p2(i,j),p2(i,j-1))=1;
        end
        if i<N-1
        A(p2(i,j),p2(i+1,j))=1;
        end
        if j<N-1
        A(p2(i,j),p2(i,j+1))=1;
        end
    end
end
%构造 L,D,U
L=zeros((N-1)^2,(N-1)^2);
D=zeros((N-1)^2,(N-1)^2);
U=zeros((N-1)^2,(N-1)^2);
for i=1:(N-1)^2
    for j=1:(N-1)^2
        if i>j
            L(i,j)=A(i,j);
        else if i==j
                D(i,j)=A(i,j);
            else
                U(i,j)=A(i,j);
            end
        end
    end
end
%构造 ADI 所需的 A1, A2,b1
A1=zeros((N-1)^2,(N-1)^2);
A2=zeros((N-1)^2,(N-1)^2);
h1=-N^2;h2=2*N^2;
for i=1:N-1
    for j=1:N-1
        A1(p2(i,j),p2(i,j))=h2;
        A2(p2(i,j),p2(i,j))=h2;
        if i>1
        A2(p2(i,j),p2(i-1,j))=h1;
        end
        if j>1
        A1(p2(i,j),p2(i,j-1))=h1;
        end
        if i<N-1
        A2(p2(i,j),p2(i+1,j))=h1;
        end
        if j<N-1
        A1(p2(i,j),p2(i,j+1))=h1;
        end
    end
end
b1=-b*(N^2);
%精确解
Z=zeros(N+1,N+1);
for i=1:(N+1)
    for j=1:(N+1)
        Z(i,j)=z((j-1)/N,(i-1)/N);
    end
end
```

```
%% 迭代求解
[X1 T1]=Jacobian(D,L,U,b);
[X2 T2]=GuassSeidel(D,L,U,b);
[X3 T3]= bestJOR(D,L,U,b);
[X4 T4]= bestSOR(D,L,U,b);
[X5 T5]= bestADI(A1,A2,b1);
[X6 T6]= congrad(A,b);
%导出共轭梯度法的解（矩阵版本）
U=zeros(N-1,N-1);
for i=1:N-1
    for j=1:N-1
        U(i,j)=X6(p2(i,j));
    end
end
U=[zeros(N-1,1) U zeros(N-1,1)];
U=[zeros(1,N+1);U;zeros(1,N+1)];
%% 画图及制表
%共轭梯度法的解曲面
x=0:1/N:1;
y=x;
[X Y]=meshgrid(x,y);
figure(1)
surf(X,Y,U)
shading faceted
xlabel('X')
ylabel('Y')
zlabel('Uij')
title('共轭梯度法的数值解曲面')
%共轭梯度法的误差曲面
figure(2)
err=abs(U-Z);
surf(X,Y,err)
shading faceted
xlabel('X')
ylabel('Y')
zlabel('eij')
title('共轭梯度法的误差曲面')
%各迭代法的迭代步数，即 Ti
T1,T2,T3,T4,T5,T6
```

2.3.2 算法及函数代码附后

3 实验三

3.1 问题描述及要求

试构造差分格式求解 Burger 方程的初边值问题

$$\begin{cases} u_t + uu_x = u_{xx}, & x \in (0,1) \ t \in (0,5) \\ u(x,0) = \sin 2\pi x, & x \in [0,1] \\ u(0,t) = u(1,t) = 0, & t \in [0,5] \end{cases}$$

要求：

①给出计算格式和算法代码（取步长$h = \tau = 0.01$）

②做出求解曲面

3.2 问题求解

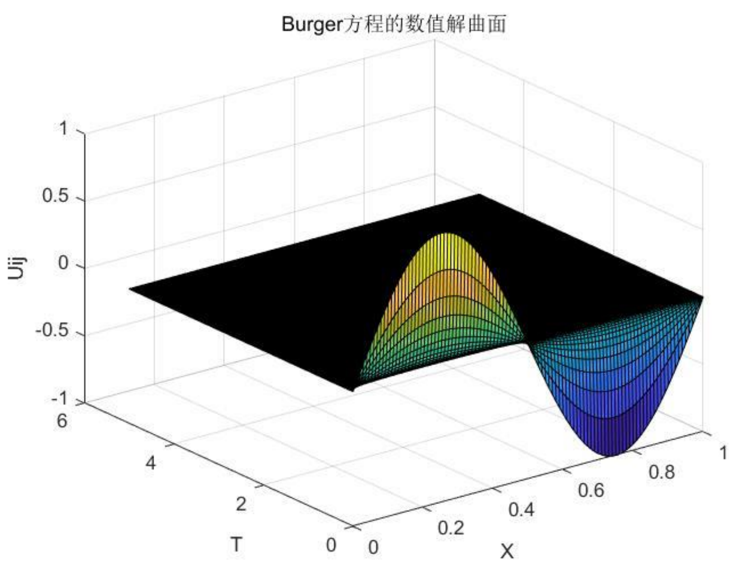
首先，对求解区域进行网格划分，步长$h=\tau=0.01$。

然后，注意到$uu_x=\frac{1}{2}\frac{\partial u^2}{\partial x}$，于是根据方程构造差分格式如下：

$$\frac{u_j^{n+1}-u_j^n}{\tau}+\frac{1}{2}\frac{(u_{j+1}^n)^2-(u_j^n)^2}{h}=\frac{u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}}{h^2}$$

接着，由差分格式导出方程式，加上初边值条件，构造问题的线性方程组

最后，由化简地差分格式可以看出，待求解的线性方程组恰好满足追赶法所需要求，然后由追赶法一层一层地求出的数值解，画出其数值解曲面如下：



3.3 实验三代码

3.3.1 主程序代码

