Fö 11, Partiell differentialekvation

Givet modellproblemet för en parabolisk partiell differentialekvation

$$u_t = u_{xx}, \quad 0 \leq x \leq 1, t \geq 0, \quad u(x,0) = sin(\pi x) \quad u(0,t) = 0, u(1,t) = 0$$

Området i x-led diskretiseras enligt

$$x_i = ih, i = 0, 1, 2, ...N, N+1 \quad h = 1/(N+1),\ u_i \approx u(x_i, t)$$

Observera att u_i är en funktion av t definierad på linjen $x = x_i$ i xt-planet.

Rumsderivatan diskretiseras längs de inre linjerna enligt

$$\frac{du_i}{dt} = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \quad i = 1, 2, \ldots N$$

och från randvillkoren får vi $u_0(t) = 0, u_{N+1}(t) = 0$.

Vi har nu följande system av ordinära differentialekvationer

$$\frac{d\mathbf{u}}{\mathbf{dt}} = \mathbf{f}(\mathbf{u}) \quad \text{med } \mathbf{u} = (\mathbf{u_1}, \mathbf{u_2}, ..., \mathbf{u_N})^{\mathbf{T}}$$

där

$$\mathbf{f_i}(\mathbf{u}) = \frac{\mathbf{u_{i+1} - 2u_i + u_{i-1}}}{\mathbf{h^2}} \quad \mathbf{i = 1, 2 \ldots N}$$

vilket också kan skrivas

$$\mathbf{f}(\mathbf{u}) = \mathbf{Au}$$

där A är en $N \times N$ tridiagonal matris med diagonalelementen $-2/h^2$ och elementen just under och över diagonalen $1/h^2$. Vi skriver $A = trid(1/h^2, -2/h^2, 1/h^2)$.

I Matlab kan vi skriva

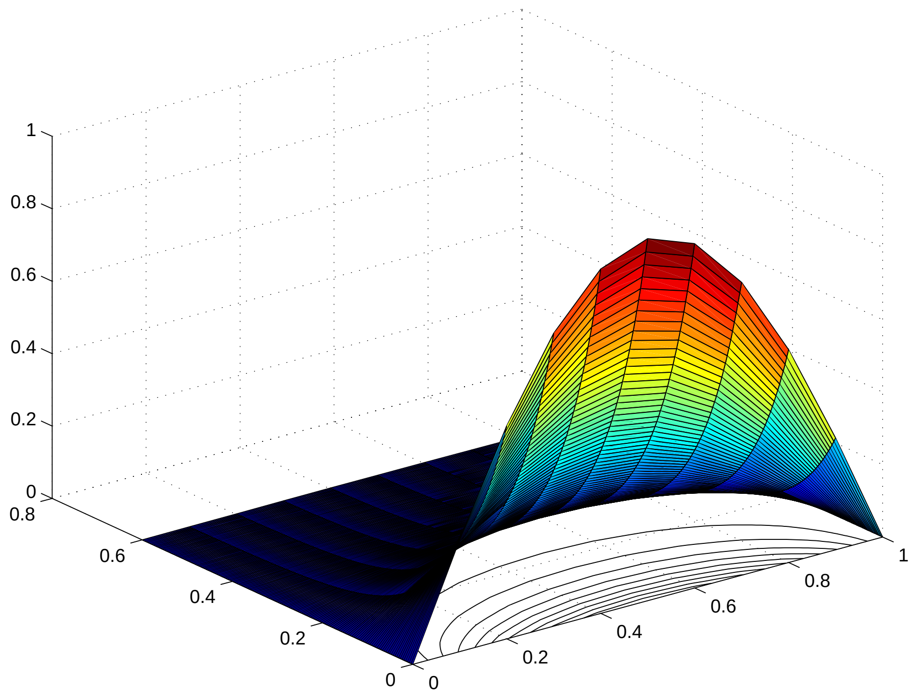
```
function f=fparab(y)
global uL, uR, h
u=[uL; y; uR]; h2=h^2;
f=( u(3:N+2) -2*u(2:N+1) + u(1:N) )/h2;
```

