微分方程

热方程解的三维图像

一、一维热方程的初边值(混合)问题

$$(*)\begin{cases} u_t = u_{xx}, \ x \in (0,1), t \in (0, 0.2] \\ u|_{t=0} = \sin(\pi x), \ x \in [0,1] \\ u|_{x=0} = u|_{x=1} = 0, \ t \in [0, 0.2] \end{cases}$$

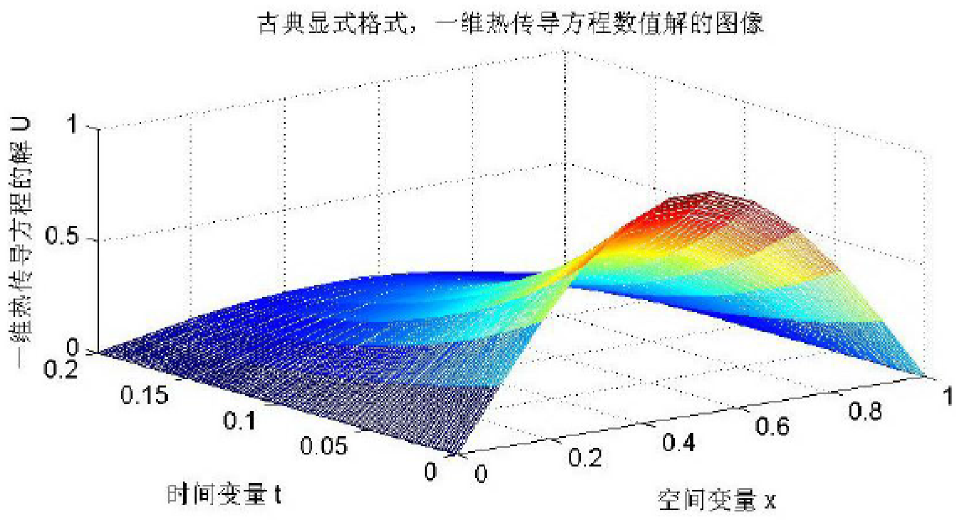
的(精确)解为

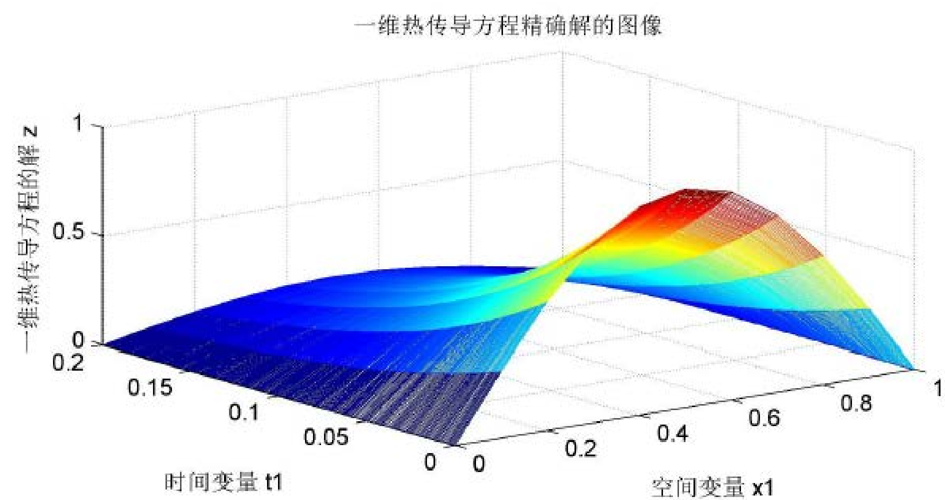
$$\boxed{u(x,t) = e^{-\pi^2 t}\sin(\pi x).}$$

注：(*)的解的最大最小值分别是1, 0, 也可由最大值原理得到。问题(*)一般通过分离变量法求解。

利用Matlab求解一维热方程初边值问题(*)