```
clc,clear
%% 求解算法的输入参数
U=zeros(501,101);
U1=0:0.01:1;
U(1,:)=sin(2*pi()*U1);
h=0.01;tt=0.01;r=tt/(h^2);
%左端追赶法的求解矩阵
a=-r*ones(98,1);b=(1+2*r)*ones(99,1);c=-r*ones(98,1);
d=zeros(99,1);
%% 运行求解
for i=2:501
%构造追赶法右端向量
    for j=2:100
```

```
        d(j-1)=f(U(i-1,j),U(i-1,j+1));
    end
 %求 U (i, :)
 U(i,2:100)=chase(a,b,c,d);
end
%% 画出解曲图
x=0:0.01:1;
t=0:0.01:5;
[X T]=meshgrid(x,t);
figure(1)
surf(X,T,U)
shading faceted
xlabel('X')
ylabel('T')
zlabel('Uij')
title('Burger 方程的数值解曲面')
```

3.3.2 算法及函数代码

```
function x=chase(a,b,c,d)
n=length(b);
x=zeros(n,1);
f=zeros(n,1);
g=zeros(n,1);
f(1)=c(1)/b(1);g(1)=d(1)/b(1);
for i =2:(n-1)
h(i)=b(i)-f(i-1)*a(i-1);f(i)=c(i)/h(i);
g(i)=(d(i)-g(i-1)*a(i-1))/h(i);
end
g(n)=(d(n)-g(n-1)*a(n-1))/(b(n)-f(n-1)*a(n-1));x(n)=g(n);
for i=(n-1):-1:1
x(i)=g(i)-f(i)*x(i+1);
end

