Prosjekt 1

Live Wang Jensen

September 4, 2016

Abstract

I dette prosjektet skal vi se nærmere på en numerisk løsning av den velkjente Poisson-ligningen, hvor Dirichlets grensebetingelser er brukt. Den andrederiverte er blitt tilnærmet med en tre-punkts-formel, og selve problemet løses ved hjelp av et lineært ligningssett. Vi skal løse disse ligningene på to ulike måter; ved Gauss-eliminasjon og ved LU-faktorisering. De to løsningsmetodene skal så sammenlignes når det kommer til FLOPS (floating point operations per second) og beregningstid.

1 Introduksjon

Vi starter med å se på den én-dimensjonale Poisson ligningen for en sfærisk kule funksjon (?). Denne ligningen løses numerisk ved å dele opp, eller disktretisere, intervallet $x \epsilon$ [0,1], og løsningen f på ligningen er gitt. Teoridelen viser at det å finne en diskret løsning vil være det samme som å løse et lineært ligningssett, $A\mathbf{u} = \tilde{\mathbf{b}}$. Det vil vise seg at matrisen A er en såkalt tridiagonal matrise, noe som forenkler Gauss-eliminasjonen betraktelig. Selve metoden er implementert i et Python-program, hvor vi har variert antall grid-points n. Den numeriske løsningen sammenlignes så med den analytiske løsningen. Deretter beregnes den maksimale relative feilen i den numeriske løsningen. Resultatet plottes som en funksjon av n. Helt til slutt skal vi bruke LU-faktorisering på matrisen A til å finne den numeriske løsningen på Poisson ligningen. Antall FLOPS og beregningstid sammenlignes så mellom de to løsningmetodene.

2 Teori

Mange differensialligninger innenfor fysikk kan skrives på formen

$$\frac{d^2y}{dx^2} + k^2(x)y = f(x) \tag{1}$$

hvor f er det uhomogene leddet i ligningen, og k^2 er en reell funksjon. Et typisk eksempel på en slik ligning finner vi i elektromagnetismen, nemlig Poisson-ligningen

$$\triangle^2\Phi = -4\pi\rho(\mathbf{r}) \tag{2}$$

Her er Φ det elektrostatiske potensialet, mens$\rho(\mathbf{r})$ er ladningsfordelingen. Dersom vi antar at Φ og $\rho(\mathbf{r})$ er sfærisk symmetriske, kan vi forenkle ligning (2) til en én-dimensjonal ligning,

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\Phi}{dr}\right) = -4\pi\rho(r) \tag{3}$$

Hvis vi bruker at $\Phi(r) = \phi(r)/r$, kan ligningen skrives som

$$\frac{d^2\phi}{dr^2} = -4\pi r\rho(r) \tag{4}$$

Det uhomogene leddet f er nå gitt ved produktet $-4\pi\rho$. Dersom vi lar $\phi \to u$ og $r \to x$, ender vi opp med en generell, én-dimensjonal Poisson ligning på formen

$$-u''(x) = f(x) \tag{5}$$

Det er denne ligningen vi skal se nærmere på i denne oppgaven. Nærmere bestemt skal vi løse ligningen

$$-u''(x) = f(x), \quad x \in (0,1), \quad u(0) = u(1) = 0 \tag{6}$$

Vi skal altså holde oss innenfor intervallet $x \in (0,1)$, med Dirichlet-grensebetingelsene gitt ved $u(0) = u(1) = 0$. Vi definerer den diskrete tilnærmingen til løsningen u med v_i, slik at $x_i = ih$, hvor $h = 1/(n+1)$ er steglengden. Vi får da at $x_0 = 0$ og $x_{n+1} = 1$. Den andrederiverte av u kan da tilnærmes med en tre-punkts formel

$$-\frac{v_{i+1} + v_{i-1} - 2v_i}{h^2} = f_i \quad \text{når} \quad i = 1, ..., n \tag{7}$$

hvor $f_i = f(x_i)$ og grensebetingelsene er gitt som $v_0 = v_{n+1} = 0$. Feilleddet her går som $O(h^2)$.

