NUMERIČKE METODE I MATEMATIČKO MODELIRANJE

7. PREDAVANJE

LINEARNA ALGEBRA U MATEMATIČKOM MODELIRANJU I NUMERIČKIM METODAMA

RJEŠAVANJE SUSTAVA LINEARNIH JEDNADŽBI

SUSTAVI LINEARNIH JEDNADŽBI

- algoritmi za rješavanje skupova linearnih jednadžbi zasnovani na Gaussovoj eliminaciji
- dekompozicija matrice u produkt gornje i donje trokutne matrice, primjena u rješavanju linearnih jednadžbi
- proračun inverzne matrice, određivanje determinante
- Ilustrativni primjer problema koji zahtjeva rješavanje sustava linearnih jednadžbi: problem rubnih vrijednosti

$$-\frac{d^2u(x)}{dx^2} = f(x, u(x))$$

$$x \in [a, b] \qquad u(a) = u(b) = 0 \longleftarrow \text{RUBNI UVJETI}$$

SUSTAVI LINEARNIH JEDNADŽBI

- druga derivacija se može aproksimirati sa

$$f'' = \frac{f_h - 2f_0 + f_{-h}}{h^2} + O(h^2)$$

- gdje je interval $x \in [a, b]$ podijeljen na n podintervala, korak iznosi $h = (b - a)/(n + 1)$

- funkcija se računa za vrijednosti $x_i = ih$, $i = 0, 1, \ldots, n + 1$
- diferencijalni operator jednadžbe:

$$u''(x_i) \approx \frac{u(x_i + h) - 2u(x_i) + u(x_i - h)}{h^2}$$

$$u_i^{''} \approx \frac{u_{i+1} - 2u_i + u_{i-i}}{h^2}$$

SUSTAVI LINEARNIH JEDNADŽBI

- početna diferencijalna jednadžba se može zapisati kao diskretizirana jednadžba sa aproksimiranim derivacijama

$$-\frac{u_{i+1} - 2u_i + u_{i-i}}{h^2} = f(x_i, u(x_i))$$

- ovdje je $i = 1, 2, \ldots, n$

- rubni uvjeti su $\begin{cases} u(a) = u_0 \\ u(b) = u_{n+1} \end{cases}$

- problem koji treba riješiti se može raspisati u matričnom obliku kao sustav linearnih jednadžbi

$$\mathbf{Au} = \mathbf{f}(\mathbf{u})$$

SUSTAVI LINEARNIH JEDNADŽBI

$$\mathbf{A} = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & \dots & \dots & \dots & \dots & \dots \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}$$

- odgovarajući vektori:

$$\mathbf{u} = (u_1, u_2, \dots, u_n)^T$$

$$\mathbf{f}(\mathbf{u}) = f(x_1, x_2, \dots, x_n, u_1, u_2, \dots, u_n)^T$$

- Dakle, treba riješiti sustav n linearnih jednadžbi

GAUSSOVA ELIMINACIJA

- treba riješiti skup linearnih jednadžbi $\mathbf{A}\mathbf{x} = \mathbf{w}$ uz pretpostavku da matrica A nije singularna i matrični elementi na dijagonali $a_{ii} \neq 0$

- kao primjer uzmimo matricu $\mathbf{A} \in \mathbb{R}^{4\times 4}$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix}$$

- Gaussova eliminacija polazi od primjene prve jednadžbe da se eliminira nepoznanica x_1 iz preostalih $n-1$ jednadžbi
- zatim se pomoću druge jednadžbe eliminira nepoznanica x_2 itd...

GAUSSOVA ELIMINACIJA

- nakon n-1 eliminacija dobije se gornja trokutasta matrica

$$\begin{aligned} b_{11}x_1 + b_{12}x_2 + b_{13}x_3 + b_{14}x_4 &= y_1 \\ b_{22}x_2 + b_{23}x_3 + b_{24}x_4 &= y_2 \\ b_{33}x_3 + b_{34}x_4 &= y_3 \\ b_{44}x_4 &= y_4 \end{aligned}$$

- sustav jednadžbi se sada može riješiti rekurzivno, polazeći od x_4 , tzv. supstitucija unazad

$$x_m = \frac{1}{b_{mm}}\left(y_m - \sum_{k=m+1}^{n} b_{mk}x_k\right)$$

$$m = n-1, n-2, \ldots, 1$$

GAUSSOVA ELIMINACIJA

- da bi se dobila trodijagonalna matrica, prva jednadžba množi se sa a_{j1}/a_{11} i onda oduzima od j-te jednadžbe, $j = 2, n$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & (a_{22} - \frac{a_{21}a_{12}}{a_{11}}) & (a_{23} - \frac{a_{21}a_{13}}{a_{11}}) & (a_{24} - \frac{a_{21}a_{14}}{a_{11}}) \\ 0 & (a_{32} - \frac{a_{31}a_{12}}{a_{11}}) & (a_{33} - \frac{a_{31}a_{13}}{a_{11}}) & (a_{34} - \frac{a_{31}a_{14}}{a_{11}}) \\ 0 & (a_{42} - \frac{a_{41}a_{12}}{a_{11}}) & (a_{43} - \frac{a_{41}a_{13}}{a_{11}}) & (a_{44} - \frac{a_{41}a_{14}}{a_{11}}) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} y_1 \\ w_2^{(2)} \\ w_3^{(2)} \\ w_4^{(2)} \end{pmatrix}$$

