Matlab 第 9 次作业批改参考

第一题

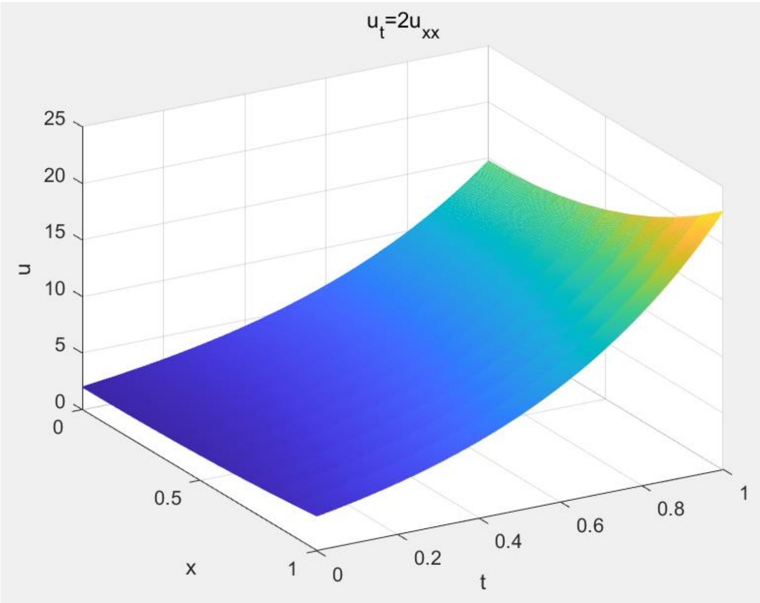
用 Forward Difference Method 解方程$u_t = 2u_{xx}, 0 \le x \le 1, 0 \le t \le 1$，其初边值条件为

$$\begin{cases} u(x,0) = 2\cosh x, & 0 \le x \le 1 \\ u(0,t) = 2e^{2t}, & 0 \le t \le 1 \\ u(1,t) = (e^2+1)e^{2t-1}, & 0 \le t \le 1 \end{cases}$$

（1）取步长$h = 0.1$，$k = 0.002$，用 mesh 画出近似解。要求给出代码和图像，注意图像基本要素。

（2）若步长 h 不变，取$k = 0.004$，同样给出图像并观察发生了什么现象。

（1）



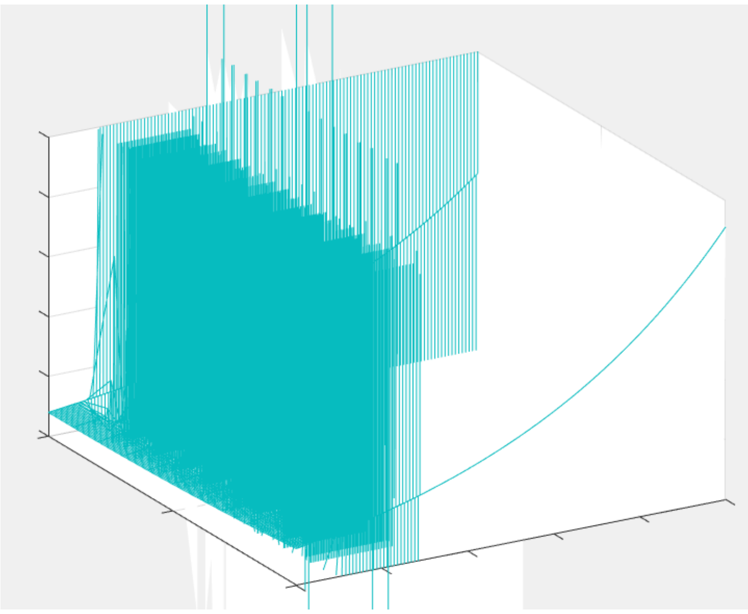
图像只是角度问题，其实大多数人的求解和作图是完成正确的。

```
hw9_1_1.m
clear,clc;
D=2;
f=@(x) 2*cosh(x);
l=@(t) 2*exp(2*t);
r=@(t) (exp(2)+1)*exp(2*t-1);
xl=0;xr=1;tb=0;tt=1;
h=0.1;k=0.002;
w=heatfd(D,f,l,r,[xl,xr],[tb,tt],h,k);
x=xl:h:xr;t=tb:k:tt;
mesh(x,t,w);
title 'u_t=2u_x_x'
xlabel 'x'; ylabel 't';zlabel 'u';
view(60,30);axis([xl xr tb tt 0 25]);
```

