

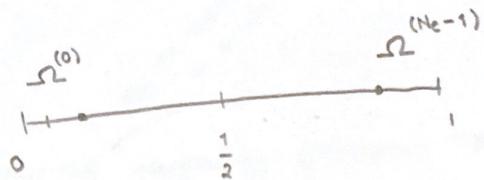
# HJEMMEKSAKEN INS270

Kandidatnummer 15616

## Problem 1

$$-\frac{du}{dx^2} = f(x), \quad \frac{du(0)}{dx} = C, \quad u(1) = D$$

Vi har  $N_e$  elementer, det viser  
sig at vi har  $N_e + 1$  globale  
noder.



a) Generelt så er prosedyren

1. Finn the weak variational form til problemet.

$$\sum_{i=0}^N \langle \psi'_i, \psi'_j \rangle c_i = \langle f(x), \psi_j \rangle - C \psi_j(0) - \langle D \psi'_N, \psi'_j \rangle \quad (1)$$

Der  $\psi_j$ -ene er Lagrange polynomer, som her er våre basis funksjoner.  
 $N$  er antall koeffisienter vi må bestemme, altså antall weiente.

Avhengig om vi er på  $x \in [0, 1/2]$  eller  $x \in [1/2, 1]$  avgjør om  
 $\psi_j$  er et flersteds- eller andre grads polynom.  $c_i$ -ene er koeffisientene  
vi ønsker å bestemme

2. likning (1) kan skrives som en matrise likning der A er  
elementmatrisen og b er elementvektoren

$$\sum_{j=0}^N A_{ij} c_j = b_i$$

der  $A_{ij} = \langle \psi'_i, \psi'_j \rangle$  og  $b_i = \langle f(x), \psi_i \rangle - C \psi_i(0) - \langle D \psi'_N, \psi'_i \rangle$

Siden vi skal se på det neste P1 elementet vil leddet  $C\psi_i(0) = 0$

pga.  $\psi_i(0) = 0$  for alle  $i \neq 0$  og vil bare gi et bidrag til  $b_0$ .

Ettersom vi har en grensebetingelse i  $x=1$  er det ikke én basisfunksjon som vil bidra til elementmatrisen.

Generelt er Lagrange polynomene gitt av

$$\psi_i = \prod_{\substack{j=0 \\ j \neq i}}^d \frac{x - x_j}{x_i - x_j}$$

For P1 elementer  
er  $d=2$ .

der  $d$  er antall noder i elementet. lagrange polynomet for den

siste noden er

$$\psi_{N-1} = \frac{x - (1-h)}{1 - (1-h)} = -\frac{x-1+h}{h}$$

For den nest siste noden:

$$\psi_{N-2} = \frac{x - 1}{(1-h) - 1} = -\frac{x-1}{h}$$

Bidragene til elementmatrisen for det neste P1 elementet vil da bare være diagonalelementet  $\langle \psi_{N-2}, \psi_{N-2} \rangle$

$$\frac{d\psi_{N-2}}{dx} = -\frac{1}{h}$$

$$\langle \psi_{N-2}', \psi_{N-2}' \rangle = \int_{1-h}^1 \frac{1}{h^2} dx = \frac{1}{h^2} [1 - (1-h)] = \frac{1}{h}$$

For vektoren for vi

$$b_{N-2} = \langle 2x-1, \psi_{N-2} \rangle - \langle D\psi_{N-2}', \psi_{N-2}' \rangle$$

b) For de  $\frac{N_e}{2}$  første elementene vil bidraget til elementmatrisen være  $P_2$   
 3x3 matriser, mens for  $P_1$  elementene vil det være  $2 \times 2$  matriser  
 (bortsett fra det siste elementet).

Når man setter sammen matrisen så kobles elementene sammen ved å plusse på det siste diagonalelementet i element  $j$  med det første diagonalelementet i element  $j+1$ .

