SF1544

Övning 6

This övning

- numerical discretization of PDE
- Monte Carlo method
- Nonlinear Least square problem

Heat equation

Problem

Heat equation

$$\frac{\partial u}{\partial t} = \sigma \frac{\partial^2 u}{\partial x^2}$$
$$u(0,x) = g(x)$$
$$u(t,0) = h(t)$$
$$u(t,1) = r(t)$$

where the domain is

$$0 \leq t \leq 1$$
$$0 \leq x \leq 1$$

Heat equation: discretization

Let consider the discretization

$$0 = t_1 < t_2 < \cdots < t_N = 1$$
$$0 = x_1 < x_2 < \cdots < x_N = 1$$

Heat equation: discretization

Let consider the discretization

$$0 = t_1 < t_2 < \cdots < t_N = 1$$
$$0 = x_1 < x_2 < \cdots < x_N = 1$$

such that

$$t_{j+1} = t_j + \Delta t$$
$$x_{i+1} = x_i + \Delta x$$

Heat equation: discretization

$$\frac{\partial u}{\partial t}(t_j, x_i) = \sigma \frac{\partial^2 u}{\partial x^2}(t_j, x_i)$$

Heat equation: discretization

Notation

$$u_i^j \approx u(t_j, x_i)$$

Heat equation: discretization

Notation

$$u_i^j \approx u(t_j, x_i)$$

Heat equation: discretization

$$\frac{\partial u}{\partial t}(t_j, x_i) = \sigma \frac{\partial^2 u}{\partial x^2}(t_j, x_i)$$

Heat equation: discretization

Notation

$$u_i^j \approx u(t_j, x_i)$$

Heat equation: discretization

$$\frac{\partial u}{\partial t}(t_j, x_i) = \sigma \frac{\partial^2 u}{\partial x^2}(t_j, x_i)$$

Heat equation: discretization interior points

$$\frac{u_i^{j+1} - u_i^j}{\Delta t} = \sigma \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{\Delta x^2}$$

Heat equation: discretization

Heat equation: discretization interior points

$$u_i^{j+1} = u_i^j - \frac{\sigma \Delta t}{\Delta x^2}\left(u_{i+1}^j - 2u_i^j + u_{i-1}^j\right)$$

Heat equation: discretization

Heat equation: discretization interior points

$$u_i^{j+1} = u_i^j - \frac{\sigma \Delta t}{\Delta x^2}\left(u_{i+1}^j - 2u_i^j + u_{i-1}^j\right)$$

Heat equation: discretization intial and boundary conditions

$$u_i^1 = g(x_i)$$
$$u_1^j = h(t_j)$$
$$u_M^j = r(t_j)$$

Heat equation: discretization (conclusion)

Heat equation

$$\frac{\partial u}{\partial t} = \sigma \frac{\partial^2 u}{\partial x^2}$$

$$u(0,x) = g(x) \qquad u(t,0) = h(t) \qquad u(t,1) = r(t)$$

Heat equation: discretization interior points

$$u_i^{j+1} = u_i^j - \frac{\sigma \Delta t}{\Delta x^2}\left(u_{i+1}^j - 2u_i^j + u_{i-1}^j\right) \quad \text{i=2, \ldots, M-1, j=1, \ldots, N-1}$$

$$u_i^1 = g(x_i), \quad u_1^j = h(t_j), \quad u_M^j = r(t_j) \qquad \text{i=1, \ldots, M, j=1, \ldots, N}$$

Heat equation: example

Heat equation

$$\frac{\partial u}{\partial t} = \frac{1}{\pi}\frac{\partial^2 u}{\partial x^2}$$
$$u(0,x) = \sin(\pi x)$$
$$u(t,0) = 0$$
$$u(t,1) = 0$$

Matlab implementation: complete it!

```
close all; clear all; clc

M=100;  N=M^2;
u=zeros(N,M);
t=linspace(0,1,N);  dt=t(2)-t(1);
x=???????????????;  dx=x(2)-x(1);

u(1,:)  = ???????;
u(:,1)  = ?;
u(:,end)= 0;

K=(dt/dx^2)*?????;
for j=1:N-1
    for i=2:M-1
        u(j+1,i)=u(j,i)+K*(u(j,i+1)-2*u(j,i)+u(j,i-1));
    end
end
imagesc(t,x,abs(u))
```