heatfd.m

```
function w=heatfd(D,f,l,r,x,y,h,k)
xl=x(1);xr=x(2);yb=y(1);yt=y(2);
M=(xr-xl)/h; N=(yt-yb)/k;    m=M-1; n=N;
sigma=D*k/(h*h);
A=diag(1-2*sigma*ones(m,1))+diag(sigma*ones(m-1,1),1)+diag(sigma*ones(m-1,1),-1);
lside=l(yb+(0:n)*k);    rside=r(yb+(0:n)*k);
w(:,1)=f(xl+(1:m)*h)';
for j=1:n
        w(:,j+1)=A*w(:,j)+sigma*[lside(j);zeros(m-2,1);rside(j)];
end
w=[lside;w;rside]';
end
```

（2）



hw9_1_2.m

```
clear,clc;
D=2;
f=@(x) 2*cosh(x);
l=@(t) 2*exp(2*t);
r=@(t) (exp(2)+1)*exp(2*t-1);
xl=0;xr=1;tb=0;tt=1; h=0.1;k=0.004;
w=heatfd(D,f,l,r,[xl,xr],[tb,tt],h,k);
x=xl:h:xr;t=tb:k:tt;
mesh(x,t,w);
title 'u_t=2u_x_x'
xlabel 'x'; ylabel 't';zlabel 'u';
view(60,30);axis([xl xr tb tt 0 25]);
```

现象：计算不稳定，结果发散。

第二题

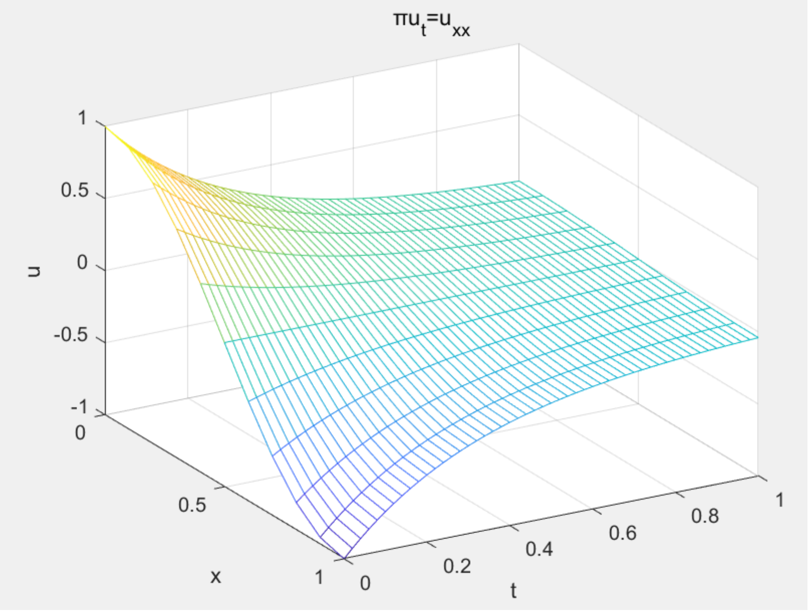
用 Backward Difference Method 解方程 $\pi u_t = u_{xx}, 0 \le x \le 1, 0 \le t \le 1$,初边值条件为

$$\begin{cases} u(x,0) = \cos \pi x, & 0 \le x \le 1 \\ u(0,t) = e^{-\pi t}, & 0 \le t \le 1 \\ u(1,t) = -e^{-\pi t}, & 0 \le t \le 1 \end{cases}$$

（1）取步长 $h = 0.1$， $k = 0.02$，用 mesh 画出近似解。要求给出代码和图像，注意图像基本要素。

（2）已知其解析解为 $u(x,t) = e^{-\pi t} \cos \pi x$，在 $(x,t) = (0.3,1)$ 处作一张准确值、近似值及误差的表，取步长 $h = 0.1$ 及 $k = 0.02, 0.01, 0.005$。

（1）



hw9_2_1.m

```
clear,clc;
f=@(x) cos(pi*x);
l=@(t) exp(-pi*t);
r=@(t) -exp(-pi*t);
D=1/pi;
xl=0;xr=1;tb=0;tt=1;
h=0.1;k=0.02;
w=heatbd(D,f,l,r,[xl,xr],[tb,tt],h,k);
x=xl:h:xr;t=tb:k:tt;
mesh(x,t,w);
title 'π u_t=u_x_x'
xlabel 'x'; ylabel 't';zlabel 'u';
view(60,30);axis([xl xr tb tt -inf inf])
```

