Numerisk lineær algebra.

Problem: Løs $A\vec{x} = \vec{b}$, $A \in \mathbb{R}^{n\times n}$, inverterbar, $\vec{x}, \vec{b} \in \mathbb{R}^n$?

Metode 1: Naiv Gauss eliminasjon

Gitt: $E_1: a_{11}x_1 + a_{12}x_2 + \cdots a_{1n}x_n = b_1$

$E_2: a_{21}x_1 + a_{22}x_2 + \cdots a_{2n}x_n = b_2$

$\vdots$

$E_n: a_{n1}x_1 + a_{n2}x_2 + \cdots a_{nn}x_n = b_n$

Behold ligning E_1. Eliminer det første leddet i alle de andre ligningene ved

$$E_i^{(2)} \leftarrow E_i - \frac{a_{i1}}{a_{11}} \cdot E_1 \quad , \quad i = 1, \cdots, n$$

dvs: $m_{i1} = \dfrac{a_{i1}}{a_{11}}$ (multiplikatorer), $i = 2, \cdots, n$

$$a_{ij}^{(2)} = a_{ij} - m_{i1}a_{1j}, \quad b_i^{(2)} = b_i - m_{i1}b_1, \quad i,j = 2, \cdots, n.$$

slik at vi får systemet:

$$\begin{aligned}
E_1 &: a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
E_2^{(2)} &: \quad\quad a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)} \\
&\vdots \\
E_n^{(2)} &: \quad\quad a_{n2}^{(2)}x_2 + \cdots + a_{nn}^{(2)}x_n = b_n^{(2)}
\end{aligned}$$

Gjenta dette på det reduserte systemet $E_2^{(2)}, \cdots E_n^{(2)}$, og så videre, inntil vi har oppnådd et øvre triangulært system:

$$\begin{aligned}
E_1 &: a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
E_2^{(2)} & \quad\quad a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)} \\
&\vdots \\
E_{n-1}^{(n-1)} &: \quad\quad a_{n-1,n-1}^{(n-1)}x_{n-1} + a_{n-1,n}^{(n-1)}x_n = b_{n-1}^{(n-1)} \\
E_n^{(n)} &: \quad\quad a_{n,n}^{(n)}x_n = b_n^{(n)}
\end{aligned}$$

Og dette er lett å løse. Start nederst:

$$x_n = b_n^{(n)} / a_{nn}^{(n)}$$

$$x_{n-1} = (b_{n-1}^{(n-1)} - a_{n-1,n}^{(n-1)} x_n) / a_{n-1,n-1}^{(n)}$$

$$\vdots$$

$$x_1 = (b_1 - \sum_{j=2}^{n} a_{1j} x_j) / a_{11}$$

Dette kan skrives opp som en algoritme:

Gauss-eliminasjon:

Gitt $A, \vec{b}$.

for $k = 1, 2, \cdots, n-1$

  for $i = k+1, \cdots, n$

    $m_{ik} = a_{ik} / a_{kk}$

    for $j = k+1, \cdots, n$

      $a_{ij} \leftarrow a_{ij} - m_{ik} a_{kj}$

    $b_i \leftarrow b_i - m_{ik} b_k$

Tilbakeløsning:

$x_n = b_n / a_{nn}$

for $i = n-1, \cdots, 1$

  $x_i = (b_i - \sum_{j=i+1}^{n} a_{ij} x_j) / a_{ii}$

Notasjon:

m_{ij} : Multiplikatorer.

a_{kk} : Pivot-elementer.

