MA2501 Numeriske metoder

Øving 11

Veiledning: 4/5.

Oppgave 1

Gitt startverdiproblemet

$$x'''(t) = e^{x(t)}$$

$$x(0) = 1, \;\; x'(0) = 1, \;\; x''(0) = 1.$$

a) Overfør problemet til et system av første ordens differensialligninger.

b) Finn tilnærmelser til $x(0.1)$ og $x(0.2)$ ved hjelp av en 2.ordens RK-metode. Bruk skrittlengden $h = 0.1$.

Oppgave 2

Løsningen til den partielle differensialligningen (Poissons ligning)

$$u_{xx} + u_{yy} = -1$$

i et område D, der $u(x, y)$ er gitt på randen til D, skal tilnærmes ved en differensemetode. Området D er gitt ved

$$D = \{(x, y) \mid 0 < x < 1, 0 < y < 1 - x\} ,$$

og $u(x, y) = 1$ på randen. Vi bruker en skrittlengde $h = 0.25$, og lar u_{ij} være tilnærmelsen til $u(i \cdot h, j \cdot h)$. Se figuren på neste side.

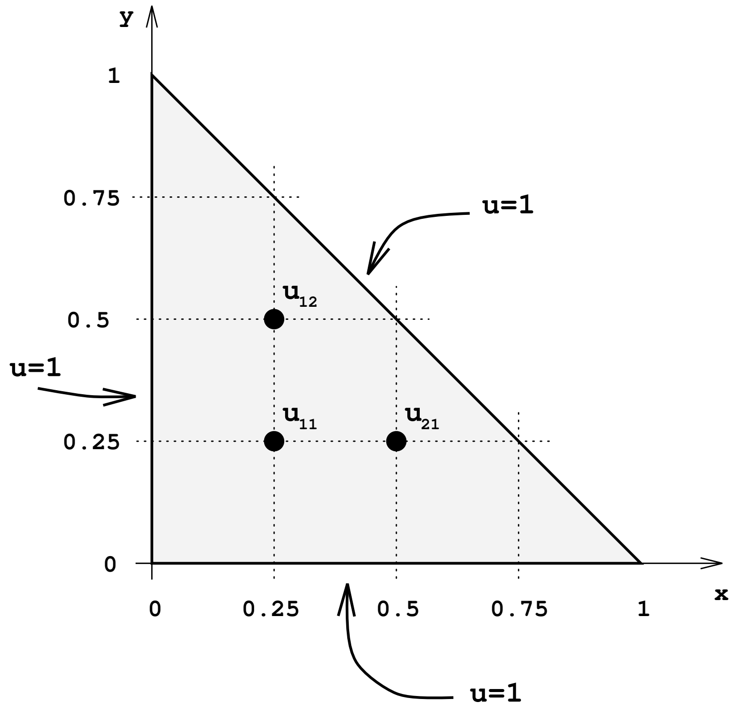
a) Finn de tre ligningene som bestemmer u_{11}, u_{12} og u_{21} når sentraldifferenser benyttes for å tilnærme de deriverte.

Ligningssystemet i **a)** kan skrives på formen

$$A\vec{u} = \vec{b}$$

der A er en 3×3 matrise, og $\vec{u}$ og $\vec{b}$ er vektorer.

b) La $\vec{u} = (u_{11}, u_{21}, u_{12})^T$ (naturlig ordning av de ukjente i vektoren $\vec{u}$). Finn A og $\vec{b}$. Løs ligningssystemet og finn u_{11}, u_{21} og u_{12}.