$$\begin{aligned} b_{11}x_1 + b_{12}x_2 + b_{13}x_3 + b_{14}x_4 &= \quad y_1 \\ a_{22}^{(2)}x_2 + a_{23}^{(2)}x_3 + a_{24}^{(2)}x_4 &= \quad w_2^{(2)} \\ a_{32}^{(2)}x_2 + a_{33}^{(2)}x_3 + a_{34}^{(2)}x_4 &= \quad w_3^{(2)} \\ a_{42}^{(2)}x_2 + a_{43}^{(2)}x_3 + a_{44}^{(2)}x_4 &= \quad w_4^{(2)} \end{aligned}$$

$$b_{1k} = a_{1k}^{(1)} \quad k = 1, \ldots, n \qquad\qquad a_{jk}^{(2)} = a_{jk}^{(1)} - \frac{a_{j1}^{(1)} a_{1k}^{(1)}}{a_{11}^{(1)}} \quad j, k = 2, \ldots, n$$

GAUSSOVA ELIMINACIJA

- vektor sa desne strane jednadžbe:

$$y_1 = w_1^{(1)}$$

$$w_j^{(2)} = w_j^{(1)} - \frac{a_{j1}^{(1)} w_1^{(1)}}{a_{11}^{(1)}} \qquad j = 2, \ldots, n$$

- općeniti izrazi:

$$a_{jk}^{(m+1)} = a_{jk}^{(m)} - \frac{a_{jm}^{(m)} a_{mk}^{(m)}}{a_{mm}^{(m)}} \qquad j, k = m+1, \ldots, n$$

$$w_j^{(m+1)} = w_j^{(m)} - \frac{a_{jm}^{(m)} w_m^{(m)}}{a_{mm}^{(m)}} \qquad j = m+1, \ldots, n$$

GAUSSOVA ELIMINACIJA

- moguća greška zbog ograničene preciznosti u članovima tipa $(a_{22} - a_{21}a_{12}/a_{11}) \rightarrow$ potrebno je ukloniti male dijagonalne elemente permutacijama redova i stupaca matrice

- npr. u slučaju skupa linearnih jednadžbi

$$\begin{pmatrix} 1 & 3 & 4 & 6 \\ 0 & 10^{-8} & 198 & 19 \\ 0 & -91 & 51 & 9 \\ 0 & 7 & 76 & 541 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$$

mali element na dijagonali se može ukloniti permutacijom:

$$\begin{pmatrix} 1 & 3 & 4 & 6 \\ 0 & -91 & 51 & 9 \\ 0 & 10^{-8} & 198 & 19 \\ 0 & 7 & 76 & 541 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_3 \\ y_2 \\ y_4 \end{pmatrix}$$

GAUSSOVA ELIMINACIJA

- najbolja numerička preciznost rezultata može se dobiti permutiranjem redova i stupaca tako da numerički najveće vrijednosti u podmatrici koja još nije obrađena budu postavljene na dijagonalu

- Gaussova eliminacija zahtjeva velik broj "floating-point" operacija (flop) :n x n matrica zahtjeva $2n^3/3 + O(n^2)$ flop

- ako procesor izvodi 10^9 flop u sekundi (flops), za matricu n=10000 potrebno je 1000 sekundi

- algoritam Gaussove eliminacije je rijetko korišten u proračunu determinante $\rightarrow$ češće u primjeni je tzv. LU dekompozicija

LU DEKOMPOZICIJA MATRICE

- L("LOWER")U("UPPER") dekompozicija, poznata i kao Crout ili Doolittle faktorizacija matrice

- dekompozicija matrice *A* pomoću matrice B sa elementima ispod dijagonale ("Lower") i matrice C koja ima dijagonalne elemente i elemente iznad dijagonale ("Upper")

$$\mathbf{A} = \mathbf{BC} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ b_{21} & 1 & 0 & 0 \\ b_{31} & b_{32} & 1 & 0 \\ b_{41} & b_{42} & b_{43} & 1 \end{pmatrix} \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ 0 & c_{22} & c_{23} & c_{24} \\ 0 & 0 & c_{33} & c_{34} \\ 0 & 0 & 0 & c_{44} \end{pmatrix}$$

- LU dekompozicija se koristi u nizu algoritama linearne algebre, kao i u rješavanju sustava linearnih jednadžbi