heatbd.m

```
function w=heatbd(D,f,l,r,x_interval,y_interval,h,k)
xl=x_interval(1);xr=x_interval(2);
yb=y_interval(1);yt=y_interval(2);
M=(xr-xl)/h; N=(yt-yb)/k; m=M-1; n=N;
sigma=D*k/(h*h);
A=diag(1+2*sigma*ones(m,1))+diag(-sigma*ones(m-1,1),1)...
    +diag(-sigma*ones(m-1,1),-1);
lside=l(yb+(0:n)*k);
rside=r(yb+(0:n)*k);
w(:,1)=f(xl+(1:m)*h)';
for j=1:n
    w(:,j+1)=A\(w(:,j)+sigma*[lside(j+1);zeros(m-2,1);rside(j+1)]);
end
w=[lside;w;rside]';
end
```

几个人 BD 方法的迭代式错了，书中的参考代码也是错的，所以还是需要真正理解算法流程编写程序更好

（2）

k	exact value	approximate value	error
0.02	0.025401	0.025857	0.000456
0.01	0.025401	0.025672	0.000271
0.005	0.025401	0.025581	0.000181

hw9_2_2.m

```
clear,clc;
x=0.3;t=1;
y_exact=exp(-pi*t)*cos(pi*x);
f=@(x) cos(pi*x);
l=@(t) exp(-pi*t);
r=@(t) -exp(-pi*t);
D=1/pi;
xl=0;xr=1;tb=0;tt=1;
h=0.1;k=[0.02,0.01,0.005];
w=zeros(1,3);
w1=heatbd(D,f,l,r,[xl,xr],[tb,tt],h,k(1));
w2=heatbd(D,f,l,r,[xl,xr],[tb,tt],h,k(2));
w3=heatbd(D,f,l,r,[xl,xr],[tb,tt],h,k(3));
w(1)=w1(end,4);
w(2)=w2(end,4);
w(3)=w3(end,4);
fprintf('k\t\t exact value\t approximate value\t error\n');
fprintf('%.2f\t %f\t\t %f\t\t\t %f\n',k(1),y_exact,w(1),w(1)-y_exact);
fprintf('%.2f\t %f\t\t %f\t\t\t %f\n',k(2),y_exact,w(2),w(2)-y_exact);
fprintf('%.3f\t %f\t\t %f\t\t\t %f\n',k(3),y_exact,w(3),w(3)-y_exact);
```

第三题

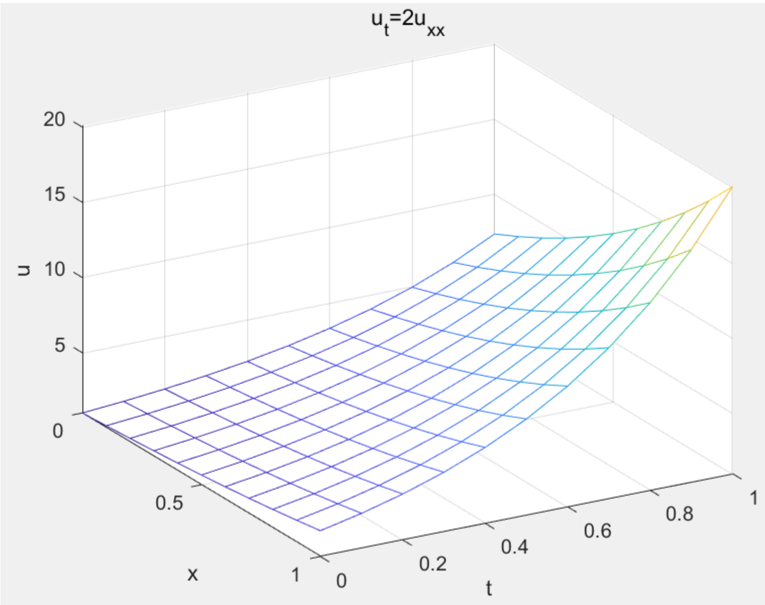
用 Crank-Nicolson Method 解方程 $u_t = 2u_{xx}, 0 \le x \le 1, 0 \le t \le 1$，其初边值条件为

$$\begin{cases} u(x,0) = e^x, & 0 \le x \le 1 \\ u(0,t) = e^{2t}, & 0 \le t \le 1 \\ u(1,t) = e^{2t+1}, & 0 \le t \le 1 \end{cases}$$

