29 september, 2014, Föreläsning 7

Tillämpad linjär algebra

Innehåll

- LU-faktorisering med partiell pivotering
- Beräkningskostnad - komplexitet
- Noggrannhet vid numerisk lösning av linjära ekvationssystem
- Konditionstal, vektor och matrisnorm

1. LU-faktorisering och partiell pivotering

Vid LU-faktoriseringen av en matris A så gör vi en Gauss-elimination där vi strävar efter att noll-ställa alla element under huvuddiagonalen. Första steget ges av

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{subtrahera } \frac{a_{21}}{a_{11}} \times \text{ rad 1 från rad 2} \Rightarrow \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} - \frac{a_{21}}{a_{11}} a_{12} \end{bmatrix}$$

Vad händer om $a_{11} = 0$?

För att undvika division med noll eller små tal kan man behöva byta plats på raderna i matrisen. Det kallas att göra en **partiell pivotering**. Om vi måste göra ett radbyte under en LU-faktorisering får vi istället en $PA = LU$-faktorisering där P är en så kallad **permutationsmatris**.

En permutationsmatris är en $n \times n$-matris som bara innehåller nollor förutom en etta i varje rad och kolonn. Exempel på några 3×3-permutationsmatriser

$$P_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad P_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Vi kan se en permutationsmatris, P, som en identitetsmatris där vissa av raderna har bytt plats. Om vi multiplicerar P med en matris A kommer produkten, PA, bli densamma som om vi bytt motsvarande rader i A.

2. **Uppgift.** Hur många 3×3-permutationsmatriser finns det och hur ser de ut?

3. **Uppgift.** Låt

$$A_1 = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 5 & 4 \\ 7 & 8 & 9 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 7 & 8 & 9 \\ 1 & 2 & 3 \\ 6 & 5 & 4 \end{bmatrix}$$

Vad blir P_1A_1 och P_2A_1? Om P_1 och P_2 är matriserna ovan.

Om $A_2 = PA_1$, hur ser P ut?

En LU-faktorisering med partiell pivotering ges av $PA = LU$ och lösningsalgoritmen för att lösa $A\vec{x} = \vec{b}$ ges av

(1) $PA\vec{x} = P\vec{b} \Rightarrow LU\vec{x} = P\vec{b}$.
(2) Låt $\vec{y} = U\vec{x}$ och lös det undertriangulära systemet $L\vec{y} = P\vec{b}$ genom **framåtsubstitution** $\Rightarrow \vec{y}$.
(3) Lös nu det övertriangulära systemet $U\vec{x} = \vec{y}$ genom **bakåtsubstitution** $\Rightarrow \vec{x}$.

4. **Uppgift.** Lös följande system med $PA = LU$-faktorisering

$$\begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 6 \end{bmatrix}$$

5. **Uppgift.** Lös följande system med $PA = LU$-faktorisering

$$\begin{bmatrix} 3 & 1 & 2 \\ 6 & 3 & 4 \\ 3 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}$$

6. Beräkningskostnad - komplexitet

När vi löser ett problem numeriskt med hjälp av en dator är beräkningskostnaden intressant. Beräkningskostnaden för en datoralgoritm mäts vanligtvis i antalet flyttalsoperationer (**flop**). En flop är en multiplikation, division, addition eller subtraktion.

Vanligtvis vill man veta hur antalet flops beror av storleken på problemet som man vill lösa (problemets **komplexitet**). Antag att vi har ett $n \times n$-system, $A\vec{x} = \vec{b}$, då gäller

- Gauss elimination: $O(n^3)$
- LU-faktorisering av A: $O(n^3)$
- Framåtsubstitution: $O(n^2)$

- Bakåtsubstitution: $O(n^2)$
- Beräkna inversen av A genom $[A|I] \Rightarrow [I|A^{-1}]$: $O(n^3)$
- Beräkna inversen av $A^{-1}\vec{b}$: $O(n^3)$
- LU-faktorisering av A om A är en *gles bandmatris*: $O(n)$
- Framåtsubstitution för en glas bandmatris: $O(n)$
- Bakåtsubstitution för en gles bandmatris: $O(n)$

Om vi vet hur många flop/s (flops) en dator gör kan vi få en uppskattning av beräkningskostnaden i tid.

7. **Uppgift.** Antag att det tar 0.2 sekunder att Gauss-eliminera en $n \times n$-matris A på din dator när $n = 1000$. Hur lång tid skulle det ta att Gauss-eliminera ett matris med $n = 10000$ obekanta?

8. **Uppgift.** Antag att det tar 0.005 sekunder att göra en bakåtsubstitution med 5000 obekanta på en dator. Hur lång tid skulle det ta att göra en Gauss-elimination av en $n \times n$-matris där $n = 5000$?

9. Noggrannhet vi numerisk lösning av linjära ekvationssystem

Konditionstalet hos ett problem säger oss något om hur fel fortplantar sig genom problemet. Om ett litet fel i indata producerar ett litet fel i utdata så har problemet ett litet konditionstal och problemet är **välkonditionerat**. Om små fel in ger stora fel ut är konditionstalet stort och problemet är **illa-konditionerat**.

Vid lösning av ett linjärt ekvationssystem, $A\vec{x} = \vec{b}$ gäller följande samband mellan det relativa felet i lösningen och det relativa felet i högerledet

$$\frac{\|\vec{x}_{ref} - \vec{x}\|_\infty}{\|\vec{x}_{ref}\|_\infty} \leq \text{cond}(A)\frac{\|\vec{b}_{ref} - \vec{b}\|_\infty}{\|\vec{b}_{ref}\|_\infty}.$$

Konditionstalet ges av

$$\text{cond}(A) = \|A\|_\infty \|A^{-1}\|_\infty$$

där $\|A\|_\infty$ är *maxnormen* av A (se nedan).

I MATLAB kan vi beräkna konditionstalet för en matris A med `cond(A,inf)`.

10. Matris och vektornorm

Vi har tidigare sett att vi kan mäta storleken av en vektor med hjälp av vektorns längd. Ett annat ord för längd i det här sammanhanget är **norm**. Vi kan även beräkna normen av en matris. Det finns flera olika val av normer

$$\|\vec{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \quad \text{Euklidiska normen}$$
$$\|\vec{x}\|_\infty = \max_j |x_j| \quad \text{Maxnorm}$$
$$\|A\|_\infty = \max_i \sum_{j=1}^{n} |a_{ij}| \quad \text{Maxnorm}$$

11. **Uppgift.** Beräkna maxnormen av följande matriser

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 1 & 5 & 1 \\ -1 & 2 & -3 \\ 1 & -7 & 0 \end{bmatrix}$$

12. **Uppgift.** Beräkna konditionstalet (i maxnorm) för följande matriser

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 6 & 3 \\ 4 & 2 \end{bmatrix}$$