Bidraget til den totale elementmatrisen  $A$  fra alle  $P_2$  elementene vil derfor være en matrise  $A_{P_2} \in \mathbb{R}^{(2 \cdot \frac{N_e}{2} + 1) \times (2 \cdot \frac{N_e}{2} + 1)} = \mathbb{R}^{(N_e+1) \times (N_e+1)}$ . Bidraget fra alle  $P_1$  elementene vil være  $A_{P_1} \in \mathbb{R}^{\left(\frac{N_e}{2}\right) \times \left(\frac{N_e}{2}\right)}$

Disse to matrisene,  $A_{P_1}$  og  $A_{P_2}$ , skal så settes sammen ved å plusse det siste diagonalelementet i  $A_{P_2}$  med det første diagonalelementet i  $A_{P_1}$ . Den resulterende matrisen blir da

$$A \in \mathbb{R}^{\left[\left(N_e+1\right)+\frac{N_e}{2}-1\right] \times \left[\left(N_e+1\right)+\frac{N_e}{2}-1\right]} = \mathbb{R}^{\frac{3N_e}{2} \times \frac{3N_e}{2}}$$

Når vi ser på bidraget fra  $P_2$  elementene vil vi få 3 ikke-null elementer på hver rad, bortsett fra i de radene der vi kombinerer elementene, der vil det være 5 ikke-null elementer. Altå vi det i elementene, der vil det være 5 ikke-null elementer.

$$\text{Antall ikke null elementer } \begin{cases} 5 & \text{i rad } 3+2j \text{ for } j=0,1,\dots \frac{N_e}{2} \\ 3 & \text{i resten.} \end{cases}$$

Der vi kobler sammen matrisene  $A_{P_2}$  og  $A_{P_1}$  vil det være 4 ikke-null elementer. I de resterende radene vil vi ha 3 ikke-null elementer, bortsett fra i siste rad, der vil det være 2 ikke-null elementer.

Skal nå finne de faktiske verdiene til de ulike matrise elementene:

Før P2 elementene:

Siden  $h = \frac{1}{N_e}$ , er avstanden mellom alle node (både lokale og globale),

node like  $h' = \frac{h}{2}$ . For elementet til venstre,  $\Omega^{(0)}$ , får vi:

$$\Psi_0 = \frac{(x-h')(x-2h')}{2h'^2}, \quad \Psi_1 = -\frac{x(x-2h')}{h'^2}, \quad \Psi_2 = \frac{x(x-h')}{2h'^2}$$

Da blir matrise elementene (dette er basert på utregningene i prosjekt 2)

$$A_{P2}^{(0)} = \begin{pmatrix} 7/6h & -4/3h & 1/6h \\ -4/3h & 8/3h & -4/3h \\ 1/6h & -4/3h & 7/6h \end{pmatrix}$$

For et indre P2 element med index e,  $\Omega^{(e)}$ , der den andre noden

er  $jh'$  og de delte node er  $(j \pm 1)h'$ , der  $j = 3, 5, 7, \dots, \frac{N_e}{2} - 1$ . Vi får

$$\Psi_{j-1} = \frac{(x-jh')(x-jh'-h')}{2h'^2}, \quad \Psi_j = -\frac{(x-jh'+h')(x-jh'-h')}{h'^2},$$

$$\Psi_{j+1} = \frac{(x-jh')(x-jh'+h')}{2h'^2}$$

Bidragene til  $A_{P2}^{(e)} = A_{P2}^{(0)}$ . Når man da setter de sammen skal

det fjerde elementet  $A_{00}^{(e)}$  plusses sammen med  $A_{22}^{(e-1)}$ .

For P1 elementene:

Her vil  $jn$  være den venstre noden i alle elementene og  $(j+1)n$  den høyre.

For alle P1 elementene får vi følgende basisfunksjoner:  $j = \frac{N_e}{2}, \frac{N_e}{2} + 1, \dots N_e - 2$

$$\psi_j = \frac{x - (j+1)n}{jn - (j+1)n} = -\frac{x - jn - n}{n}$$

$$\psi_{j+1} = \frac{x - jn}{jn+n - jn} = \frac{x - jn}{n}$$

Må nå finne matriseelementene ved å regne ut  $\langle \psi'_m, \psi'_n \rangle$

$$A_{P1}^{(e)} = \begin{pmatrix} \langle \psi'_j, \psi'_j \rangle & \langle \psi'_j, \psi'_{j+1} \rangle \\ \langle \psi'_{j+1}, \psi'_j \rangle & \langle \psi'_{j+1}, \psi'_{j+1} \rangle \end{pmatrix}$$

$$\psi'_j = -\frac{1}{n}, \quad \psi'_{j+1} = \frac{1}{n}$$

$$\langle \psi'_j, \psi'_j \rangle = \int_{jn}^{jn+n} \frac{1}{n^2} dx = \frac{1}{n^2} [jn+n - jn] = \frac{1}{n}$$