（1）取步长为 $h = k = 0.1$，画出近似解，给出图像和代码，注意基本要素。

（2）已知其解析解为 $u(x,t) = e^{2t+x}$，分别用 Forward Difference Method 和 Backward Difference Method 求解该方程，均取步长为 $h = 0.1$，$k = 0.0025$，再分别用 Crank-Nicolson Method 和 MATLAB 内置函数 pdepe 进行求解，均取步长为 $h = k = 0.1$，要求作出四种求解方法与解析解的在 $t = 1$ 时的误差曲线，要求给出代码和图像，注意基本要素。

（1）



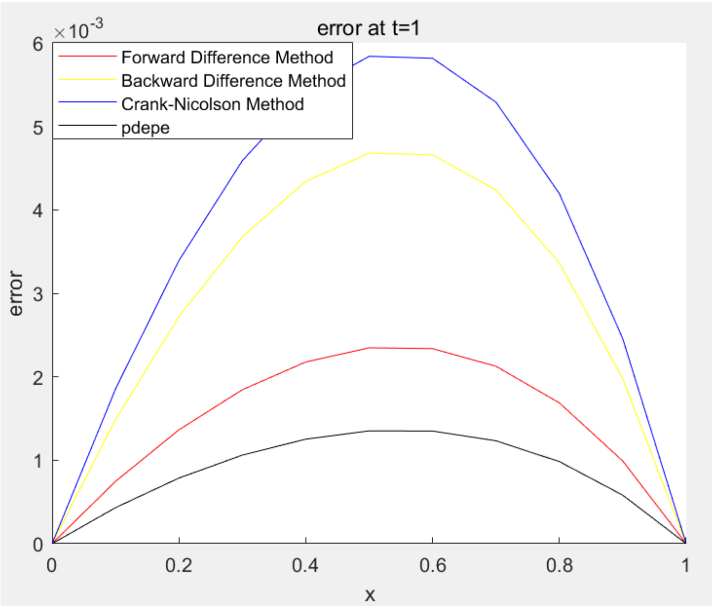
hw9_3_1.m

```
clear,clc;
f=@(x) exp(x);
l=@(t) exp(2*t);
r=@(t) exp(2*t+1);
D=2;
xl=0;xr=1;tb=0;tt=1;
h=0.1;k=0.1;
w=heatcn(D,f,l,r,[xl,xr],[tb,tt],h,k);
x=xl:h:xr;t=tb:k:tt;
mesh(x,t,w);
title 'u_t=2u_x_x'
xlabel 'x'; ylabel 't';zlabel 'u';
view(60,30);axis([xl xr tb tt -inf inf]);
```

heatcn.m

```
function w=heatcn(D,f,l,r,x_interval,y_interval,h,k)
xl=x_interval(1);xr=x_interval(2);yb=y_interval(1);yt=y_interval(2);
M=(xr-xl)/h; N=(yt-yb)/k; m=M-1; n=N;
sigma=D*k/(h*h);
A=diag(2+2*sigma*ones(m,1))+diag(-sigma*ones(m-1,1),1)...
    +diag(-sigma*ones(m-1,1),-1);
B=diag(2-2*sigma*ones(m,1))+diag(sigma*ones(m-1,1),1)...
    +diag(sigma*ones(m-1,1),-1);
lside=l(yb+(0:n)*k);
rside=r(yb+(0:n)*k);
w(:,1)=f(xl+(1:m)*h)';
for j=1:n
    w(:,j+1)=A\(B*w(:,j)+sigma*[lside(j)+lside(j+1);zeros(m-2,1);rside(j)+rside(j+1)]);
end
w=[lside;w;rside]';
end
```

（2）



hw9_3_2.m

从这个图像其实就能看出前面的算法是否正确，有的人没有取绝对值也是对的，就是图像翻转了而已。

```
clear,clc;
f=@(x) exp(x);
l=@(t) exp(2*t);
r=@(t) exp(2*t+1);
D=2;
xl=0;xr=1;tb=0;tt=1;
h=0.1;k=[0.0025,0.1];
w_fd=heatfd(D,f,l,r,[xl,xr],[tb,tt],h,k(1));
w_bd=heatbd(D,f,l,r,[xl,xr],[tb,tt],h,k(1));
```

```
w_cn=heatcn(D,f,l,r,[xl,xr],[tb,tt],h,k(2));
%pdepe
x_pdepe=xl:h:xr;t_pdepe=tb:k(2):tt;
m=0;
w_pdepe=pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x_pdepe,t_pdepe);
%plot error at t=1
x=xl:h:xr;
u_exact=exp(2+x);
figure;hold on;
plot(x,abs(w_fd(end,:)-u_exact),'r');
plot(x,abs(w_bd(end,:)-u_exact),'y');
plot(x,abs(w_cn(end,:)-u_exact),'b');
plot(x,abs(w_pdepe(end,:)-u_exact),'k');
legend('Forward Difference Method','Backward Difference Method',...
    'Crank-Nicolson Method','pdepe','Location','northwest');
xlabel 'x';ylabel 'error';title 'error at t=1';
function [c,f,s] = pdex1pde(x,t,u,dudx) % Equation to solve
c = 0.5;
f = dudx;
s = 0;
end
function u0 = pdex1ic(x) % Initial conditions
u0 = exp(x);
end
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t) % Boundary conditions
pl = ul-exp(2*t);
ql = 0;
pr = ur-exp(2*t+1);
qr = 0;
end
```