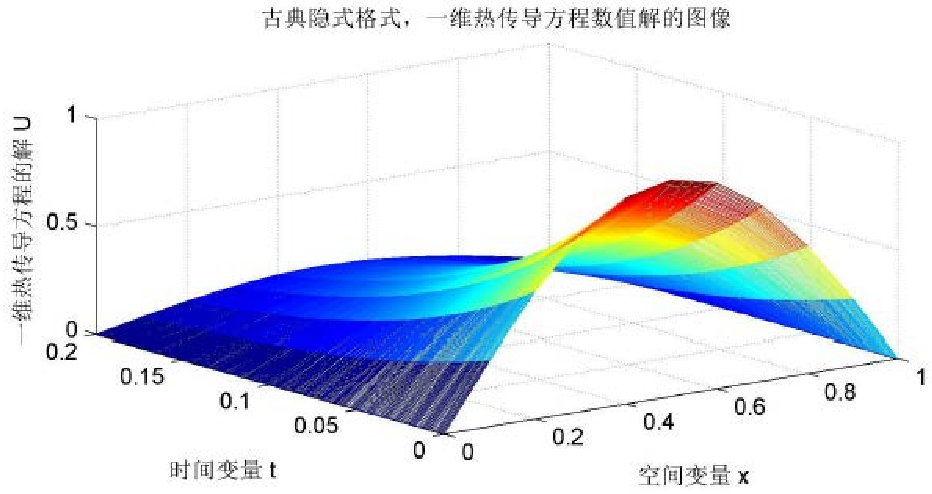
1. 古典显式格式

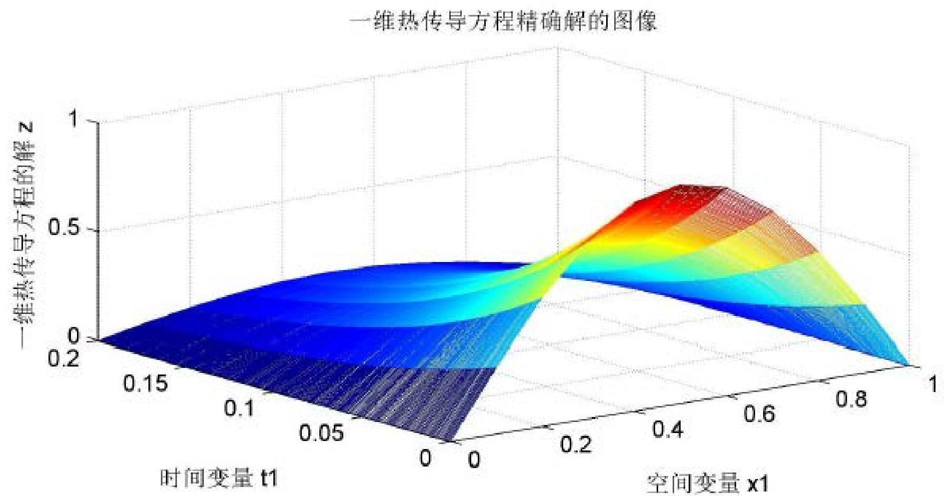




利用Matlab求解一维热方程初边值问题(*)