$$\langle \psi'_j, \psi'_{j+1} \rangle = \langle \psi'_{j+1}, \psi'_j \rangle = \int_{jn}^{jn+n} -\frac{1}{n^2} dx = -\frac{1}{n}$$

$$\langle \psi'_{j+1}, \psi'_{j+1} \rangle = \int_{jn}^{jn+n} \frac{1}{n^2} dx = \frac{1}{n}$$

$$A_{P1}^{(e)} = \begin{pmatrix} 1/n & -1/n \\ -1/n & 1/n \end{pmatrix}$$

Vi har nå alle elementene som inngår i A.

# 1c)

```
1 import numpy as np
2
3
4 def piecewise_func(c, Ne, psi, D, N):
5     """
6         Input parametere:
7             c      : Koeffisient vektoren, oppnnd ved lse matriselikningen Ac = b
8             Ne    : Antall elementer
9             psi   : En liste med basisfunksjonene som funksjoner, alts at de kan ta et argument x
10            D     : Grensebetingelsen for u(1,t) = D
11            N     : Antall punkter jeg vil evaluere u i hvert element
12
13
14     c = np.append(c, D)      # Legger til grensebetingelsen slik at vi unngr en if-test nr vi finner lsningen
15
16     N_p1 = int(Ne/2)        # Antall P1 elementer
17     N_p2 = int(Ne/2)        # Antall P2 elementer
18
19
20     x = np.linspace(0, 1, Ne+1)
21
22     # Degrees of freedom map som holder de lokale nodene for P2 elementene
23     dof_map_P2 = []
24     node = 0
25     for e in range(N_p2):
26         dof_map_P2.append([])
27         for i in range(node, 3+node):
28             dof_map_P2[e].append(i)
29         node += 2
30
31     # Degrees of freedom map som holder de lokale nodene for P1 elementene
32     dof_map_P1 = []
33     for e in range(N_p1):
34         dof_map_P1.append([])
35         for i in range(node, 2+node):
36             dof_map_P1[e].append(i)
37         node += 1
38
39
40     x_total = []      # Holder den totale x-arrayen
41     u = []           # Holder den totale lsningen
42
43     counter = 0
44     for index, e in enumerate(dof_map_P2):
45         start = x[index]
46         slutt = x[index+1]
47
48         x_lokal = np.linspace(start, slutt, N)
49
50         u.append(c[dof_map_P2[e][0]]*psi[dof_map_P2[e][0]](x_lokal) + c[dof_map_P2[e][1]]*psi[dof_map_P2[e][1]](x_lokal) + c[dof_map_P2[e][2]]*psi[dof_map_P2[e][2]](x_lokal))
51
52         x_total = np.concatenate((x_total, x_lokal))
53         counter = index
54
55     for index, e in enumerate(dof_map_P1):
56         start = x[index+counter]
57         slutt = x[index+1+counter]
58
59         x_lokal = np.linspace(start, slutt, 10)
60
61         u.append(c[dof_map_P1[e][0]]*psi[dof_map_P1[e][0]](x_lokal) + c[dof_map_P1[e][1]]*psi[dof_map_P1[e][1]](x_lokal) + c[dof_map_P1[e][2]]*psi[dof_map_P1[e][2]](x_lokal))
62
63         x_total = np.concatenate((x_total, x_lokal))
64
65
66     u = np.concatenate(u, axis=0)      # Siden u er en liste med arrayer, gjjr dette u til n array
67
68     # For unng at samme x-koordinat blir brukt to ganger, finner indeksen til de elementene i x_total som er like
69     idx = np.unique(x_total, return_index=True)[0]
70
71     X = x_total[idx]
```

```

72     U = u[idx]
73
74     return x, U

```

For å unngå if-tester inne i for-løkkene er selve utregningen delt opp i to løkker, avhengig av hvor vi er på domenet. Ettersom vi allerede vet hvor på x-aksen de ulike elementene er definert, trenger vi ikke if-tester siden vi kan finne de med indeksering, gitt at elementene er sortert i logisk og stigende rekkefølge. Det ville vært verre om vi hadde hatt en tilfeldig sortering av elementene, da det måtte ha gått med ekstra regneoperasjoner til å sortere disse.

