Eksamensoppgave 5 - INF5620

Henrik Andersen Sveinsson

December 16, 2013

1 Oppgavetekst

We look at the PDE problem

$$
\begin{aligned}
u_t &= \nabla \cdot ((1+\alpha_0 u^4)\nabla u), && \mathbf{x} \in \Omega,\ t > 0 \\
u(\mathbf{x}, 0) &= I(\mathbf{x}), && \mathbf{x} \in \Omega \\
u(\mathbf{x}, t) &= g(\mathbf{x}), && \mathbf{x} \in \partial\Omega_D,\ t > 0 \\
-(1+\alpha_0 u^4)\frac{\partial}{\partial n} u(\mathbf{x}, t) &= h(u - T_s), && \mathbf{x} \in \partial\Omega_R
\end{aligned}
$$

Here, u(x,t) is the temperature in some solid material, I is the initial temperature, g is the controlled temperature at a part $\partial\Omega_D$ of the boundary, while at the rest of the boundary, $\partial\Omega_D$, we apply Newton's cooling law with h as a heat transfer coefficient and T_s as the temperature in the surrounding air.

1.1 a

Perform a Crank-Nicolson time discretization and define a Picard iteration method for the resulting spatial problems. (Do not pay attention to the boundary conditions.)

1.2 b

Perform a Backward-Euler time discretization and derive the variational form for the resulting spatial problems. Use a Picard iteration method to linearize the variational form.

1.3 c

Apply Newton's method to the nonlinear variational form $F = 0$ in b). Set up expressions for the right-hand side ($-F$) and the coefficient matrix (Jacobian of F) in the linear system that must be solved in each Newton iteration.

2 Crank-Nicolson i tid og FEM med Picard i rom

Først skriver vi likningen på generell form:

$$u_t = \nabla \cdot (\alpha(u)\nabla u) \tag{1}$$

Vi begynner med å sette opp en Crank-Nicolson-diskretisering av tidsproblemet:

$$\frac{u^{n+1}-u^n}{\Delta t} = [\nabla \cdot (\alpha(u)\nabla u)]^{n+\frac{1}{2}} \tag{2}$$

For å evaluere i $n+\frac{1}{2}$ velger vi et aritmetisk snitt av hele høyresiden. Vi innfører samtidig notasjonen $u^{n+1} = u$, $u^n = u_1$.

$$\frac{u-u_1}{\Delta t} = \frac{1}{2}\left[\nabla \cdot (\alpha(u)\nabla u) + \nabla \cdot (\alpha(u_1)\nabla u_1)\right] \tag{3}$$

Dette leder til

$$u - \Delta t\frac{1}{2}\nabla \cdot (\alpha(u)\nabla u) = u_1 + \Delta t\frac{1}{2}\nabla \cdot (\alpha(u_1)\nabla u_1) \tag{4}$$

I denne linkningen er det kun u som er ukjent. Den inngår som et lineært ledd bak nabla-operatoren, men som et ikke-lineært ledd foran. Vi kan bruke picard-iterasjon for å løse dette problemet, og håpe på at løsningen nærmer seg den riktige.

Picard-iterasjon går ut på at den ukjente i det ikke-lineære leddet "gjettes", i dette tilfellet gjetter vi i utgangspunktet $u = u_1$. Vi løser det romlige problemet med dette satt inn. Vi kaller denne løsningen u_-. u_- går nå inn i de ikke-lineære leddene, som en litt bedre tilnærming til u enn den vi hadde. Dette kan gjøres mange ganger, for å komme nærmere den eksakte løsningen av det romlige problemet.

For ordens skyld skriver vi opp Picard-formuleringen av likningen:

$$u - \Delta t\frac{1}{2}\nabla \cdot (\alpha(u_-)\nabla u) = u_1 + \Delta t\frac{1}{2}\nabla \cdot (\alpha(u_1)\nabla u_1) \tag{5}$$

3 Backward Euler og variasjonell formulering

Vi setter nå opp problemet med Backward Euler. Det gir penere regning, siden vi ikke trenger å snitte i halvstegene. Vi går rett på notasjonen som ligger tettest opp mot programmeringen.