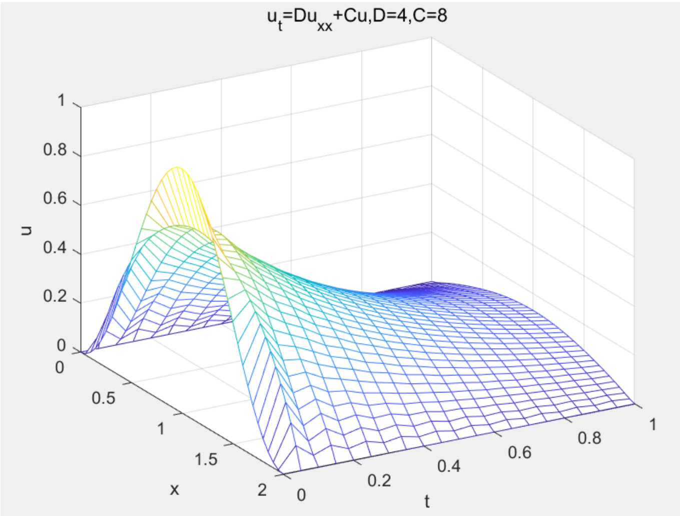
第四题

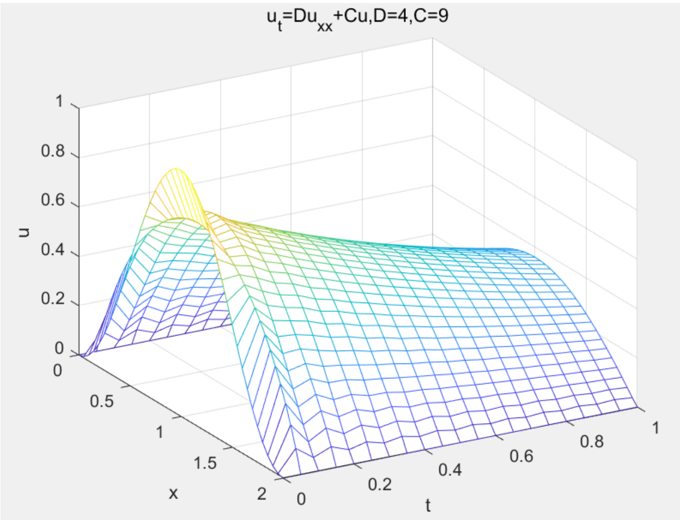
$$\begin{cases} u_t = Du_{xx} + Cu \\ u(x,0) = \sin^2 \dfrac{\pi}{L} x, \quad 0 \le x \le L \\ u(0,t) = 0, \qquad\qquad 0 \le t \le T \\ u(L,t) = 0, \qquad\qquad 0 \le t \le T \end{cases}$$

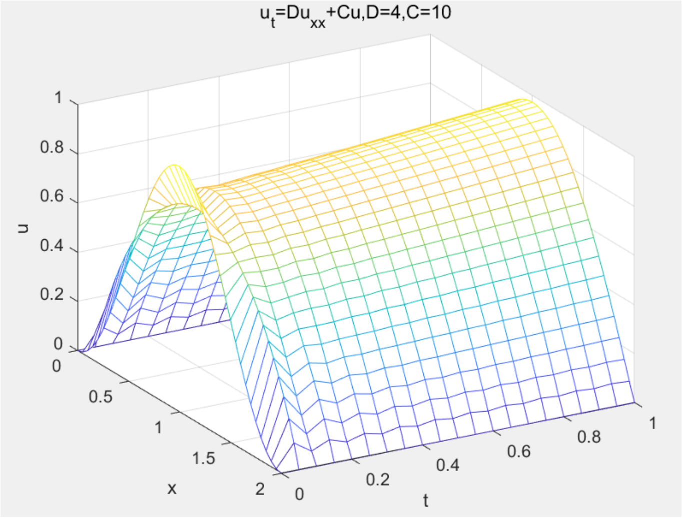
（1）令 $D=4$， $L=2$， $T=1$，分别取 $C=8,\ 9,\ 10$，取步长 $h=k=0.05$，用 Crank-Nicolson Method 进行求解，分别绘制三个求解的结果。若 $u(x,t)$ 表示在位置 x 和时间 t 的人口密度，请分析其规律。要求给出代码、图像及分析。（可参考 *Ref.[1]* P390）