Eksempel

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 3 \\ 3x_1 + 4x_2 &= 3 \\ 2x_1 + 10x_2 + 4x_3 &= 10 \end{aligned} \iff \begin{array}{c} \begin{array}{cc} A & \quad b \end{array} \\ \left[\begin{array}{ccc|c} 1 & 2 & 1 & 3 \\ 3 & 4 & 0 & 3 \\ 2 & 10 & 4 & 10 \end{array}\right] \end{array}$$

$a_{11} = 1$
$m_{21} = 3$
$m_{31} = 2$

$k=1$:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 3 \\ 0 & -2 & -3 & -6 \\ 0 & 6 & 2 & 4 \end{array}\right] \qquad \begin{array}{l} a_{22}^{(2)} = -2 \\ m_{31} = -3 \end{array}$$

$k=2$:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 3 \\ 0 & -2 & -3 & -6 \\ 0 & 0 & -7 & -14 \end{array}\right]$$

Tilbakeløsning:
$$x_3 = -{}^{14}/_{-7} = 2$$
$$x_2 = (-6 + 3\cdot 2)/-2 = 0$$
$$x_1 = (3 - 2\cdot 0 + 2)/1 = 1$$

Tilbake til det generelle problemet: $A\vec{x} = \vec{b}$.

La

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ m_{n1} & & m_{n-1,n} & 1 \end{bmatrix}$$

m_{ij} : multiplikatorer

Nedre triangulær, med 1 på diagonalen

$$U = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{nn}^{(n)} \end{bmatrix}$$

Etter Gauss-eliminasjonen

Øvre triangulær, med pivot-elementene på diagonalen

Det kan vises at

$$A = L\cdot U \qquad : \text{LU-faktorisering.}$$

Å løse et lineært ligningssystem $A\vec{x} = \vec{b}$ kan deles i tre:

Gauss-eliminasjonen	:	$A = L\cdot U$
Forover løsning	:	$L\vec{y} = \vec{b}$
Bakover løsning	:	$U\cdot\vec{x} = \vec{y}$

Kompleksitet av algoritmer.

I de fleste store, numeriske simuleringer er det lineær algebraen som tar mest CPU-tid. Det gjelder altså operasjoner av typen $\vec{x}^T\vec{y}$, $A\vec{x}$, $A+B$, osv., og å løse ligninger $A\vec{x}=\vec{b}$.
Det er dermed interessant å vite noe om hvor mye arbeidsmengden øker når størrelsen på problemet øker. Vi kan måle dette i antall flyttalloperasjoner (flop):

1 flop er 1 addisjon og en multiplikasjon
eller 1 divisjon.

Eksempel: Matrise-vektor operasjon.

$$\vec{y} = A\vec{x}, \quad A \in \mathbb{R}^{n\times n}, \quad \vec{x} \in \mathbb{R}^n$$

$$\vec{y}_i = \sum_{j=1}^{n} a_{ij}x_j, \quad i = 1,\cdots,n \quad : \quad \left.\begin{array}{l} n^2 \text{ multiplikasjoner} \\ n(n-1) \text{ addisjoner} \end{array}\right\} \sim n^2 \text{ flop}$$

LU-faktorisering og tilbakeløsning:

$$A = L\cdot U: \quad \underbrace{\sum_{k=1}^{n-1}(n-k)^2}_{a_{ij}:\ +,\ *} + \underbrace{\sum_{k=1}^{n}(n-k)}_{m_{ij}:\ /} = \frac{1}{3}n^3 - \frac{1}{3}n \sim \frac{1}{3}n^3 \text{ flop.}$$

$$\left.\begin{array}{ll} Ly = b : & \sum_{k=2}^{n}(k-1) = \tfrac{1}{2}n^2 - \tfrac{1}{2}n \\ Ux = b : & \sum_{k=2}^{n}(k-1) + n = \frac{1}{2}n^2 + \frac{1}{2}n \end{array}\right\} \sim \frac{1}{2}n^2 \text{ flop hver.}$$

Nar LU-faktoriseringen først er funnet, krever tilbakeløsningen bare $\sim n^2$ flop, like mye som <u>en</u> matrise-vektor multiplikasjon.

Nyttig hvis en skal løse mange ligninger med samme A, men ulike høyresider.

Når har vi en LU-faktorisering?

Når pivotelementene $a_{kk}^{(k)} \neq 0$, men kan vi vite det allerede før prosessen starter. Svaret er, noen ganger, men ikke alltid.

