FYS3150 Computational Physics 2014

Oblig 1

Løysning av lineære likningar på matriseform.

Øyvind Sigmundson Schøyen

6. september 2014

Innhold

1 Introduksjon

I dette prosjektet har me tatt for oss ein ein-dimensjonal Poisson likning med Dirichlet randpunkt. Me vil simulere ein numerisk løysning på ein andre-ordens differensiallikning. Måten me vil gjer dette på er ved å omforme settet med lineære likningar til ei matriselikning. Då kan me bruke radoperasjoner til å lage ein algoritme som gjer det mogleg for oss å løyse likningssettet. Denne matriselikninga vil gje oss ei tridiagonalmatrise A som stort sett består av nullar. Hensikta er då å sjå at me kan kaste"alle nullane og kun behalde elementa frå diagonalane. Me vil representere desse som vektorar for å bruke minst mogleg plass. Dette vil og gjere det mogleg for oss å kunne utføre fleire iterasjoner. Ein vanleg PC vil ikkje kunne klare å representere ein stort større matrise enn 1000×1000. Dette fordi han ikkje har stort nok minne. Viss me derimot bruker vektorar vil me kunne ha ei "matrise"(tre vektorar) som er mykje større. I tillegg vil radreduksjonen gå fleirfoldige mange gonger kjappare.

2 Omforming av differensiallikning til ei matriselikning

Me er interesserte i å forme om uttrykket for tilnærminga til den andrederiverte. Me startar med å fjerne brøken. Då får me

$$-\frac{v_{i+1} + v_{i-1} - 2v_i}{h^2} = f_i \quad \Rightarrow \quad -v_{i+1} + v_{i-1} - 2v_i = h^2 f_i = \tilde{b}_i, \ \forall \ i \in [1, n].$$

Me er no interesserte i å skrive dette om til vektorar og ei matrise. Me kan då skrive det som

$$\begin{aligned}
&(2, -1) \cdot (v_1, v_2) = \tilde{b}_1, \\
&(-1, 2, -1) \cdot (v_1, v_2, v_3) = \tilde{b}_2, \\
&(-1, 2, -1) \cdot (v_2, v_3, v_4) = \tilde{b}_3, \\
&\dots \\
&(-1, 2, -1) \cdot (-v_{i+1}, -v_{i-1}, v_i) = \tilde{b}_i, \\
&\dots \\
&(-1, 2) \cdot (v_{n-1}, v_n) = \tilde{b}_n.
\end{aligned}$$

Det kjem fram frå dette uttrykket at koeffisientane står i ro. Viss me no setter koeffisientane i ei matrise vil me kun trenge ein vektor $\mathbf{v}$ beståande av alle $v_i, \ \forall \ i \in [1, n]$ kor dei vil få dei rette koeffisientane i eit matriseprodukt.

Me setter opp matriselikninga.

$$A\mathbf{v} = \begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \dots \\ 0 & -1 & 2 & -1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \dots \\ v_{n-1} \\ v_n \end{pmatrix} = \begin{pmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \\ \dots \\ \tilde{b}_{n-1} \\ \tilde{b}_n \end{pmatrix} = \tilde{\mathbf{b}}$$

Eit matriseprodukt av dette uttrykket vil gje oss

$$\begin{aligned} &2v_1 - v_2 = \tilde{b}_1 \\ &-v_1 + 2v_2 - v_3 = \tilde{b}_2 \\ &\dots \\ &-v_{i-1} + 2v_i - v_{i+1} = \tilde{b}_i \\ &\dots \\ &-v_{n-2} + 2v_{n-1} - v_n = \tilde{b}_{n-1} \\ &-v_{n-1} + 2v_n = \tilde{b}_n. \end{aligned}$$

Dette fordi koeffisientane er omringa av nullar som automatisk vil fjerne resten av ledda frå $\mathbf{v}$.

3 Algoritmen

For å løyse ei matriselikning bruker me radoperasjoner på matrisa. Desse vil igjen bli utførde på løysningsvektoren (her kalla $\tilde{\mathbf{b}}$). Målet med likninga er å omforme matrisa til ei spesiell øvre- og nedrematrise som me kan løyse.

3.1 Dekomponering

I den fyrste matrisa vil me fjerne alt som er over diagonalen slik at ho utelukkande består av einarar på diagonalen og $\gamma_i, \ \forall\ i \in [2, n]$ på diagonalen under. I den andre vil me ha ei matrise beståande utelukkande av $\lambda_i, \ \forall\ i \in$

$[1, n]$ på diagonalen og $c_i, \ \forall\ i \in [1, n-1]$ på diagonalen over. Då får me

$$A = \begin{pmatrix} b_1 & c_1 & \dots & \dots & \dots & \dots \\ a_2 & b_2 & c_2 & \dots & \dots & \dots \\ \dots & a_3 & b_3 & c_3 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & a_{n-1} & b_{n-1} & c_{n-1} \\ \dots & \dots & \dots & \dots & a_n & b_n \end{pmatrix} =$$

$$LU = \begin{pmatrix} 1 & \dots & \dots & \dots & \dots & \dots \\ \gamma_2 & 1 & \dots & \dots & \dots & \dots \\ \dots & \gamma_2 & 1 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \gamma_{n-1} & 1 & \dots \\ \dots & \dots & \dots & \dots & \gamma_n & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 & c_1 & \dots & \dots & \dots & \dots \\ \dots & \lambda_2 & c_2 & \dots & \dots & \dots \\ \dots & \dots & \lambda_3 & c_3 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \lambda_{n-1} & c_{n_1} \\ \dots & \dots & \dots & \dots & \dots & \lambda_n \end{pmatrix}.$$

Multiplikasjon av matrisene LU vil gje oss A. Me bruker dette til å finne eit uttrykk for $\gamma_i, \ \forall\ i \in [2, n]$ og $\lambda_i, \ \forall\ i \ \in [1, n]$. Det vil gje oss

$$\begin{aligned} &\lambda_1 = b_1, \\ &\lambda_1 \gamma_2 = a_2, \\ &c_1 \gamma_2 + \lambda_2 = b_2, \\ &\dots \\ &\gamma_i = \frac{a_i}{\lambda_{i-1}}, \\ &\lambda_i = b_i - c_{i-1}\gamma_i, \\ &\dots \\ &\gamma_n = \frac{a_n}{\lambda_{n-1}}, \\ &\lambda_n = b_n - c_{n-1}\gamma_n. \end{aligned}$$