（2）令 $D=1$， $L=10$，取步长 $h=k=0.05$，若要人可以长时间地生存，这里的 C 最小为多少？要求给出代码和结果。（可用公式 $C>\pi^2 D/L^2$ 对你的结果进行校验）

（1）







hw9_4_1.m

```
clear,clc;
D=4; C=[8,9,10];
xl=0;xr=2;tb=0;tt=1;
h=0.05;k=0.05;
for i=1:3
    [w,z]=population(D,C(i),[xl,xr],[tb,tt],h,k);
    x=xl:h:xr;t=tb:k:tt;
    figure;
    mesh(x,t,w);
    title(['u_t=Du_x_x+Cu,D=4,C=',num2str(C(i))]);
    xlabel 'x'; ylabel 't';zlabel 'u';
    view(60,30);axis([xl xr tb tt -inf inf]);
end
```

population.m

```
function [w,z]=population(D,C,x_intercal,y_interval,h,k)
xl=x_intercal(1);xr=x_intercal(2);yb=y_interval(1);yt=y_interval(2);
f=@(x) sin(pi/xr*x).^2;
l=@(t) 0*t;
r=@(t) 0*t;
M=(xr-xl)/h; N=(yt-yb)/k; m=M-1; n=N;
sigma=D*k/(h*h);
A=diag(2-k*C+2*sigma*ones(m,1))+diag(-sigma*ones(m-1,1),1)...
    +diag(-sigma*ones(m-1,1),-1);
B=diag(2+k*C-2*sigma*ones(m,1))+diag(sigma*ones(m-1,1),1)...
    +diag(sigma*ones(m-1,1),-1);
lside=l(yb+(0:n)*k); rside=r(yb+(0:n)*k);
w(:,1)=f(xl+(1:m)*h)';
for j=1:n
    w(:,j+1)=A\(B*w(:,j)+sigma*[lside(j)+lside(j+1);zeros(m-2,1);rside(j)+rside(j+1)]);
```

```
end
w=[lside;w;rside]';
z=w(end,M/2+1)-w(end-1,M/2+1);
end
```

规律：

1.同一时间点内的分布规律为接近边缘人口密度小，接近中间人口密度大的特点；

2.C 会导致人口密度随时间延长的增减，C 较小时，人口密度随时间延长而减小，C 较大时，人口密度随时间延长而增大，从三幅图可以观察到，这个转折点在 C=9 与 C=10 之间，从参考书中，知道这个转折点为 $C=\pi^2$

（2）

```
the minimum of C is 0.098754
```

公式结果为 0.098696，结果很接近，求解成功

hw9_4_2.m

```
clear,clc;
D=1;
xl=0;xr=10;tb=0;tt=10;
h=0.05;k=0.05;
a=0;b=1;TOL=1e-8;
[~,za]=population(D,a,[xl,xr],[tb,tt],h,k);
[~,zb]=population(D,b,[xl,xr],[tb,tt],h,k);
while (b-a)/2>TOL
    c=(a+b)/2;
    [~,zc]=population(D,c,[xl,xr],[tb,tt],h,k);
    if za*zc<0
        b=c;
        zb=zc;
    else
        a=c;
        za=zc;
    end
end
C=(a+b)/2;
fprintf('the minimum of C is %f\n',C);
```