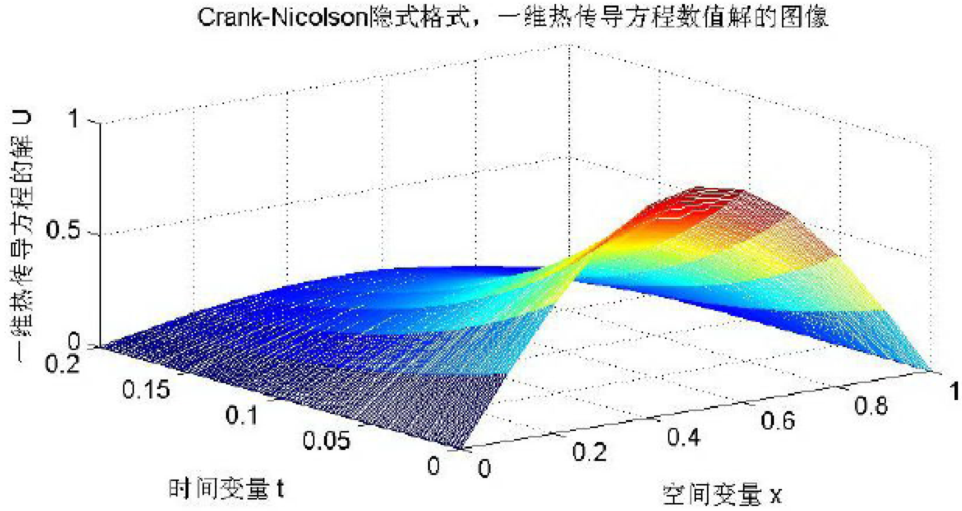
2. 古典隐式格式

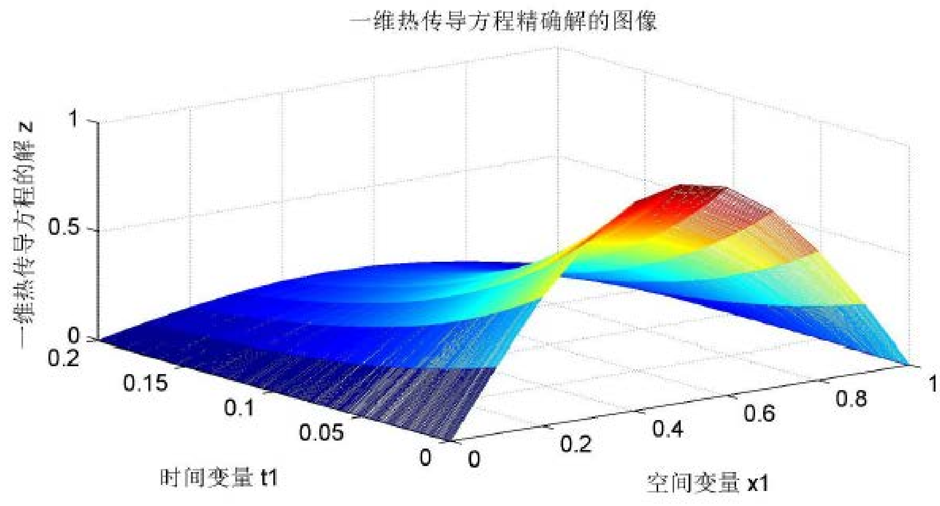




利用Matlab求解一维热方程初边值问题(*)

3. Crank-Nicolson隐式格式





二、一维热方程的初值(Cauchy)问题

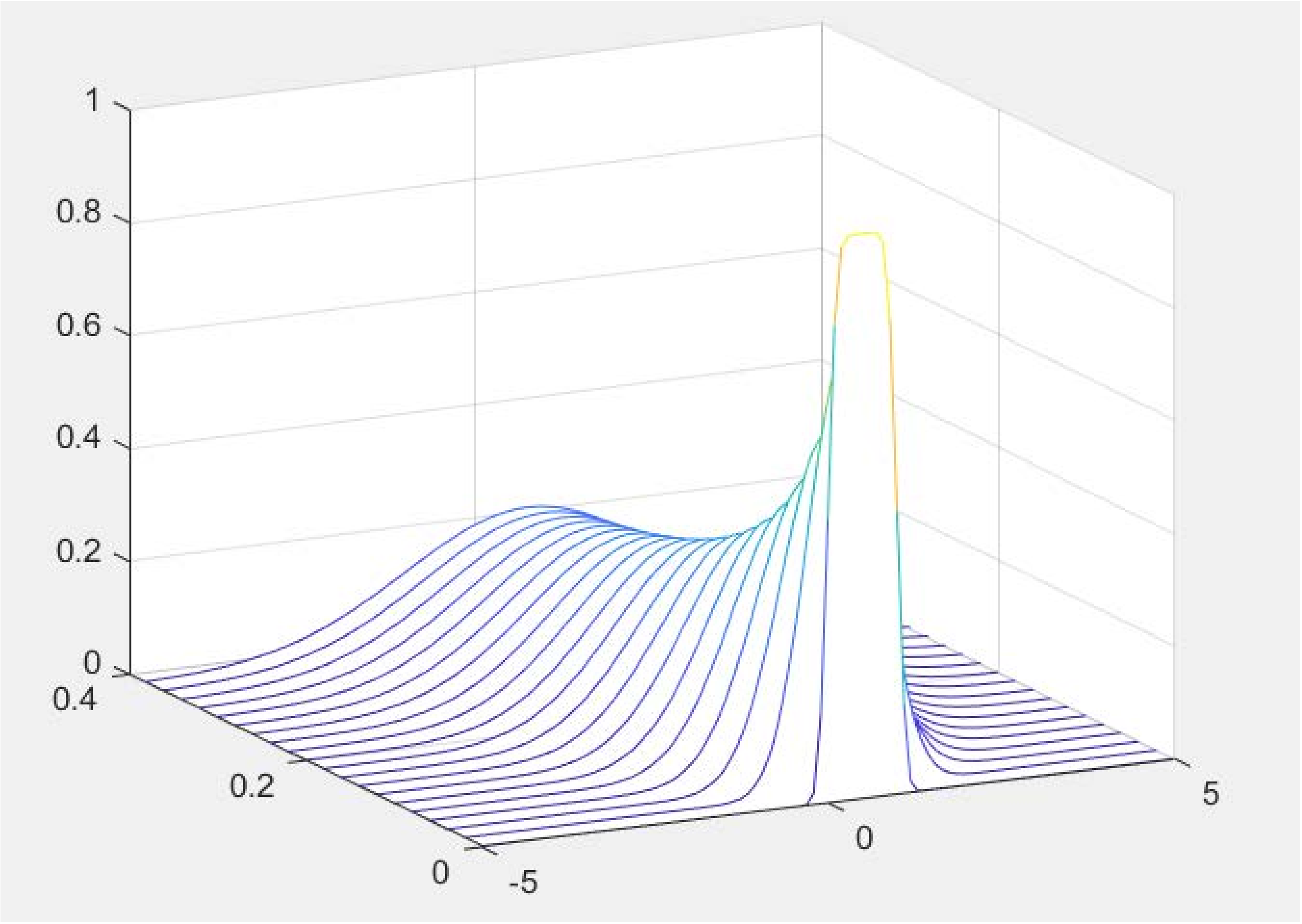
$$
(**)\begin{cases} u_t = a^2 u_{xx}, \ x \in \mathbb{R}, t > 0 \\ u|_{t=0} = \varphi(x) = \begin{cases} 1, & 0 < x < 1 \\ 0, & x \le 0 \text{或} x \ge 1 \end{cases} \end{cases}
$$