function  z=f( x,y)
%迭代格式转化之后的右端
z=x-0.5*(y^2-x^2);
end
```

4 实验四

4.1 问题描述及要求

用线性元法求下列边值问题的数值解（取单元长度 h=0.1）:

$$-y''+\frac{\pi^2}{4}y=\frac{\pi^2}{2}\sin\frac{\pi}{2}x,\ \ 0<x<1,y(0)=0,y'(1)=0$$

（精确解为$y=\sin\frac{\pi}{2}x$）

要求:

①要求在同一坐标系下给出真解及有限元解的曲线图

②计算有限元解的误差$\|u-u_h\|_{L^2}$，$\|u-u_h\|_{H^1}$

4.2 问题求解

第一步，对求解区域进行网格划分，按照要求取$h=0.1$

第二步，基于虚功原理，由边值条件及方程形式得到 Garlerkin 方程为：

$$\sum_{i=1}^{n} a(\varphi_i,\varphi_j)u_i=\int_a^b f\varphi_j dx,\ j=1,2,\cdots,n.$$

第三步，由内积及基函数的表达式算出方程的系数矩阵为

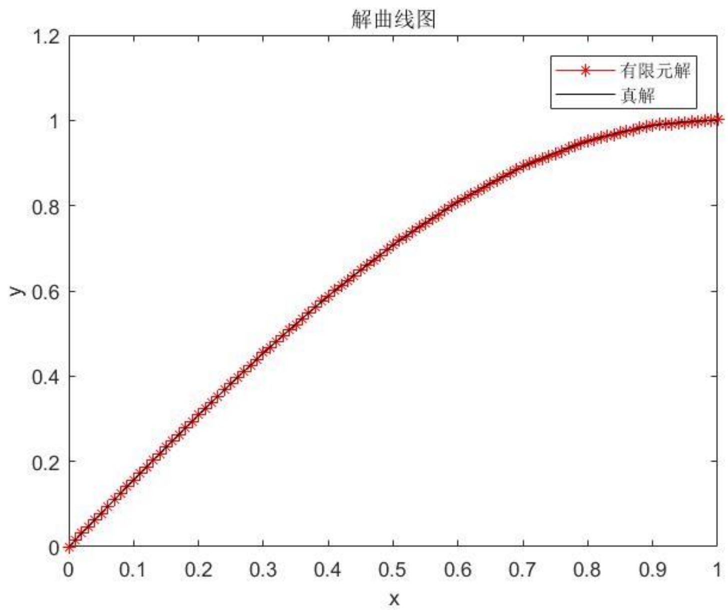
$$A=\begin{bmatrix} \frac{2}{h}+\frac{\pi^2}{6}h & -\frac{1}{h}+\frac{\pi^2}{24}h & & \\ -\frac{1}{h}+\frac{\pi^2}{24}h & \ddots & \ddots & \\ & \ddots & \ddots & -\frac{1}{h}+\frac{\pi^2}{24}h \\ & & -\frac{1}{h}+\frac{\pi^2}{24}h & \frac{1}{h}+\frac{\pi^2}{12}h \end{bmatrix}$$

由此看出系数矩阵为三对角型，适用于追赶法求解。

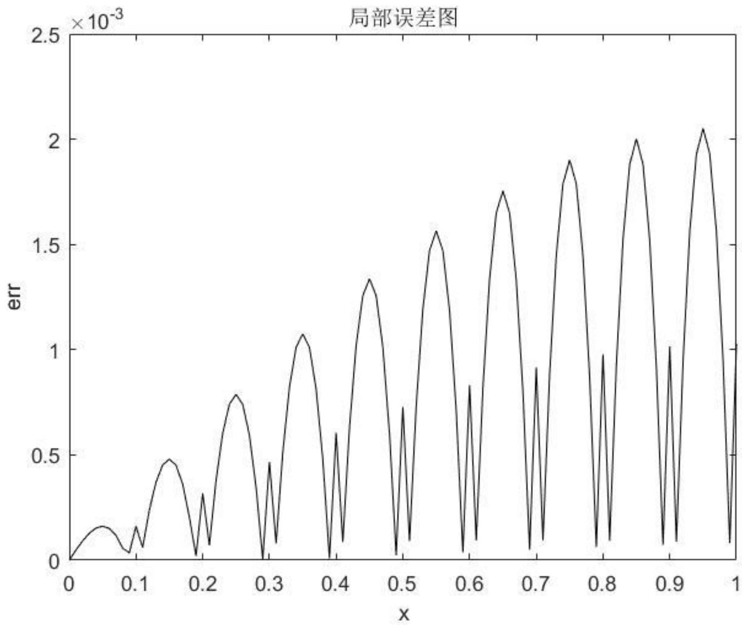
第四步，由前一步的计算的系数矩阵及右端向量的结果，由追赶法计算 Galerkin 方程的解，得到基函数的系数：

$$u=[u_1\ \cdots\ u_{10}]$$

第五步，由计算的有限元解以及真解，在同一坐标系下画出曲线图如下：



由于在上图中无法明显辨别有限元解与真解的区别，所以在各计算点出画出局部绝对误差曲线图如下：



最后根据要求计算出有限元解与真解之间的两种整体误差如下：

有限元解的整体误差表

$\|u-u_h\|_{L^2}$	$\|u-u_h\|_{H^1}$
0.0009765	0.0513343

4.3 实验四代码

4.3.1 主程序代码

