

Tentamen för Matematisk överbrygningskurs

Tid och plats: 2018-06-07, kl 14:00-18:00, L. **Ansvarig:** Jacques Huitfeldt, 031-7721093.

Betygsgränser: 20, 30 resp. 40 poäng. Tentan omfattar totalt 50 poäng.

Hjälpmedel: Inga hjälpmedel tillåtna, utom bifogat formelblad.

Uppgift 1. När en bilstrålkastare tänds värms glödtråden upp mycket fort och börjar lysa. När den släcks ser man att glödtråden svalnar och svartnar relativt långsamt. Med lämpligt val av skalor för tiden t och temperaturen u , gäller

$$\begin{cases} u' = q - (u^4 - 1), & 0 \leq t \leq T \\ u(0) = 1 \end{cases}$$

där $q = 1000$ då strålkastaren är tänd och $q = 0$ då den är släckt.

Strålkastaren tänds vid tiden $t = 0$ och släcks då $t = 0.003$.

(a). Beskriv högerledet i differentialekvationen som en funktionsfil i MATLAB.

(b). Skriv den kod i MATLAB som behövs för att lösa problemet med `ode45` för $T = 0.01$ och rita upp temperaturen som funktion av tiden.

Hela uppgiften ger maximalt **(8p)**

Uppgift 2. Betrakta egenvärdesproblemet

$$\begin{cases} -u'' + 5u = \lambda u, & 0 < x < L \\ u'(0) + u(0) = 0, & u(L) = 0 \end{cases}$$

(a). Gör en indelning av intervallet så att vi får n inre punkter. Skriv ned det egenvärdesproblem för matris vi får då vi i problemet ersätter derivator med differensapproximationer.

(b). Härled de differensapproximationer du använde i (a). Det skall framgå vilken noggrannhetsordning approximationerna har.

(c). Skriv ned den kod i MATLAB som behövs för att bygga upp matrisen samt lösa egenvärdesproblemet, komplettera lösningen med randvärden och rita upp. Matrisen skall lagras som en gles matris med `spdiags`. Vi skall beräkna de tre lösningarna med egenvärden närmast noll. Tag $L = 1$ och $n = 50$. Använd `eigs` för lösning av egenvärdesproblemet och `subplot` vid uppritning.

Hela uppgiften ger maximalt **(8p)**

Uppgift 3. Antag att vi har en $n \times n$ -matris \mathbf{A} och att det för matrisens egenvärden λ_i gäller att $|\lambda_1| \leq \dots \leq |\lambda_{n-1}| < |\lambda_n|$, samt att motsvarande egenvektorer \mathbf{v}_i , $i = 1, 2, \dots, n$, är linjärt oberoende.

Givet en startvektor \mathbf{u}_0 visa att potensmetoden $\mathbf{u}_{k+1} = \mathbf{A}\mathbf{u}_k$, $k = 0, 1, \dots$, konvergerar mot egenvektorn \mathbf{v}_n sammanhängande med det till belopp största egenvärdet λ_n .

Uppgiften ger maximalt **(8p)**

Uppgift 4. Betrakta det icke-linjära ekvationssystemet

$$\begin{cases} x_1^3 + x_2 - 1 = 0 \\ x_1 + x_2 \exp(x_1 x_2) + 1 = 0 \end{cases}$$

(a). Skriv systemet på kompakt form $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. Vad blir $\mathbf{f}(\mathbf{x})$ och vad blir $D\mathbf{f}(\mathbf{x})$, dvs. beräkna derivatamatrixen.

(b). Härled Newtons metod för ekvationen $\mathbf{f}(\mathbf{x}) = \mathbf{0}$.

(c). Tag ett steg med Newtons metod utgående från startapproximationen $\mathbf{x}_0 = (0, 1)$ för hand, dvs. utför beräkningarna med papper och penna.

(d). Skriv ned den kod i MATLAB som utgående från startapproximationen $\mathbf{x}_0 = (0, 1)$ beräknar lösningen med Newtons metod.

Hela uppgiften ger maximalt (8p)

Uppgift 5. Vi skall använda Steepest descentmetoden för att minimera

$$f(\mathbf{x}) = x_1^2 - 3x_1 + x_1 x_2 + x_2^2$$

(a). Utgå från startapproximationen $\mathbf{x}_0 = (1, 1)$ och ta ett steg med Steepest descentmetoden, dvs. bilda nästa approximation \mathbf{x}_1 . Använd optimalt steglängdsval. Räkna för hand och redovisa alla beräkningar.