这里的方法很多，其实合理然后算出来接近那个值就是好方法。
具体的还有求导数的，还有求近 100 次的结果的平均值看是否变化等。

第五题

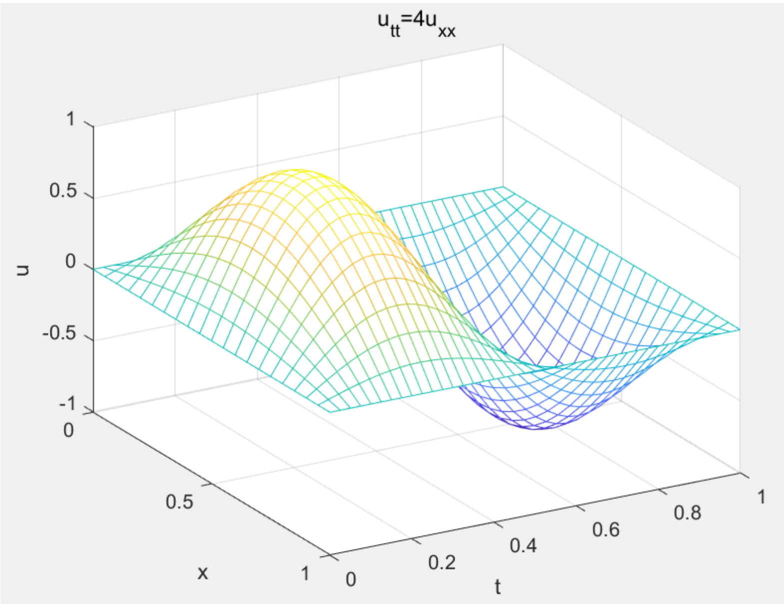
用 Finite Difference Method 求解以下方程

$$\begin{cases} u_{tt} = 4u_{xx}, & 0 \le x \le 1, 0 \le t \le 1 \\ u(x,0) = 0, & 0 \le x \le 1 \\ u_t(x,0) = 2\pi \sin \pi x, & 0 \le x \le 1 \\ u(0,t) = 0, & 0 \le t \le 1 \\ u(1,t) = 0, & 0 \le t \le 1 \end{cases}$$

（1）取 $h = 0.05$ 和满足 CFL condition 的 k，画出近似解，要求给出代码和图像。

（2）已知其解析解为 $u(x,t) = \sin \pi x \sin 2\pi t$，在 $(x,t) = (\frac{1}{4}, \frac{3}{4})$ 处作一张准确值、近似值及误差的表，取步长 $h = ck = 2^{-p}(p = 4,\dots,8)$。

（1）CFL condition：$\sigma = \frac{ck}{h} \le 1 \Rightarrow k \le \frac{h}{c} = 0.025$，取 k=0.025



hw9_5_1.m

```
clear,clc;
c=2;
f=@(x) 0*x;
g=@(x) 2*pi*sin(pi*x);
l=@(t) 0*t;
r=@(t) 0*t;
xl=0;xr=1;tb=0;tt=1; h=0.05; k=0.025;
w=wave(c,f,g,l,r,[xl,xr],[tb,tt],h,k);
x=xl:h:xr;t=tb:k:tt;
mesh(x,t,w);
title('u_t_t=4u_x_x'); xlabel 'x'; ylabel 't';zlabel 'u';
view(60,30);axis([xl xr tb tt -inf inf]);
```

wave.m

```
function w=wave(c,f,g,l,r,x,y,h,k)
xl=x(1);xr=x(2);yb=y(1);yt=y(2);
M=(xr-xl)/h; N=(yt-yb)/k; m=M-1; n=N;
sigma=c*k/h;
A=diag(2-2*sigma^2*ones(m,1))+diag(sigma^2*ones(m-1,1),1)...
    +diag(sigma^2*ones(m-1,1),-1);
lside=l(yb+(0:n)*k); rside=r(yb+(0:n)*k);
w(:,1)=f(xl+(1:m)*h)';
w(:,2)=0.5*A*w(:,1)+k*g(xl+(1:m)*h)'+0.5*sigma^2*[lside(1);zeros(m-2,1);rside(1)];
for j=2:n
    w(:,j+1)=A*w(:,j)-w(:,j-1)+sigma^2*[lside(j);zeros(m-2,1);rside(j)];
end
w=[lside;w;rside]';
end
```

（2）

h	k	exact value	approximate value	error
0.06250000	0.031250000	-0.707107	-0.711671	0.004564
0.03125000	0.015625000	-0.707107	-0.708244	0.001137
0.01562500	0.007812500	-0.707107	-0.707391	0.000284
0.00781250	0.003906250	-0.707107	-0.707178	0.000071
0.00390625	0.001953125	-0.707107	-0.707125	0.000018