Dersom vi multipliserer leddet h^2 i ligning (7) på hver side, kan vi definere leddet på høyre side av ligningen som $\tilde{b_i} = h^2 f_i$. Vi ender altså opp med

$$-v_{i+1} - v_{i-1} + 2v_i = \tilde{b_i} \tag{8}$$

Dersom vi setter inn ulike verdier av i i ligningen ovenfor, ender vi opp med en tilhørende ligning på samme form:

$$i = 1: \quad v_2 + v_0 - 2v_1 = \tilde{b_1}$$

$$i = 2: \quad v_3 + v_1 - 2v_2 = \tilde{b_2}$$

osv. Vi ender altså opp med et lineært ligningssett som kan settes opp som en matriseligning:

$$
\mathbf{A} = \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \dots \\ 0 & -1 & 2 & -1 & 0 & \dots \\ & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & & -1 & 2 & -1 \\ 0 & \dots & & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \dots \\ \dots \\ \dots \\ v_n \end{pmatrix} = \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \dots \\ \dots \\ \dots \\ \tilde{b}_n \end{pmatrix} \tag{9}
$$

hvor A er en tridiagonal nxn-matrise. Hvis vi kaller vektorene

$$
\begin{pmatrix} v_1 \\ v_2 \\ \dots \\ \dots \\ \dots \\ v_n \end{pmatrix} = \mathbf{v} \quad \text{og} \quad \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \dots \\ \dots \\ \dots \\ \tilde{b}_n \end{pmatrix} = \tilde{\mathbf{b}} \tag{10}
$$

kan vi på forkortet form skrive ligningen som

$$
\mathbf{A}\mathbf{v} = \tilde{\mathbf{b}} \tag{11}
$$

Her er A og $\tilde{\mathbf{b}}$ kjent, mens vektoren $\mathbf{v}$ er den ukjente.

I vårt tilfelle er funksjonen f er gitt som $f(x) = 100e^{-10x}$. Dersom vi bruker dette i ligning (5) og integrerer denne ligningen analytisk, ender vi opp med en løsning på formen

$$
u(x) = 1 - (1 - e^{-10})x - e^{-10x} \tag{12}
$$

Det er denne analytiske løsningen vi skal sammenligne den numeriske løsningen med.

3 Løsningsmetoder

3.1 Gauss-eliminasjon

Generell tridiagonal matrise

Vi kan tenke oss at elementene langs diagonalen i matrisen vår, A, utgjør en vektor b, samtidig som elementene på hver side av diagonalen utgjør vektorene a og c. Matriseligningen kan da skrives

$$
\mathbf{A} = \begin{pmatrix} b_1 & c_1 & 0 & \dots & \dots & \dots \\ a_2 & b_2 & c_2 & \dots & \dots & \dots \\ & a_3 & b_3 & c_3 & \dots & \dots \\ & \dots & \dots & \dots & \dots & \dots \\ & & & a_{n-2} & b_{n-1} & c_{n-1} \\ & & & & a_n & b_n \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \dots \\ \dots \\ \dots \\ v_n \end{pmatrix} = \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \dots \\ \dots \\ \dots \\ \tilde{b}_n \end{pmatrix}. \tag{13}
$$

En tilfeldig ligning i dette ligningssettet vil kunne skrives som

$$a_i v_{i-1} + b_i v_i + c_i v_{i+1} = \tilde{b}_i \quad \text{når} i = 1, 2, ..., n \tag{14}$$

Spesialtilfelle
I vårt tilfelle har elementene langs diagonalen identisk verdi.

4 Vedlegg

Github-adresse: LDKFJLDKSF

References

[1] *project*1_2016.*pdf* found at the official Github-page of the course *FYS3150 - Computational Physics* https://github.com/CompPhysics/ComputationalPhysics, 03.09.2016