```
clc,clear
%迭代系数矩阵的构造
h=0.1;
u=zeros(10,1);%目标系数
%右端向量
d=zeros(10,1);
for j=1:9
    d(j)=(4/h*sin(j*h*pi()/2))-2/h*(sin((j+1)*h*pi()/2)+sin((j-1)*h*pi()/2));
end
d(10)=2/h*(1-sin((1-h)*pi()/2));
%构造追赶法系数矩阵
a=ones(9,1);
b=ones(10,1);
c=ones(9,1);
a=(-1/h+pi()^2*h/24)*a;
c=(-1/h+pi()^2*h/24)*c;
b=(2/h+pi()^2*h/6)*b;
b(10)=1/h+pi()^2*h/12;
%利用追赶法求解并展示曲线
```

```
u=chase(a,b,c,d);
x1=0:0.01:1;
y1=sin(pi()/2*x1);
y2=zeros(101,1);
for i=1:9
    for j=0:1:10
        y2(10*i+j+1)=u(i+1)/h*j*0.01+u(i)*(1-1/h*j*0.01);
    end
end
for j=0:10
    y2(j+1)=u(1)/h*j*0.01;
end
figure(1)
plot(x1,y2,'r*-',x1,y1,'k-');
xlabel('x' ); ylabel( 'y' );
legend('有限元解','真解');
title('解曲线图');
%画出局部误差图
figure(2)
r=abs(y1-y2');
plot(x1,r,'k-');
xlabel('x' ); ylabel( 'err' );
title('局部误差图');
%%
%求两种误差
%基函数内积矩阵
A1=zeros(10,10);
for i=1:10
    if i>1 && i<10
    A1(i,i)=2/3*h;
    A1(i,i+1)=1/6*h;
    A1(i,i-1)=1/6*h;
    else
        if i==1
           A1(i,i)=2/3*h;
           A1(i,i+1)=1/6*h;
        else
            A1(i,i-1)=1/6*h;
            A1(i,i)=1/3*h;
        end
    end
end
A1(10,10)=1/3*h;
%基函数的导数内积矩阵
A2=zeros(10,10);
for i=1:10
  if i>1 && i<10
    A2(i,i)=2/h;
    A2(i,i+1)=-1/h;
    A2(i,i-1)=-1/h;
  else
        if i==1
           A2(i,i)=2/h;
           A2(i,i+1)=-1/h;
        else
            A2(i,i-1)=-1/h;
            A2(i,i)=1/h;
        end
  end
end
A2(10,10)=1/h;
%真解与基函数的内积向量
A3=2/(pi()^2)*d;
%内积的导数与真解的导数的内积向量
A4=zeros(10,1);
for i=1:9
    A4(i)=1/h*(2*sin(pi()*i*h/2)-sin(pi()*(i-1)*h/2)-sin(pi()*(i+1)*h/2));
end
A4(10)=1/h*(1-sin((1-h)*pi()/2));
%% 计算误差
%误差 L2
B1=zeros(10,10);
for i=1:10
    for j=1:10
        B1(i,j)=A1(i,j)*u(i)*u(j);
    end
end
```