Matlab implementation: solution

```
close all; clear all; clc

M=100;  N=M^2;
u=zeros(N,M);
t=linspace(0,1,N);  dt=t(2)-t(1);
x=linspace(0,1,M);  dx=x(2)-x(1);

u(1,:)   = sin(pi*x);
u(:,1)   = 0;
u(:,end)= 0;

K=(dt/dx^2)*(1/pi);
for j=1:N-1
    for i=2:M-1
        u(j+1,i)=u(j,i)+K*(u(j,i+1)-2*u(j,i)+u(j,i-1));
    end
end
imagesc(t,x,abs(u))
```

MATLAB DEMO

Monte Carlo method

Exercise

Compute the area included between the two parabolas

$$y = x^2 - x + \frac{1}{2}$$
$$y = -x^2 + x + \frac{1}{2}$$

for $0 \leq x \leq 1$

Monte Carlo method

Exercise

Compute the area included between the two parabolas

$$y = x^2 - x + \frac{1}{2}$$
$$y = -x^2 + x + \frac{1}{2}$$

for $0 \leq x \leq 1$

IDEA: generate N random points $(x_i, y_i) \in [0,1] \times [0,1]$

Monte Carlo method

Exercise

Compute the area included between the two parabolas

$$y = x^2 - x + \frac{1}{2}$$
$$y = -x^2 + x + \frac{1}{2}$$

for $0 \leq x \leq 1$

IDEA: generate N random points $(x_i, y_i) \in [0,1] \times [0,1]$

$$\frac{\#\text{ points between the parabolas}}{N} = \frac{\#\text{ area between the parabolas}}{\#\text{ total area}}$$

Monte Carlo method

Exercise

Compute the area included between the two parabolas

$$y = x^2 - x + \frac{1}{2}$$
$$y = -x^2 + x + \frac{1}{2}$$

for $0 \leq x \leq 1$

IDEA: generate N random points $(x_i, y_i) \in [0,1] \times [0,1]$

$$\frac{\#\text{ points between the parabolas}}{N} = \frac{\#\text{ area between the parabolas}}{\#\text{ total area}}$$

$\#$ total area $=1$

MATLAB DEMO

Matlab implementation: solution

```
close all
clear all
clc

p1=@(x) x.^2-x+1/2;
p2=@(x) -x.^2+x+1/2;
x=linspace(0,1,100);
plot(x,p1(x),'-k'); hold on
plot(x,p2(x),'-r');

N=1e4;
j=0;
for i=1:N
    x=rand; y=rand;
    if p1(x)<y && p2(x)>y
        j=j+1;
    end
end
Area=j/N
```

Nonlinear Least Square problem

4.25 Följande tabell visar den uppmätta positionen y vid olika tidpunkter för en massa i ett dämpat svängningsförlopp: $Y(t) = -0.17\,e^{-bt}(\cos\omega t + \frac{b}{\omega}\sin\omega t)$.

t	0.8	1.7	2.5	3.3	4.1
y	0.12	-0.09	0.06	-0.05	0.032

Utnyttja de fem mätningarna för att bestämma parametrarna b och ω så bra som möjligt. Man vet att mätningarna gjorts nära max- och minlägena, vilket leder till följande goda startgissningar för parametrarna: $\omega = \frac{2\pi}{t_3 - t_1}$ och $b = \frac{\omega}{2\pi}(\ln y_1 - \ln y_3)$.

Matlab implementation: solution

```
close all
clear all
clc

t=[0.8 1.7 2.5 3.3 4.1]'; y=[0.12 -0.09 0.06 -0.05 0.032]';
a=-0.17; s=0.001;
w=2*pi/(t(3)-t(1)); b=(log(y(1))-log(y(3)))*w/(2*pi);
c=[w b]';
for iter=1:4
    F=a*exp(-b*t).*(cos(w*t)+b/w*sin(w*t));      f=F-y;
    fnorm=norm(f)

    ws=w+s; Fw=a*exp(-b*t).*(cos(ws*t)+b/ws*sin(ws*t));
    bs=b+s; Fb=a*exp(-bs*t).*(cos(w*t)+bs/w*sin(w*t));
    J=[(Fw-F)/s (Fb-F)/s];

    dc=-J\f;     c=c+dc; w=c(1); b=c(2);
end
```