的(精确)解为

$$
\boxed{u(x,t) = \frac{1}{2a\sqrt{\pi t}} \int_0^1 e^{-\frac{(\xi - x)^2}{4a^2 t}} d\xi.}
$$

注：(**)的解经过充分长时间后衰减到零。

利用Matlab求解一维热方程初值问题(**)



附：Matlab代码

1、古典显式格式求解一维热方程

```
function [U x t]=PDEParabolicClassicalExplicit(uX,uT,phi,psi1,psi2,M,N,C)
%古典显式格式求解抛物型偏微分方程
%[U x t]=PDEParabolicClassicalExplicit(uX,uT,phi,psi1,psi2,M,N,C)
%
%方程：u_t=C*u_xx 0 <= x <= uX,0 <= t <= uT
%初值条件：u(x,0)=phi(x)
%边值条件：u(0,t)=psi1(t), u(uX,t)=psi2(t)
%
%输出参数：U -解矩阵，第一行表示初值，第一列和最后一列表示边值，第二行表示第 2 层……
%         x -空间变量
%         t -时间变量
%输入参数：uX -空间变量 x 的取值上限
%         uT -时间变量 t 的取值上限
%         phi -初值条件，定义为内联函数
%         psi1 -边值条件，定义为内联函数
%         psi2 -边值条件，定义为内联函数
%         M -沿 x 轴的等分区间数
%         N -沿 t 轴的等分区间数
%         C -系数，默认情况下 C=1
%
```

```
%应用举例:
%uX=1;uT=0.2;M=15;N=100;C=1;
%phi=inline('sin(pi*x)');psi1=inline('0');psi2=inline('0');
%[U x t]=PDEParabolicClassicalExplicit(uX,uT,phi,psi1,psi2,M,N,C);

%设置参数 C 的默认值
if nargin==7
    C=1;
end

%计算步长
dx=uX/M;%x 的步长
dt=uT/N;%t 的步长
```

```
x=(0:M)*dx;
t=(0:N)*dt;

r=C*dt/dx/dx;%步长比
r1=1-2*r;

if r > 0.5
    disp('r > 0.5,不稳定')
end



%计算初值和边值
U=zeros(M+1,N+1);
for  i=1:M+1
    U(i,1)=phi(x(i));
end
for  j=1:N+1
    U(1,j)=psi1(t(j));
    U(M+1,j)=psi2(t(j));
end
```

```
%逐层求解
for  j=1:N
     for  i=2:M
         U(i,j+1)=r*U(i-1,j)+r1*U(i,j)+r*U(i+1,j);
     end
end

U=U';

%作出图形
mesh(x,t,U);
title('古典显式格式，一维热传导方程的解的图像')
xlabel('空间变量  x')
ylabel('时间变量  t')
zlabel('一维热传导方程的解  U')

return;
```