```
B2=u.*A3;
r1=(1/2-2*sum(B2(:))+sum(B1(:)))^0.5
%误差 H1
B3=zeros(10,10);
for i=1:10
    for j=1:10
        B3(i,j)=A2(i,j)*u(i)*u(j);
    end
end
B4=u.*A4;
r2=(pi()^2/8-2*sum(B4(:))+sum(B3(:)))^0.5;
r2=r2+r1
```

4.3.2 算法及函数代码

```
function x=chase(a,b,c,d)
n=length(b);
x=zeros(n,1);
f=zeros(n,1);
g=zeros(n,1);
f(1)=c(1)/b(1);g(1)=d(1)/b(1);
for i =2:(n-1)
h(i)=b(i)-f(i-1)*a(i-1);f(i)=c(i)/h(i);
g(i)=(d(i)-g(i-1)*a(i-1))/h(i);
end
g(n)=(d(n)-g(n-1)*a(n-1))/(b(n)-f(n-1)*a(n-1));x(n)=g(n);
for i=(n-1):-1:1
x(i)=g(i)-f(i)*x(i+1);
end
```

5 附录

5.1 实验一算法代码

```
function z=Radau2a(a,b,y0,d,h)
A=[5/12 -1/12;3/4 1/4];
B=[3/4 1/4];c=[1/3,1]';
A1=[0 0 0;1/3 0 0;0 2/3 0]; %三阶 Heun 方法计算初值
B1=[1/4 0 3/4];t0=a;s=2;
Z1=[];Z2=[];Z3=[];Z4=[];
i=1;
for n=a+h:h:b
    Z1(i)=t0;Z2(i)=y0(1);Z3(i)=y0(2);
    Z4(i)=norm(y0-[exp(sin(t0*t0)) exp(cos(t0*t0))]',2);
    err1=1;err2=1;t=kron(ones(s,1),t0)+h*c;
    tc11=t0;tc12=t0+c(1)*h/3;tc13=t0+2*c(1)*h/3;
    Y11=y0;Y12=y0+h*c(1)*kron(A1(2,1),eye(d))*fa(tc11,Y11);
    Y13=y0+h*c(1)*kron(A1(3,1:2),eye(d))*[fa(tc11,Y11);fa(tc12,Y12)];

Y10=y0+h*c(1)*kron(B1(1:3),eye(d))*[fa(tc11,Y11);fa(tc12,Y12);fa(tc13,Y13)];
    tc21=t0;tc22=t0+c(2)*h/3;tc23=t0+c(2)*h/3;
    Y21=y0;Y22=y0+h*c(2)*kron(A1(2,1),eye(d))*fa(tc21,Y21);
    Y23=y0+h*c(2)*kron(A1(3,1:2),eye(d))*[fa(tc21,Y21);fa(tc12,Y22)];

Y20=y0+h*c(2)*kron(B1(1:3),eye(d))*[fa(tc21,Y21);fa(tc22,Y22);fa(tc23,Y23)];
    Y0=[Y10;Y20];
    while err1>=10^(-12) && err2>=10^(-12)
        F=[fa(t(1),[Y0(1);Y0(2)]);fa(t(2),[Y0(3);Y0(4)])];
        r=Y0-kron(ones(s,1),y0)-h*kron(A,eye(d))*F;
        dF=[dfa(t(1),[Y0(1);Y0(2)])                    zeros(2,2);zeros(2,2)
dfa(t(2),[Y0(3);Y0(4)])];
        Y1=Y0-inv(eye(s*d)-h*kron(A,eye(d))*dF)*r;
        err1=norm(Y1-Y0);err2=norm(r);Y0=Y1;
    end
    y1=y0+h*kron(B,eye(d))*[fa(t(1),[Y0(1);Y0(2)]);fa(t(2),[Y0(3);Y0(4)])];
    t0=t0+h;y0=y1;i=i+1;
end
```