A er strengt diagonal dominant hvis (s.d.d)

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \quad , \quad i = 1, \cdots, n$$

Teorem: Hvis A er strengt diagonaldominant så har A en LU-faktorisering, og A er inverterbar.

Bevis: La $A \in \mathbb{R}^{n\times n}$ være s.d.d.
Vi ønsker å vise at s.d.d. er en egenskap som bevares gjennom eliminasjonsprosessen. Det holder å vise det for første steg, dvs. vise at

Bruker
$|x+y| \leq |x| + |y|$ (t1)
$|x-y| \geq |x| - |y|$ (t2)

$$(*) \qquad |a_{ii}^{(2)}| > \sum_{\substack{j=2 \\ j\neq i}}^{n} |a_{ij}^{(2)}| \quad , \quad i = 2, \cdots, n$$

der $a_{ij}^{(2)} = a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j}$

Vi har

$$\sum_{\substack{j=2 \\ j\neq i}}^{n} |a_{ij}^{(2)}| = \sum_{\substack{j=2 \\ j\neq i}}^{n} \left| a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j} \right| \overset{(t1)}{\leq} \sum_{\substack{j=2 \\ j\neq i}}^{n} \left(|a_{ij}| + \frac{|a_{i1}|}{|a_{11}|} \cdot |a_{1j}| \right)$$

$$= \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| - |a_{i1}| + \frac{|a_{i1}|}{|a_{11}|} \cdot \sum_{j=2}^{n} |a_{1j}| - |a_{1i}|$$

$$\overset{\text{s.d.d.}}{<} |a_{ii}| - |a_{i1}| + \frac{|a_{i1}|}{|a_{11}|} \cdot |a_{11}| - \frac{|a_{i1}|}{|a_{11}|} \cdot |a_{1i}|$$

$$= |a_{ii}| - \frac{|a_{i1}|}{|a_{11}|} \cdot |a_{1i}| \overset{(t2)}{\leq} \left| a_{ii} - \frac{a_{i1}}{a_{11}} \cdot a_{1i} \right| = |a_{ii}^{(2)}|$$

Så $|a_{ii}^{(2)}| > \sum_{\substack{j=2 \\ j\neq i}}^{n} |a_{ij}^{(2)}|$. Fortsett slik, så vil

$$|a_{ii}^{(k)}| > \sum_{\substack{j=k \\ j\neq i}}^{n} |a_{ij}^{(k)}| \quad , \quad i = k, \cdots, n$$

for $k = 2, 3, \cdots, n$.

Pivotering:

Gauss-eliminasjons algoritmen bryter sammen hvis et pivot-element $a_{kk}^{(k)} = 0$, selv om A er inverterbar.
Men problemer kan også oppstå pga. avrundingsfeil, som følgende eksempel demonstrerer:

Eksempel:
Anta at tall representeres med 3 signifikante sifre, dvs. $y \in \mathbb{R}$ lagres som $y = \pm d_1 . d_2 d_3 \cdot 10^e$, $d_i \in \{0, \dots, 9\}$, $d_1 \neq 0$.
Løs systemet

$$\begin{aligned} -10^{-3} x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2 \end{aligned} \quad , \left(\begin{aligned} x_1 &= 0.998 \\ x_2 &= 1.001 \end{aligned} \right)$$

på en maskin som lagrer tall med denne representasjonen.

$$m_{21} = -1.00 \cdot 10^{+3}, \quad a_{22}^{(2)} = 1.00 + 1.00 \cdot 10^3 = 1.00\cancel{1} \cdot 10^3$$
$$b_2^{(2)} = 2.00 + 1.00 \cdot 10^3 = 1.00\cancel{2} \cdot 10^3$$

Tilbakeløsning: $x_2 = \dfrac{1.00 \cdot 10^3}{1.00 \cdot 10^3} = 1.00$, $\quad x_1 = \dfrac{1.00 - 1.00}{-1.00 \cdot 10^{+3}} = 0.00$