(b). Skriv ned den kod i MATLAB som, utgående från startapproximationen i (a), beräknar en lösning till minimeringsproblemet med Steepest descentmetoden. Använd `fminbnd` för att lösa delproblemen.

Hela uppgiften ger maximalt (8p)

Uppgift 6. Betrakta följande hyperboliska partiella differentialekvation

$$\begin{cases} u''_{tt} = u''_{xx}, & 0 < x < 1, 0 \leq t \leq 5 \\ u(0, t) = 0, u(1, t) = \sin(3t), & 0 \leq t \leq 5 \\ u(x, 0) = \sin(2\pi x), u'_t(x, 0) = 0, & 0 < x < 1 \end{cases}$$

(a). Vi skall lösa problemet med linjemetoden. Skriv ned det begynnelsevärdesproblem vi får då vi approximerar x -derivatorna med differenskvoter. Vi vill ha svaret på vektorform.

(b). Skriv ned den kod som i MATLAB behövs för att med `ode45` beräkna och sedan rita upp lösningen $u(x, t)$.

Hela uppgiften ger maximalt (10p)

Differensapproximationer

$$D_+u(x) = \frac{u(x+h) - u(x)}{h} = u'(x) + \mathcal{O}(h), \quad D_-u(x) = \frac{u(x) - u(x-h)}{h} = u'(x) + \mathcal{O}(h)$$

$$D_0u(x) = \frac{u(x+h) - u(x-h)}{2h} = u'(x) + \mathcal{O}(h^2)$$

$$D_+D_-u(x) = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = u''(x) + \mathcal{O}(h^2)$$

$$\frac{u(x+2h) - 2u(x+h) + 2u(x-h) - u(x-2h)}{2h^3} = u'''(x) + \mathcal{O}(h^2)$$

$$\frac{u(x+2h) - 4u(x+h) + 6u(x) - 4u(x-h) + u(x-2h)}{h^4} = u^{(4)}(x) + \mathcal{O}(h^2)$$

Taylorutveckling

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots$$

$$+ \frac{f^{(n)}(a)}{n!}(x-a)^n + R_{n+1}(x)$$

$$R_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1} = \mathcal{O}((x-a)^{n+1})$$

Taylorutveckling i två variabler

$$f(x, y) = f(a, b) + f'_x(a, b)h + f'_y(a, b)k + \frac{1}{2}(f''_{xx}(a, b)h^2 + 2f''_{xy}(a, b)hk + f''_{yy}(a, b)k^2) + \dots$$

där $h = x - a$ och $k = y - b$.

Lite stolpar i MATLAB som stöd för minnet.

Elementära funktioner

`abs(x)`, `sqrt(x)`, `exp(x)`, `log(x)`, `cos(x)`, `sin(x)`, `tan(x)`

Matris- och vektorfunktioner

`size(A)`, `length(v)`

`sum(v)`, `prod(v)`, `max(v)`, `min(v)`, `sort(v)`, `norm(v)`, `dot(u,v)`

Matrisuppbyggande funktioner

`A=ones(m,n)`, `A=zeros(m,n)`, `A=eye(m,n)`, `A=diag(d,k)`, `A=spdiags(D,K,m,n)`

Villkorssatser

```
if uttryck
    satser
end
```

```
if uttryck
    satser
else
    satser
end
```

```
if uttryck
    satser
elseif uttryck
    satser
end
```

Repetitionssatser

```
while uttryck
    satser
end
```

```
for variabel=uttryck
    satser
end
```

```
for variabel=start:steg:slut
    satser
end
```

Egna funktioner

```
function ut=funktionsnamn(parametrar)
    satser
```

```
handtagsnamn=@(parametrar) sats
```

Beräkningsfunktioner

`x=fzero(f,x0)`, `x=fminbnd(f,x0,x1)`, `q=integral(f,a,b)`, `[t,U]=ode45(f,tspan,u0)`

`x=A\b`, `[V,D]=eig(A)`, `[V,D]=eigs(A,K,'SM')`

`x=fsolve(f,x0)`, `x=fminunc(f,x0)`, `q=integral2(f,a,b,c,d)`

Grafik

`x=linspace(a,b,n)`

`plot(x,y)`, `plot3(x,y,z)`, `quiver(u,v,du,dv,sc)`, `quiver3(...)`

`[X,Y]=meshgrid(x,y)`

`surf(X,Y,Z)`, `contour(X,Y,Z,lv)`