hw9_5_2.m

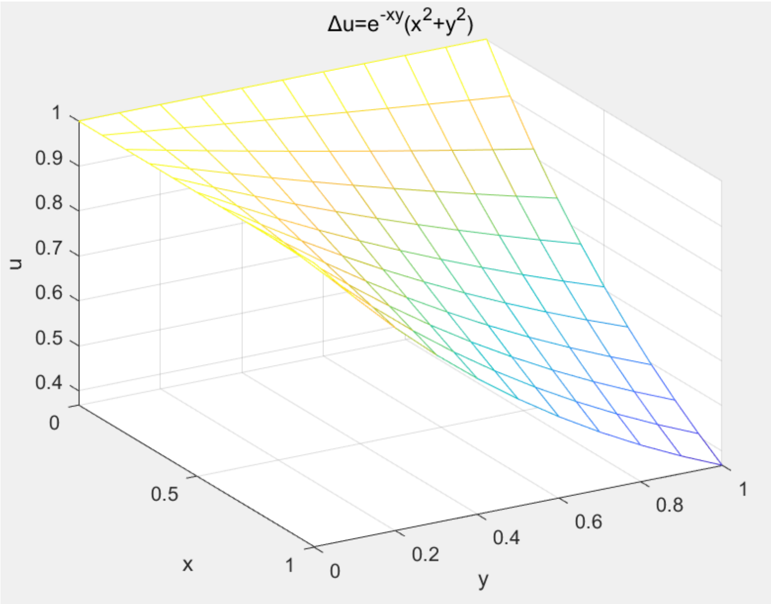
```
clear,clc;
x=1/4;t=3/4;
y_exact=sin(pi*x)*sin(2*pi*t);
c=2;
f=@(x) 0*x;
g=@(x) 2*pi*sin(pi*x);
l=@(t) 0*t;
r=@(t) 0*t;
xl=0;xr=1;tb=0;tt=1;
h=2.^-(4:8);
k=h/c;
fprintf('h\t\t\t k\t\t\t\t exact value\t approximate value\t error\n');
for i=1:length(h)
    w=wave(c,f,g,l,r,[xl,xr],[tb,tt],h(i),k(i));
    w_ap=w(1+round(3/4/k(i)),1+round(1/4/h(i)));
    fprintf('%.8f\t %.9f\t %f\t\t %f\t\t\t %f\n',h(i),k(i),y_exact,w_ap,abs(w_ap-y_exact));
end
```

第六题

用 Finite Difference Method 求解下列方程

$$\begin{cases} \Delta u = e^{-xy}(x^2+y^2), & 0\le x\le 1, 0\le y\le 1 \\ u(x,0)=1, & 0\le x\le 1 \\ u(x,1)=e^{-x}, & 0\le x\le 1 \\ u(0,y)=1, & 0\le y\le 1 \\ u(1,y)=e^{-y}, & 0\le y\le 1 \end{cases}$$

取步长为 $h=k=0.1$，画出近似解。要求给出代码和图像。



hw9_6.m

```
clear,clc;
f=@(x,y) exp(-x.*y).*(x.^2+y.^2);
g1=@(x) 1;%yb
g2=@(x) exp(-x);%yt
g3=@(y) 1;%xl
g4=@(y) exp(-y);%xr
h=0.1;k=0.1;
xl=0;xr=1;yb=0;yt=1;
w=poisson(f,g1,g2,g3,g4,[xl,xr],[yb,yt],h,k);
x=xl:h:xr;y=yb:k:yt;
mesh(x,y,w);
title('△u=e^-^x^y(x^2+y^2)');
xlabel 'x'; ylabel 'y';zlabel 'u';
view(60,30);axis([xl xr yb yt -inf inf]);
```

poisson.m

```
function w=poisson(f,g1,g2,g3,g4,x_interval,y_interval,h,k)
xl=x_interval(1);xr=x_interval(2);yb=y_interval(1);yt=y_interval(2);
```

```
M=(xr-xl)/h; N=(yt-yb)/k; m=M+1; n=N+1;mn=m*n;
h2=h^2;k2=k^2;
A=zeros(mn,mn);b=zeros(mn,1);
x=xl:h:xr;y=yb:k:yt;
for i=2:m-1
    for j=2:m-1
        A(i+(j-1)*m,i+(j-1)*m)=-2/h2-2/k2;
        A(i+(j-1)*m,i+1+(j-1)*m)=1/h2;
        A(i+(j-1)*m,i-1+(j-1)*m)=1/h2;
        A(i+(j-1)*m,i+j*m)=1/k2;
        A(i+(j-1)*m,i+(j-2)*m)=1/k2;
        b(i+(j-1)*m)=f(x(i),y(j));
    end
end
for i=1:m % bottom and top boundary points
    j=1;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g1(x(i));
    j=n;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g2(x(i));
end
for j=2:n-1 % left and right boundary points
    i=1;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g3(y(j));
    i=m;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g4(y(j));
end
v=A\b;
w=reshape(v,m,n);
w=w';
end
```