```
figure(1)
plot(Z1,Z2,'b',Z1,Z3,'r');
xlabel('时间 t');ylabel('数值解取值 y');
legend('y1','y2','Location','Northeast');
title('RadauIIA 法数值解曲线图');
figure(2)
plot(Z1,Z4);
xlabel('时间 t');ylabel('误差 err');
title('RadauIIA 法整体误差图');

function z=adams(a,b,y0,d,h)
t0=a;alpha=[0 -1 -1]'; beta=[-1 8 5]'/12;
A=[0 0 0;1/3 0 0;0 2/3 0];
B=[1/4 0 3/4];
c=[0 1/3 2/3]';
Z1=[];Z2=[];Z3=[];Z4=[];
i=1;
tc1=t0+c(1)*h;tc2=t0+c(2)*h;tc3=t0+c(3)*h;
Y1=y0;Y2=y0+h*kron(A(2,1),eye(d))*fa(tc1,Y1);
Y3=y0+h*kron(A(3,1:2),eye(d))*[fa(tc1,Y1);fa(tc2,Y2)];
y1=y0+h*kron(B(1:3),eye(d))*[fa(tc1,Y1);fa(tc2,Y2);fa(tc3,Y3)];
for n=a+2*h:h:b
    Z1(i)=t0;Z2(i)=y1(1);Z3(i)=y1(2);
%     Z4(i)=norm(y1-[(2*pi-t0-h)*cos(t0+h)  (2*pi-t0-h)*sin(t0+h)]',2);
    Z4(i)=norm(y1-[exp(sin((t0+h)*(t0+h)))  exp(cos((t0+h)*(t0+h)))]',2);
    t1=t0+h;t2=t0+2*h;tc1=t0+(1+c(1))*h;tc2=t0+(1+c(2))*h;tc3=t0+(1+c(3))*h;
    Y1=y1;Y2=y1+h*kron(A(2,1),eye(d))*fa(tc1,Y1);
    Y3=y1+h*kron(A(3,1:2),eye(d))*[fa(tc1,Y1);fa(tc2,Y2)];
    y20=y1+h*kron(B(1:3),eye(d))*[fa(tc1,Y1);fa(tc2,Y2);fa(tc3,Y3)];
    w=h*(beta(1)*fa(t0,y0)+beta(2)*fa(t1,y1))-(alpha(1)*y0+alpha(2)*y1);
    err1=1;err2=1;
    while err1>10^(-12)  && err2>=10^(-12)
        r=y20-h*beta(3)*fa(t2,y20)-w;
        y21=y20-inv(eye(d)-h*beta(3)*dfa(t2,y20))*r;
        err1=norm(y21-y20);err2=norm(r);
        y20=y21;
    end
y2=y20;t0=t0+h;y0=y1;y1=y2;
i=i+1;
end
figure(3)
plot(Z1,Z2,'b',Z1,Z3,'r');
xlabel('时间 t');ylabel('数值解取值 y');
legend('y1','y2','Location','Northeast');
title('Adams-Monlton 法数值解曲线图');
figure(4)
plot(Z1,Z4);
xlabel('时间 t');ylabel('误差 err');
title('Adams-Monlton 法整体误差图');

function u=fa(t,Y)
u=[2*t*Y(1)*log(max(Y(2),10^(-3)))  (-2)*t*Y(2)*log(max(Y(1),10^(-3)))]';

function z=dfa(tl,Y)
syms x y t;
if Y(1)>10^(-3)  && Y(2)>10^(-3)
    J=jacobian([2*t*x*log(y)  (-2)*t*y*log(x)]',[x y]);
end
if Y(1)>10^(-3)  && Y(2)<=10^(-3)
    J=jacobian([2*t*x*log(10^(-3))  (-2)*t*y*log(x)]',[x y]);
end
if Y(1)<=10^(-3)  && Y(2)>10^(-3)
    J=jacobian([2*t*x*log(y)  (-2)*t*y*log(10^(-3))]',[x y]);
end
if Y(1)<=10^(-3)  && Y(2)<=10^(-3)
    J=jacobian([2*t*x*log(10^(-3))  (-2)*t*y*log(10^(-3))]',[x y]);
end
z=eval(subs(J,{x,y,t},{Y(1),Y(2),tl}));
```

5.2 实验二算法代码