Om vi löser det ordinära differentialekvationssystemet med Eulers metod får vi

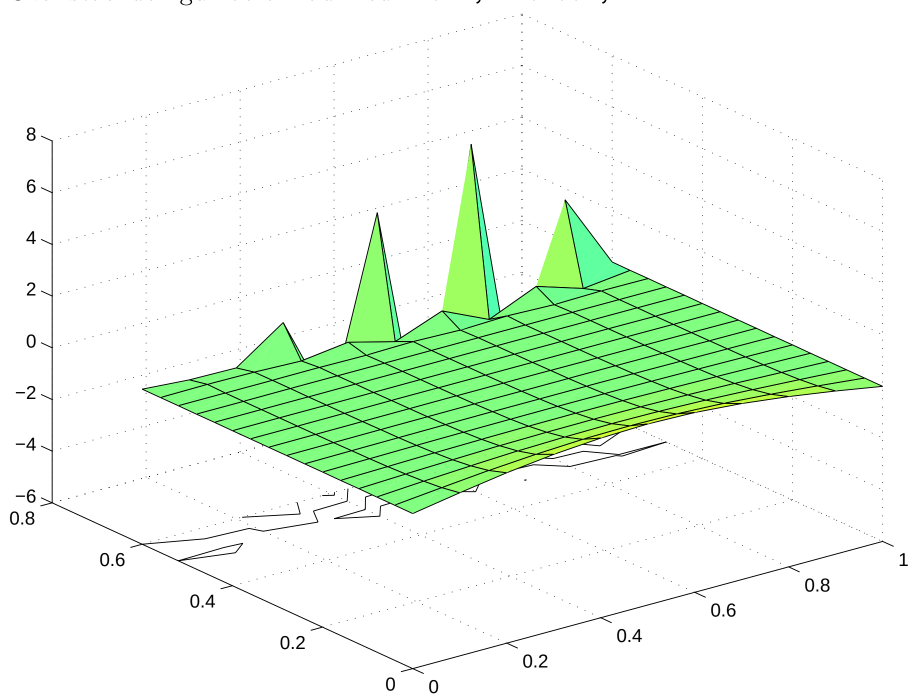
$$\mathbf{u_{j+1}} = \mathbf{u_j} + \mathbf{kf}(\mathbf{u_j}), \qquad \mathbf{j = 0, 1, \ldots}$$

```
clf, clear
global uL uR h
h=0.1; k=0.04; x=0:h:1; t=0:k:0.6;
[X,T]=meshgrid(x,t);
N=length(x); M=length(t); U=[];

u=sin(pi*x(2:N-1))'; uL=0; uR=0;

for j=1:M
  u=u+k*fparab(u); U=[U;[uL u' uR]];
end

surfc(X,T,U)
```

Lösningar med olika h-värden och k-värden finns i figurerna nedan.



Ovanstående figur beräknad med `h=0.1; k=0.004;`



Ovanstående figur beräknad med `h=0.1; k=0.04;` Som synes blir beräkningarna instabila, felen oscillerar och växer med tiden.

Om vi i stället använder Bakänges Euler får vi

$$\mathbf{u_{j+1}} = \mathbf{u_j} + \mathbf{kf}(\mathbf{u_{j+1}}) \quad \text{dvs} \quad \mathbf{u_{j+1}} - \mathbf{kf}(\mathbf{u_{j+1}}) = \mathbf{u_j}$$

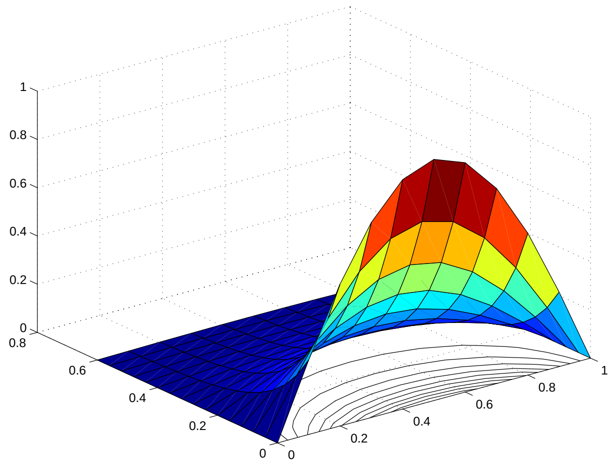
Detta är ett linjärt ekvationssystem med en tridiagonal koefficientmatris

$$I - kA = trid(-\kappa, 1 + 2\kappa, -\kappa)$$

med $\kappa = k/h^2$ att lösas i varje tidssteg. Vi kan också låta datorn beräkna koefficientmatrisen A med `glesjac`.

```
clf, clear
global uL uR h
h=0.1; k=0.04; x=0:h:1; t=0:k:0.6;
[X,T]=meshgrid(x,t);
N=length(x); M=length(t); U=[];
u=sin(pi*x(2:N-1))'; uL=0; uR=0;
A=glesjac(@fparab, u);
for j=1:M
  u=(eye(N-2)-k*A)\u; U=[U;[uL u' uR]];
end
surfc(X,T,U)
```

Resultatet med steglängderna `h=0.1; k=0.04;` blir enligt figuren nedan



Som synes har vi inga instabiliteter i lösningen, trots det stora tidssteget!

```
function J=glesjac(Fcn,z);
  NR=length(z); F=feval(Fcn,z); jac=[];jac=sparse(jac); stegtol=1.E-8;
  for i=1:NR,
   z0=z;
   st=z0(i)*stegtol; if st==0, st=1.E-10; end
   z0(i)=z0(i)+st;
   jac=[jac ( feval(Fcn,z0)-F )/st];
  end
  J=jac;
```

Det enda som skiljer från `minjac` är satsen `jac=sparse(jac);`