$$\frac{u-u_1}{\Delta t} = \nabla \cdot ((1+\alpha_0 u^4)\nabla u) \tag{6}$$

Vi setter så alle ledd med u på venstre side

$$u - \Delta t\nabla((1+\alpha_0 u^4)\nabla u) = u_1 \tag{7}$$

Så lager vi oss en variasjonell formulering:

$$(R, v) = 0 \tag{8}$$

Det gir:

$$(u,v) + \Delta t\left[((1+\alpha_0 u^4)\nabla u, \nabla v) - \left((1+\alpha_0 u^4)\frac{\partial u}{\partial n}, v\right)_{\partial\Omega}\right] = (u_1, v) \tag{9}$$

Her er det på tide å sette inn randbetingelsen:

$$-(1+\alpha_0 u^4)\frac{\partial}{\partial n}u(\mathbf{x},t) = h(u - T_s), \quad \mathbf{x} \in \partial\Omega_R \tag{10}$$

og

$$u(\mathbf{x},t) = g(x),\ x \in \Omega_D \tag{11}$$

Vi bruker en Boundary-function på den delen av randen som har Dirichlet-betingelse. Det vil si at randbetingelsen er $u = 0$ når vi setter opp variasjonsformuleringen.

Dette gir oss en endelig varisajonell formulering:

$$(u, v) + \Delta t \left[((1 + \alpha_0 u^4)\nabla u, \nabla v) + (h(u - T_s), v)_{\partial\Omega_R}\right] = (u_1, v) \tag{12}$$

Så innfører vi picard-iterasjon for å linearisere den variasjonelle formen. Det er i praksis bare å sette inn $u_{\text{-}}$ i alle de ikke-lineære leddene.

4 Newtons metode på den ikke-lineære variasjonelle formuleringen

Det skal finnes en u som oppfyller den variasjonelle formuleringen. Målet med Newtons metode er å finne denne ved å følge gradienten til den variasjonelle formuleringen med hensyn på ekspansjonskoeffisientene i den basisen vi velger å utspenne løsningen i.

Vi setter opp likningen vi vil løse for newtons metode. Vi setter $(1+\alpha_0 u^4) = \alpha(u)$ for penere notasjon. Vi ønsker egentlig å løse

$$F = 0 \tag{13}$$

men det klarer vi ikke pga det ikke-lineære leddet. Derfor går vi heller for å løse det for lineariseringen $\hat{F}$ av F. Når dette gjøres iterativt kalles det newtons metode. Vi ønsker å justere u_- slik at:

$$F(u) \approx \hat{F}(u) = F(u_-) + F'(u_-)(u - u_-) = 0 \tag{14}$$

Det vi effektivt kommer til å løse er

$$J_{i,j}\delta u = -F(u_-) \tag{15}$$

for δu, og så oppdatere u_-.

$$(u, v) + \Delta t \left[(\alpha(u)\nabla u, \nabla v) + (h(u - T_s), v)_{\partial\Omega_R}\right] - (u_1, v) = 0 \tag{16}$$

Nå antar vi at $u = \sum_{j\in\mathcal{I}_s} c_j\psi_j$. Da er $v = \psi_i \ \forall i \in \mathcal{I}_s$.

Vi må regne ut jacobimatrisen til F:

$$J_{i,j} = \frac{\partial F_i}{\partial c_j} \tag{17}$$

Vi gjør så derivasjonen med hensyn på c_i:

$$J_{i,j} = (\psi_j, \psi_i) + \Delta t\left[(\alpha'(u)\psi_j\nabla u, \nabla\psi_i) + (\alpha(u)\nabla\psi_j, \nabla\psi_i) + (h\psi_j, \psi_i)_{\partial\Omega_R}\right] \tag{18}$$

(u_1, v) forsvinner siden den bruker c-verdier fra et annet tidssteg.

Nå kjenner vi jacobimatrisen. Vi setter inn de kjente verdiene for u fra forrige tidssteg, og bruker dem som utgangspunkt for newtons metode. Vi kaller disse for $u_{\text{-}}$, og kjører metoden til den har konvergert tilstrekkelig bra. I det siste randleddet kjenner vi u, siden den er gitt ved $u = g$. Den endelige jacobimatrisen er dermed:

$$J_{i,j} = (\psi_j, \psi_i) + \Delta t\left[(\alpha'(u_{\text{-}})\psi_j\nabla u_{\text{-}}, \nabla\psi_i) + (\alpha(u_{\text{-}})\nabla\psi_j, \nabla\psi_i) + (h\psi_j, \psi_i)_{\partial\Omega_R}\right] \tag{19}$$