```
function [X t]= Jacobian(D,L,U,b)
A=-D^(-1)*(L+U);B=D^(-1)*b;
```

```
n=length(b);X0=zeros(n,1); r0=B;
t=0;
while norm(r0)>=10^(-12)
    X1=A*X0+B;
    r0=X1-X0;
    X0=X1;
    t=t+1;
end
X=X0;
t;
end

function [X t]= GuassSeidel(D,L,U,b)
A=-(D+L)^(-1)*U;B=(D+L)^(-1)*b;
n=length(b);X0=zeros(n,1); r0=B;
t=0;
while norm(r0)>=10^(-12)
    X1=A*X0+B;
    r0=X1-X0;
    X0=X1;
    t=t+1;
end
X=X0;
end

function [X t]= bestJOR(D,L,U,b )
n=length(b);
I=eye(n);
C=-D^(-1)*(L+U);
w=2/(2-max(eig(C))-min(eig(C)));
A=I-w*D^(-1)*(L+U+D);B=w*D^(-1)*b;
X0=zeros(n,1); r0=B;
t=0;
while norm(r0)>=10^(-12)
    X1=A*X0+B;
    r0=X1-X0;
    X0=X1;
    t=t+1;
end
X=X0;
end

function [X t]= bestSOR(D,L,U,b)
n=length(b);
I=eye(n);
C=-D^(-1)*(L+U);
w=2/(1+(1-norm(eig(C),inf)^2)^(0.5));
A=(D+w*L)^(-1)*((1-w)*D-w*U);B=w*(D+w*L)^(-1)*b;
X0=zeros(n,1); r0=B;
t=0;
while norm(r0)>=10^(-12)
    X1=A*X0+B;
    r0=X1-X0;
    X0=X1;
    t=t+1;
end
X=X0;
end

function [X t]= bestADI(A1,A2,b1)
%最优 ADI 方法
w=(1/64)^2/(2*(sin(pi()/64)));
n=length(b1);
I=eye(n);
T=(I+w*A2)^(-1)*(I-w*A1)*(I+w*A1)^(-1)*(I-w*A2);
B=(I+w*A2)^(-1)*(I-w*A1)*(I+w*A1)^(-1)*w*b1+(I+w*A2)^(-1)*w*b1;
X0=zeros(n,1);r0=B;
t=0;
while norm(r0)>=10^(-12)
X1=T*X0+B;
r0=X1-X0;
X0=X1;
t=t+1;
end
X=X0;
end
```

```
function [X t]= congrad(A,b)
%共轭梯度法
n=length(b); X0=zeros(n,1);r0=b-A*X0; P0=r0 ;
t=0;
while norm(r0)>=10^(-12)
alpha=r0'*r0/(P0'*A*P0); X1=X0+alpha*P0;
r1=r0-alpha*A*P0; beta=r1'*r1/(r0'*r0);
P1=r1+beta*P0; r0=r1; X0=X1 ; P0=P1 ;
t=t+1;
end
X=X0;
end

function zz= z( x ,y )
%精确解
zz=exp(pi()*(x+y))*sin(pi()*x)*sin(pi()*y);
end

function X= f(x,y)
%非齐次项
X=2*pi()^2*exp(pi()*(x+y))*(sin(pi()*x)*cos(pi()*y)+sin(pi()*y)*cos(pi()*x));
End
```

6 参考文献及资料

【1】 微分方程数值解法/李荣华，刘播,第四版—北京：高等教育出版社

【2】 计算方法/张诚坚，何南忠，覃婷婷，第二版—高等教育出版社

【3】 微分方程数值解课堂资料