Ettersom vi har et x-array kan vi for hvert element regne ut følgende vektorisert:

$$u(\mathbf{x}) = c_n \psi_n(\mathbf{x}) + \dots + c_m \psi_m(\mathbf{x}), \quad (1)$$

for  $\mathbf{x} \in [nh, mh]$ , og det er dette som skjer i linje 49 og 60.

## 1d)

Trapesmetoden for numerisk integrasjon er gitt av

$$\int_a^b f(x)dx \approx h \left( \frac{f(a) + f(b)}{2} + \sum_{i=1}^{N-1} f(x_i) \right), \quad (2)$$

der  $h$  er steglengden og  $N$  er antall integrasjonspunkter.  $L_2$  normen, som er definert som

$$L_2 = \sqrt{\int_{\Omega} (u_e(x) - u(x))^2 dx}, \quad (3)$$

blir da

$$L_2 \approx \sqrt{h \left( \frac{(u_e(a) - u(a))^2 + (u_e(b) - u(b))^2}{2} + \sum_{i=1}^{N-1} (u_e(x_i) - u(x_i))^2 \right)}, \quad (4)$$

siden  $f(x) = u_e(x) - u(x)$ .

```

1 import numpy as np
2
3
4 def L2_norm(u_e, u, Ns):
5     """
6         Input parametere:
7             u_e : den eksakte løsningen som er en funksjon av x
8             u   : den tilnærmede løsningen i diskrete punkter
9             Ns  : antall integrasjonspunkter
10    """
11
12    h = 1/(Ns-1)
13    x = np.linspace(0, 1, Ns)
14
15
16    # Regner ut arealet av trapesene som tilhører de indre punktene
17    indre_punkter = 0
18    for i in range(1,Ns):
19        indre_punkter += (u_e(x[i]) - u[i])**2
20
21    endepunkter = ((u_e(x[0]) - u[0])**2 + (u_e(x[-1]) - u[-1])**2)/2
22
23    return np.sqrt(h*(endepunkter + indre_punkter))

```

Antar at  $N_s$  samsvarer med antall punkter vi har brukt for å finne den numeriske løsningen i forrige oppgave.

Når det gjelder konvergensrate vet vi at siden basisfunksjonene til P2 elementene er kvadratiske polynomer, har de en konvergensrate på 3, mens basisfunksjonene for P1 elementene har en rate på 2. Dette er basert på et a priori estimat av løsningen, som sier at feilen konvergerer som  $h^{s+1}$  der  $s$  er polynomgraden til basisfunksjonene. Derfor er det naturlig å anta at konvergensraten til  $L_2$  normen vil ligge et sted mellom 2 og 3.

For å finne koeffisientene i likningssystemet  $Ac = b$ , utfører man en matrise invertering av  $A$ . Dette betyr at  $A$  må være ikke-singulær. Et problem som kan oppstå mtp. denne betingelsen er hvis noen av elementene i  $A$  er veldig små. Dette kan gi numeriske feil, som kan føre til at koeffisientene ikke blir helt korrekte.

$$2) \quad \frac{du(1)}{dx} = -a(u(1) - D) \quad , \quad \frac{du(0)}{dx} = C$$

Vi vet at den 1D Poisson likningen kan uttrykkes med følgene weak variasjonal form:

$$\langle u''(x), \psi \rangle = -\langle f(x), \psi \rangle \quad \forall \psi \in V$$

der  $V$  er et vektorrom  $V = \text{Span}\{\psi_0, \dots, \psi_N\}$ .

Vi deler integrer før å redusere til et 1. ordens problem. Vi får da

$$\langle u', \psi' \rangle = \langle f, \psi \rangle + u'(1)\psi(1) + u'(0)\psi(0)$$

Setter inn grensbehandlingene ser vi at vi får

$$\begin{aligned} \langle u', \psi' \rangle &= \langle f, \psi \rangle - a(u(1) - D)\psi(1) + C\psi(0) \\ &= \langle 2x-1, \psi \rangle - au(1)\psi(1) + aD\psi(1) + C\psi(0) \end{aligned}$$

I det forrige problemet måtte man kreve at  $\psi(1) = 0$ , som gjorde at vi mistet det leddele. Vi vet at løsningen fortsatt kan uttrykkes som en lineær kombinasjon av vektorene i  $V$ .