Oppgave 3

Løs Burgers' ligning

$$\begin{cases} u_t + \left(\dfrac{u^2}{2}\right)_x & = & \mu u_{xx} \\ \\ u(x,0) & = & 1.5 \cdot x \cdot (1-x)^2 \\ u(0,t) & = & u(1,t) = 0 \end{cases}$$

på området $0 \leq x \leq 1$ og $0 \leq t \leq 6$. Burgers' ligning ble introdusert i 1948 som en matematisk modell som skulle beskrive turbulens. Senere er en kommet fram til at modellen er for enkel til å gi en full beskrivelse av fenomenet. Men fortsatt brukes den som en testligning for numeriske skjemaer som er tenkt brukt til å løse Euler's og Navier-Stokes' ligninger.

Bruk diskretiseringen

$$\frac{u_{i,j+1} - u_{i,j}}{k} + \frac{u^2_{i+1,j+1} - u^2_{i-1,j+1}}{4h} = \mu \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2}, \qquad (1)$$
$$i = 1, 2, \cdots, n, \quad j = 0, 1, \cdots, m$$

hvor $u_{i,j} \approx u(x_i, t_j)$, $x_i = i \cdot h$ og $t_j = j \cdot k$. k og h er skrittlengdene i hhv. t og x-retningen, og $n = 1.0/h$, $m = 6.0/k$. Husk at $u_{0,j}$, $u_{n,j}$ og $u_{i,0}$ er gitt fra rand- og initialbetingelsene.

Diskretiseringen er implisitt, dvs. at et ikke-lineært ligningssystem må løses mhp. $u_{i,j+1}$, $i = 1, \cdots, n-1$ for hvert tidsskritt. Dette kan løses vha. Newton's metode, med verdien fra forrige tidsskritt som startverdi

Oppgaven går ut på lage et MATLAB-program som løser Burgers' ligning vha. diskretiseringen gitt i (1), og som deretter plotter løsningen. Men før du for alvor går igang med programmeringen, så

a) sjekk at du virkelig har forstått hvordan man kommer fram til diskretiseringen (1).

b) skriv opp det ulinære ligningssystemet du må løse for for hvert tidskritt. Hva er Jacobimatrisa for ligningen?

Sett $h = k = 0.01$ og $\mu = 0.001$, og sett igang med programmeringen. Du kan bruke koden på neste side som utgangspunkt (er lagt ut).

Hint: I MATLAB lønner det seg vanligvis å operere direkte på vektorer. Det betyr at f.eks.

```
u(2:n-1) = p*(u(1:n-2)-2*u(2:n-1)+u(3:n)));
```

er det samme som

```
for i=2:n-1
  u(i) = p*(u(i-1)-2*u(i)+u(i+1));
end;
```

men første alternativ går mye raskere.

Lag eventuelt en animasjon av løsningen.

```
% Forberedelser
%-------------
h=0.01;      % Skrittlengder i x- og t- retning.
k=0.01;
xend = 1;
tend = 6;
n = xend/h+1;   % Antall diskretiseringspunkter i hhv. x- og t-
m = tend/k+1;   % retning, randpunktene inkludert.
mu=1.e-3
u = zeros(n,m); % Lager en matrise som etterhvert skal inneholde
                % l{\o}sningen
x=[0:h:xend]';
t=[0:k:tend];
u(:,1) = 1.5*x.*(1-x).^2; % Legger startverdiene i 1. kolonne i u.

% Hovedloekke
%-------------
for j=1:m-1
  u(:,j+1) = u(:,j); % Setter startverdien til Newton-iterasjonene
                     % lik verdien i forrige skritt.
  du = ones(n-2,1);

  % Newton-iterasjoner.
  while norm(du,2) > 1.e-4
%====================================================================
%    Denne biten maa du fylle ut selv. Den best{\aa}r av:
%     Regn ut funksjonen f(u):
%     Finn Jacobi-matrisa :
%     Sett: du = -J\f (l{\o}s ligningssystemet) og sett u(2:n-1,j+1) =
%          u(2:n-1,j+1) + du;
%====================================================================
  end;
end;

% Plotter l{\o}sningen
mesh(x,t,u');
view( -15,35);  % Bestemmer vinkelen du ser plottet fra.
xlabel('x');
ylabel('t');
```