2、古典隐式格式求解一维热传导方程

```
function [U x t]=PDEParabolicClassicalImplicit(uX,uT,phi,psi1,psi2,M,N,C)
%古典隐式格式求解抛物型偏微分方程
%[U x t]=PDEParabolicClassicalImplicit(uX,uT,phi,psi1,psi2,M,N,C)
%
%方程：u_t=C*u_xx 0 <= x <= uX,0 <= t <= uT
%初值条件：u(x,0)=phi(x)
%边值条件：u(0,t)=psi1(t), u(uX,t)=psi2(t)
%
%输出参数：U -解矩阵，第一行表示初值，第一列和最后一列表示边值，第二行表示第 2 层……
%         x -空间变量
%         t -时间变量
%输入参数：uX -空间变量 x 的取值上限
```

```
%        uT  -时间变量 t 的取值上限
%        phi  -初值条件，定义为内联函数
%        psi1  -边值条件，定义为内联函数
%        psi2  -边值条件，定义为内联函数
%        M  -沿 x 轴的等分区间数
%        N  -沿 t 轴的等分区间数
%        C  -系数，默认情况下 C=1
%
%应用举例：
%uX=1;uT=0.2;M=50;N=50;C=1;
%phi=inline('sin(pi*x)');psi1=inline('0');psi2=inline('0');
%[U  x  t]=PDEParabolicClassicalImplicit(uX,uT,phi,psi1,psi2,M,N,C);


%设置参数 C 的默认值
if nargin==7
    C=1;
end

%计算步长
dx=uX/M;%x 的步长
dt=uT/N;%t 的步长
```

```
x=(0:M)*dx;
t=(0:N)*dt;

r=C*dt/dx/dx;%步长比

Diag=zeros(1,M-1);%矩阵的对角线元素
Low=zeros(1,M-2);%矩阵的下对角线元素
Up=zeros(1,M-2);%矩阵的上对角线元素
for  i=1:M-2
     Diag(i)=1+2*r;
     Low(i)=-r;
     Up(i)=-r;
end
Diag(M-1)=1+2*r;

%计算初值和边值
U=zeros(M+1,N+1);
for  i=1:M+1
     U(i,1)=phi(x(i));
end
for  j=1:N+1
```

```
    U(1,j)=psi1(t(j));
    U(M+1,j)=psi2(t(j));
end

%逐层求解，需要使用追赶法（调用函数 EqtsForwardAndBackward）
for  j=1:N
    b1=zeros(M-1,1);
    b1(1)=r*U(1,j+1);
    b1(M-1)=r*U(M+1,j+1);
    b=U(2:M,j)+b1;
    U(2:M,j+1)=EqtsForwardAndBackward(Low,Diag,Up,b);
end

U=U';

%作出图形
mesh(x,t,U);
title('古典隐式格式，一维热传导方程的解的图像')
xlabel('空间变量  x')
ylabel('时间变量  t')
zlabel('一维热传导方程的解  U')

return;
```

```
function  x=EqtsForwardAndBackward(L,D,U,b)
%追赶法求解三对角线性方程组 Ax=b
%x=EqtsForwardAndBackward(L,D,U,b)
%x:三对角线性方程组的解
%L:三对角矩阵的下对角线，行向量
%D:三对角矩阵的对角线，行向量
%U:三对角矩阵的上对角线，行向量
%b:线性方程组 Ax=b 中的 b，列向量
%
%应用举例:
%L=[-1  -2  -3];D=[2  3  4  5];U=[-1  -2  -3];b=[6  1  -2  1]';
%x=EqtsForwardAndBackward(L,D,U,b)
%检查参数的输入是否正确
n=length(D);m=length(b);
n1=length(L);n2=length(U);
if n-n1  ~=  1  ||  n-n2  ~=  1  ||  n  ~=  m
    disp('输入参数有误！')
    x=' ';
    return;
end
```

```
%追的过程
for  i=2:n
      L(i-1)=L(i-1)/D(i-1);
      D(i)=D(i)-L(i-1)*U(i-1);
end
x=zeros(n,1);
x(1)=b(1);
for  i=2:n
      x(i)=b(i)-L(i-1)*x(i-1);
end

%赶的过程
x(n)=x(n)/D(n);
for  i=n-1:-1:1
      x(i)=(x(i)-U(i)*x(i+1))/D(i);
end

return;
```