Vi ser at vi trenger en verdi for  $u(1)$ . Elementmatrisen vil endres i bidraget fra det siste P1 elementet. Der vi i sted kun filte et diagonalelement for vi nå en fullstendig  $2 \times 2$  matrise som må assemblees med resten.

$$A_{P1}^{(N-1)} = \begin{pmatrix} \langle \psi'_{N-2}, \psi'_{N-1} \rangle & \langle \psi'_{N-2}, \psi'_{N-1} \rangle \\ \langle \psi'_{N-2}, \psi'_{N-1} \rangle & \langle \psi'_{N-1}, \psi'_{N-1} \rangle \end{pmatrix}$$

Vektoren  $\alpha$  vil også utvides med ett element, som da vil være

$$b_{N+1} = \langle 2x-1, \psi_{N+1} \rangle - a u(1) \psi(1) - a D \psi(1).$$

Vi vet at  $\psi_i(1) = \delta_{i,N+1}$ , sett at de nye bidragene fra Robin betingelsen ikke endrer på vektor-elementet som korresponderer med den siste basisfunksjonen.

## Problem 2

a) Backward Euler i tid finner vi fra å Taylor utvikle bakover i tid. Ergo har vi

$$u(x, t - \Delta t) = u(x, t) - \Delta t \frac{\partial u(x, t)}{\partial t} + O(\Delta t^2)$$

$$\frac{\partial u(x, t)}{\partial t} = \frac{u(x, t) - u(x, t - \Delta t)}{\Delta t} + O(\Delta t)$$

Innferer notasjonen  $u(x_j, t_m) \equiv u_j^m$  der  $x_j = j\Delta x$  og  $t_m = m\Delta t$

der  $j = 0, \dots, N_x$  slik at  $\Delta x = \frac{1}{N_x}$  og  $m = 1, \dots, N_t$  slik at  $\Delta t = \frac{T}{N_t}$ .

1D diffusjonsløsningen diskretisert i tid blir da

$$\frac{u_j^m - u_j^{m-1}}{\Delta t} = \frac{\partial^2 u_j^m}{\partial x^2} + \sin(u_j^m) + 2$$

difference

b) For å finne finite diskretisering i rom Taylor utvikler vi både fremover og tilbake i tid.

$$u_j^{m+1} = u_j^m + \Delta x \frac{\partial u_j^m}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 u_j^m}{\partial x^2} + \frac{\Delta x^3}{6} \frac{\partial^3 u_j^m}{\partial x^3} + O(\Delta x^4)$$

$$u_j^{m-1} = u_j^m - \Delta x \frac{\partial u_j^m}{\partial x} + \frac{\Delta x^2}{2} \frac{\partial^2 u_j^m}{\partial x^2} - \frac{\Delta x^3}{6} \frac{\partial^3 u_j^m}{\partial x^3} + O(\Delta x^4)$$

Vi plisser de to likningene sammen slik at vi får

$$u_j^{m+1} + u_j^{m-1} = 2u_j^m + \Delta x^2 \frac{\partial^2 u_j^m}{\partial x^2} + O(\Delta x^4)$$

$$\Rightarrow \frac{\partial^2 u_j^m}{\partial x^2} = \frac{u_j^{m+1} - 2u_j^m + u_j^{m-1}}{\Delta x^2} + O(\Delta x^2)$$

Vi får da følgende approksimasjon for diffusjonslikningen

$$\frac{u_j^m - u_j^{m-1}}{\Delta t} = \frac{u_{j+1}^m - 2u_j^m + u_{j-1}^m}{\Delta x^2} + \sin(u_j^m) + 2 \quad (1)$$

Denne approksimasjonen har en trunceringsfeil av størrelsesorden  $O(\Delta x^2, \Delta t)$ .  
 Vi løser opp likning (1) slik at vi får alle leddene som er avhengig av samme tidssteg på samme side.

$$u_j^m - \frac{\Delta t}{\Delta x^2} (u_{j+1}^m - 2u_j^m + u_{j-1}^m) - \Delta t \sin(u_j^m) = u_j^{m-1} + 2$$

Leddene  $u_{j+1}^m$  og  $u_{j-1}^m$  vil gi problemer i sine respektive endepunkter.

Når  $j=N_x$  vil  $u_{j+1}^m$  gå utenfor domenet, og det samme når  $j=0$ .