Bytt om rekkefølgen:
$$\begin{aligned} x_1 + x_2 &= 2 \\ -10^{-3} x_1 + x_2 &= 1 \end{aligned}$$

$$m_{21} = -1.00 \cdot 10^{-3}, \quad a_{22}^{(2)} = 1.00 + 1.00 \cdot 10^{-3} = 1.00\cancel{1} \cdot$$
$$b_2^{(2)} = 1.00 + 2.00 \cdot 10^{-3} = 1.00\cancel{2} \cdots$$

Tilbakeløsning: $x_2 = \dfrac{1.00}{1.00} = 1.00$, $\quad x_1 = 2.00 - 1.00 = 1.00$

Hva skjer? Etter gauss-eliminasjonen har vi

$$\begin{aligned} a_{11} x_1 + a_{12} x_2 &= b_1 \\ a_{22}^{(2)} x_2 &= b_2^{(2)} \end{aligned} \quad \rightarrow \tilde{x}_2 = x_2 + \varepsilon_2 \quad (\varepsilon \text{ avr. feil})$$

$$\tilde{x}_1 = \frac{b_1 - a_{12} \tilde{x}_2}{a_{11}} = \underbrace{\frac{b_1 - a_{12} x_2}{a_{11}}}_{x_1} - \underbrace{\frac{a_{12}}{a_{11}} \varepsilon_2}_{\varepsilon_1 \text{ (feil i } x_1)}$$

For å unngå slike problemer ønsker vi å bytte rekkefølgen av ligningene, slik at pivot-elementene blir store. (og ikke små).

Strategier: Ikke gjennomgått. Les selv!

- Full pivotering: For hvert eliminasjonssteg k, finn $\max_{i,j \leq k \leq n} |a_{ij}^{(k)}|$ og bruk dette som pivotelement. (bytt rader og kolonner.
  Brukes vanligvis ikke.

- Delvis radvis pivotering (vanligst).
  For hver k, finn den minste p en slik at $|a_{pk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|$.
  Bytt rad i og p.

  Men delvis pivotering løser ikke problemet for ligningen:

$$
\begin{aligned}
-x_1 + 10^3 x_2 &= 10^3 \\
x_1 + x_2 &= 2
\end{aligned}
$$

Revidert fra her:

men det ville vært løst om hver ligning ble skalert, dvs. hver rad i A og b deles med $\max_j |a_{ij}|$

- Skalert delvis pivotering gjør velger pivotrader som om denne skaleringen er gjort! Og det gjøres slik:

  - Før eliminasjonsprosessen: La $s \in \mathbb{R}^n$ med elementene

$$
s_i = \max_j |a_{ij}|
$$

  - For hvert eliminasjonssteg k, finn q slik at

$$
\frac{|a_{qk}^{(k)}|}{s_q} = \max_{k \leq i \leq n} \frac{|a_{ik}^{(k)}|}{s_i}
$$

    dvs. q er raden der $\frac{|a_{ik}^{(k)}|}{s_i}$ er størst.

Bytt rad k og q.

I praksis bytter man ikke om radene.
Isteden holder man orden på rekkefølgen vha.
en pivot vektor $\vec{p}$.

- Start med $\vec{p} = [1, 2, \cdots, n]$
- Erstatt alle radindekser i med p_i $(a_{p_i,j}\ ,\ b_{p_i})$
- For hvert eliminasjonssteg k,
  finn pivotraden q, og bytt p_k og p_q.

Permutasjons matriser:

P: permutasjoner av identitetsmatrisa I_n:
I hver rad og kolonne er det et og bare et element lik 1, resten er 0.

$P \cdot A$: Endrer rekkefølgen på radene
AP : Endrer rekkefølgen på kolonnene.

Det kan vises at:

Hvis A er inverterbar, så fins en
- Permutasjonsmatrise P
- Nedre triangulær matrise L, med 1 på diagonalen
- Øvre triangulær matrise U, $u_{kk} \neq 0$

slik at

$$PA = L \cdot U$$