3、Crank-Nicolson隐式格式求解一维热方程

```
function [U x t]=PDEParabolicCN(uX,uT,phi,psi1,psi2,M,N)
%Crank-Nicolson 隐式格式求解抛物型偏微分方程
%[U x t]=PDEParabolicCN(uX,uT,phi,psi1,psi2,M,N)
%
%方程：u_t=u_xx  0 <= x <= uX,0 <= t <= uT
%初值条件：u(x,0)=phi(x)
%边值条件：u(0,t)=psi1(t), u(uX,t)=psi2(t)
%
%输出参数：U -解矩阵，第一行表示初值，第一列和最后一列表示边值，第二行表示第 2 层……
%          x -空间变量
%          t -时间变量
%输入参数：uX -空间变量 x 的取值上限
%          uT -时间变量 t 的取值上限
%          phi -初值条件，定义为内联函数
%          psi1 -边值条件，定义为内联函数
%          psi2 -边值条件，定义为内联函数
%          M -沿 x 轴的等分区间数
%          N -沿 t 轴的等分区间数
%
```

```
%应用举例:
%uX=1;uT=0.2;M=50;N=50;
%phi=inline('sin(pi*x)');psi1=inline('0');psi2=inline('0');
%[U x t]=PDEParabolicCN(uX,uT,phi,psi1,psi2,M,N);
%计算步长
dx=uX/M;%x 的步长
dt=uT/N;%t 的步长

x=(0:M)*dx;
t=(0:N)*dt;

r=dt/dx/dx;%步长比

Diag=zeros(1,M-1);%矩阵的对角线元素
Low=zeros(1,M-2);%矩阵的下对角线元素
Up=zeros(1,M-2);%矩阵的上对角线元素
for  i=1:M-2
    Diag(i)=1+r;
    Low(i)=-r/2;
    Up(i)=-r/2;
end
Diag(M-1)=1+r;
```

```
%计算初值和边值
U=zeros(M+1,N+1);
for  i=1:M+1
     U(i,1)=phi(x(i));
end
for  j=1:N+1
     U(1,j)=psi1(t(j));
     U(M+1,j)=psi2(t(j));
end

B=zeros(M-1,M-1);
for  i=1:M-2
     B(i,i)=1-r;
     B(i,i+1)=r/2;
     B(i+1,i)=r/2;
end
B(M-1,M-1)=1-r;

%逐层求解，需要使用追赶法（调用函数 EqtsForwardAndBackward）
for  j=1:N
     b1=zeros(M-1,1);
```

```
    b1(1)=r*(U(1,j+1)+U(1,j))/2;
    b1(M-1)=r*(U(M+1,j+1)+U(M+1,j))/2;
    b=B*U(2:M,j)+b1;
    U(2:M,j+1)=EqtsForwardAndBackward(Low,Diag,Up,b);
end

U=U';

%作出图形
mesh(x,t,U);
title('Crank-Nicolson 隐式格式，一维热传导方程的解的图像')
xlabel('空间变量  x')
ylabel('时间变量  t')
zlabel('一维热传导方程的解  U')

return;
```

4、一维热方程热方程初值问题

```
clear; a=2;u0=1;  dpsi=0.1;
x=-5:0.1:5;  t=(0.001:0.005:.401);
psi=0:0.1:1;
[X,T,Psi]=meshgrid(x,t,psi);
F=u0./(2*a*sqrt(pi*T)).*exp(-(X-
Psi).^2./((2*a)^2*T));
u=dpsi*trapz(F,3); w=squeeze(u(1,:));
figure(1);
h=plot(x,w,'linewidth',3);axis([-
5,5,0,1.1]);
for n=2:length(t);   w=squeeze(u(n,:));
    set(h,'ydata',w); drawnow;
pause(0.001);
end;
figure(2);
waterfall(x,t(1:4:81),u(1:4:81,:));
axis([-5 5 0 .4 0 1]);
```