Må da bruke Neumann grensebetingelsen

$$\frac{\partial u^m}{\partial x} \underset{\text{centered difference}}{\approx} \frac{u_{j+1}^m - u_{j-1}^m}{2\Delta x} = 0 \Rightarrow u_{j+1}^m = u_{j-1}^m$$

Vi får da følgende system av ikke-lineære likninger

$$j=0 : u_0^m - \frac{\Delta t}{\Delta x^2} (2u_1^m - 2u_0^m) - \Delta t \sin(u_0^m) = u_0^{m-1} + 2$$

$$j \in [1, N_x-1] : u_j^m - \frac{\Delta t}{\Delta x^2} (u_{j+1}^m - 2u_j^m + u_{j-1}^m) - \Delta t \sin(u_j^m) = u_j^{m-1} + 2$$

$$j=N_x : u_{N_x}^m - \frac{\Delta t}{\Delta x^2} (2u_{N_x-1}^m - 2u_{N_x}^m) - \Delta t \sin(u_{N_x}^m) = u_{N_x}^{m-1} + 2$$

Ergo vi har  $N_x+1$  likninger, som alle er på formen over.

c) Siden Newton's metode er en iterativ metode definerer vi iterasjonstallet  $K$ , som sier hvilken iterasjon vi er i.

Vi definerer den generelle likningen

$$F(u_j^{m,k}) = u_j^{m,k} - \frac{\Delta t}{\Delta x^2} (u_{j+1}^{m,k} - 2u_j^{m,k} + u_{j-1}^{m,k}) - \Delta t \sin(u_j^{m,k}) - u_j^{m-1,k} - 2 = 0$$

Newton's metode lineariserer likningen over ved at den approksimerer

$F(u_j^{m,k+1})$  rundt  $u_j^{m,k-1} \equiv \bar{u}$ , som vil være en vektor man sjettet på i første iterasjonen. Definerer vi  $u_j^{m,k+1} \equiv u$  får vi

$$F(u) = F(\bar{u}) + (u - \bar{u}) F'(\bar{u}) + \frac{1}{2} F''(\xi)(u - \bar{u})^2$$

der  $\xi = \max_{u \in [a, b]} F''(u)$

Vi før da likningen

$$u = \bar{u} - \frac{F(\bar{u})}{F'(\bar{u})}$$

der  $F'(\bar{u})$  er derivat mhp.  $u$ .

I næste steg må man da sette  $\bar{u} = u^{(1)}$  der  $u^{(1)}$  er løsningen fra forrige steg. Vi kan definere følgende

algoritme

1. Ta et gjett på  $\bar{u}$  i første iterasjon, og itererer til man har konvergens

2. Hvis  $F'(\bar{u})(u - \bar{u}) = -F(\bar{u})$  mhp. vektoren  $(u - \bar{u}) \equiv \delta u$

3.  $u = \bar{u} + \delta u$

4.  $\bar{u} \leftarrow u$

d) Fra 2a har vi

$$\frac{u^m - u^{m-1}}{\Delta t} = \frac{\partial^2 u^m}{\partial x^2} + \sin u^m + 2 \quad (4)$$

Når vi skal linearisere via Taylor rekken er ideen at vi approksimerer  $u^m$  med  $u^{m,k}$  slik at

$$u^m = u^{m,k} + \delta u.$$

Setter så dette uttrykket inn i likning (4), Taylor utvikler de ikke-lineære leddene og beholder de leddene som er lineare mph  $\delta u$ .

Når vi setter inn får vi

$$\frac{u^{m,k} + \delta u - u^{m-1}}{\Delta t} = \frac{\partial^2}{\partial x^2} (u^{m,k} + \delta u) + \sin(u^{m,k} + \delta u) + 2 \quad (5)$$

Det eneste ikke-lineære leddet vi har er  $\sin(u^{m,k} + \delta u)$

Taylorutvikler det får vi

$$\sin(u^{m,k} + \delta u) = \sin(u^{m,k}) + \delta u \frac{d\sin(u^{m,k})}{du} + O(\delta u^2) \approx \sin(u^{m,k}) + \delta u \cos(u^{m,k})$$

Setter dette inn i likning (5)

$$\frac{u^{m,k} + \delta u - u^{m-1}}{\Delta t} = \frac{\partial^2 u^{m,k}}{\partial x^2} + \frac{\partial^2 \delta u}{\partial x^2} + \sin(u^{m,k}) + \delta u \cos(u^{m,k}) + 2$$

Høyer slik at vi får alle leddene med  $\delta u$  på samme side.

$$\frac{u^{m,k} - u^{m-1}}{\Delta t} - \frac{\partial^2 u^{m,k}}{\partial x^2} - \sin(u^{m,k}) - 2 = \frac{\partial^2 \delta u}{\partial x^2} - \frac{\delta u}{\Delta t} + \delta u \cos(u^{m,k})$$

Vi ser at venstre siden kun inneholder lempete ledd, og høyre siden er en lineær PDE for  $\delta u$ , som er stasjonær.

2) Vi har følgende PDE som må diskretiseres

$$\frac{\partial^2 \delta u}{\partial x^2} - \frac{\delta u}{\Delta t} + \delta u \cos(u^{m,k}) = g(u^{m,k})$$

der  $g(u^{m,k})$  tilsvarer venstre siden i følgende likning.

Vi diskretiserer leddeleddet som er dobbeldeleddert

$$\frac{\delta u_{j+1} - 2\delta u_j + \delta u_{j-1}}{\Delta x^2} - \frac{\delta u_j}{\Delta t} + \delta u_j \cos(u^{m,k}) = g(u^{m,k})$$

$$\delta u_{j+1} - 2\delta u_j + \delta u_{j-1} - \frac{\Delta x^2}{\Delta t} \delta u_j \Delta x^2 \cos(u^{m,k}) \delta u_j = \Delta x^2 g(u^{m,k})$$

Dette kan settes opp som et likningssystem der man løser for  $\delta u$ .

På samme måte som tidligere må vi ta hensyn til grensebetingelsene.

Vi har fortsatt Neumann betingelsen

$$\frac{\delta u}{\Delta x} = 0$$

$\delta u = (u - \bar{u})$  der  $\bar{u}$  er en lesning fra tidligere iterasjon.

Antar derfor at de samme grensesettingelsene gjelder for vår aktuelle PDE. Vi bruker disse til å uttrykke

$$\delta u_1 = \delta u_n \text{ og } \delta u_{N+1} = \delta u_{N-1}.$$

## 2f)

I den første metoden så diskretiserer vi før vi lineariserer. Denne metoden, så vidt jeg kan finne, kalles på norsk for ettersleps metoden [1]. Siden vi Taylor utvikler om  $u^-$ , og kun beholder de lineære leddene vil vi i teorien ha tregere konvergens enn når vi gjennomfører såkalt kvasi-linearisering [1]. Dette er den andre metoden vi brukte, der vi først lineariserer og så diskretiserer. Siden de begge er basert på Newtons metode, som kan vises at har kvadratisk konvergens. Ettersom ettersleps metoden har tregere konvergens enn kvasi-lineariseringen, antar jeg derfor at denne har nærmere kvadratisk konvergens.

En flervariabel versjon av ettersleps metoden medfører at man må regne ut Jacobi determinanten. Dette kan være en kostbar operasjon, som kan bidra til tregere konvergens.

## 2g)

Siden vi har diskrete punkter, så kan vi tilnærme dobbelt integralet med to uavhengige summer, som løper over de ulike  $x$  og  $t$  verdiene. Ettersom den eksakte løsning er definert overalt, mens den approksimerte er begrenset til diskrete punkter tilnærmer vi  $L_2$  normen med følgende dobbel sum

$$\int_0^T \int_0^1 (u_e(x, t) - u(x, t))^2 dx dt \approx \Delta x \Delta t \sum_{i=0}^{N_x} \sum_{j=0}^{N_t} f(x_i, t_j), \quad (5)$$

der  $f(x, t) = (u_e(x, t) - u(x, t))^2$ ,  $\Delta x = 1/N_x$  og  $\Delta t = T/N_t$ . Der vi vet at for  $j = 0$  så har vi intital betingelsen. Fra oppgave 2b) vet vi at den diskrete 1D diffusjonslikningen har en trunkeringsfeil som går som  $\mathcal{O}(\Delta x^2, \Delta t)$ , ergo antar vi at konvergensraten er lineær, ettersom ledet  $\Delta t$  dominerer  $\Delta x^2$ .

## Referanser

- [1] Jan Aarseth. Numeriske beregninger. <http://folk.ntnu.no/leifh/teaching/tkt4140/NumeriskeBeregninger.pdf>, 2014.