LU DEKOMPOZICIJA MATRICE

- Matrica A se može LU faktorizirati ako je njezina determinanta različita od 0

- Ako postoji LU faktorizacija i matrica A nije singularna, onda je LU faktorizacija jedinstvena i determinanta matrice A je dana sa

$$det\{\mathbf{A}\} = c_{11}c_{22}\dots c_{nn}$$

- Algoritam za izračun matrica B i C u primjeru 4x4 matrice: kreće se od prvog stupca A matrice:

$$\begin{aligned} a_{11} &= c_{11} \\ a_{21} &= b_{21}c_{11} \\ a_{31} &= b_{31}c_{11} \\ a_{41} &= b_{41}c_{11} \end{aligned} \quad \Longrightarrow \quad c_{11}, b_{21}, b_{31}, b_{41}$$

LU DEKOMPOZICIJA MATRICE

- za drugi stupac A matrice:

$$\begin{aligned} a_{12} &= c_{12} \\ a_{22} &= b_{21}c_{12} + c_{22} \\ a_{32} &= b_{31}c_{12} + b_{32}c_{22} \\ a_{42} &= b_{41}c_{12} + b_{42}c_{22} \end{aligned} \quad \Rightarrow \quad c_{12}, c_{22}, b_{32}, b_{42}$$

- kada se prelazi iz prvog na drugi stupac, više ne trebamo a_{i1} elemente, općenito svojstvo kroz čitav algoritam

- generalizirana procedura se može zapisati kao

$$\begin{aligned} i < j: \quad & b_{i1}c_{1j} + b_{i2}c_{2j} + \cdots + b_{ii}c_{ij} = a_{ij} \\ i = j: \quad & b_{i1}c_{1j} + b_{i2}c_{2j} + \cdots + b_{ii}c_{jj} = a_{ij} \\ i > j: \quad & b_{i1}c_{1j} + b_{i2}c_{2j} + \cdots + c_{ij}c_{jj} = a_{ij} \end{aligned}$$

LU DEKOMPOZICIJA MATRICE

Sumarni algoritam za LU dekompoziciju po stupcima (j):

- 1) izračunati prvi element $c_{1j} = a_{1j}$
- 2) izračunati elemente $c_{ij}, i = 2, \ldots, j-1$

$$c_{ij} = a_{ij} - \sum_{k=1}^{i-1} b_{ik} c_{kj}$$

- 3) izračunati dijagonalne elemente c_{jj}

$$c_{jj} = a_{jj} - \sum_{k=1}^{j-1} b_{jk} c_{kj}$$

- 4) izračunati elemente $b_{ij}, i > j$ (dijagonalni elementi su 1)

$$b_{ij} = \frac{1}{c_{jj}} \left(a_{ij} - \sum_{k=1}^{i-1} b_{ik} c_{kj} \right)$$

- u slučaju $\mathbf{C} = \mathbf{B}^T$, gdje je $c_{ii} = b_{ii}$, $1 \leq i \leq n$, može se primjeniti i algoritam Cholesky faktorizacije

- prijelomna točka kod LU dekompozicije je slučaj kada je c_{jj} blizu ili jednak nuli, što rezultira ozbiljnim problemom

Npr. 2 x 2 matrica

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

U ovom slučaju algoritam za LU dekompoziciju ne radi jer $c_{11} = 0$

Matrica se može malo modificirati da se ukloni nula:

$$\mathbf{A} = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix}$$

LU DEKOMPOZICIJA MATRICE

- za modificiranu matricu dobije se

$$c_{11} = 10^{-20}$$

$$b_{21} = 10^{20}$$

$$c_{12} = 1$$

$$c_{11} = a_{11} - b_{21} = 1 - 10^{20}$$

- B i C matrice dobivene LU dekompozicijom:

$$\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 10^{20} & 1 \end{pmatrix} \qquad \mathbf{C} = \begin{pmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{pmatrix}$$

- sada je LU dekompozicija stabilna, ali nije stabilna unazad jer zbog ograničenja preciznosti $c_{22} \sim -10^{20}$

$$\begin{pmatrix} 1 & 0 \\ 10^{20} & 1 \end{pmatrix} \begin{pmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{pmatrix} = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 0 \end{pmatrix} \neq \mathbf{A}$$

LU DEKOMPOZICIJA MATRICE

- rješenje problema je u permutacijama redova i stupaca i LU dekompozicija se izvodi sa modificiranom matricom

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \Longrightarrow \mathbf{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

LU DEKOMPOZICIJA MATRICE

- LU dekompozicija može se izvesti koristeći postojeću funkciju iz "Numerical Recipes in C"
- tekst knjige na webu:

  www.nrbook.com

Potrebne funkcije nalaze se u lib.cpp i lib.h (web).

- LU dekompozicija:

  ludcmp(double $**$a, int n, int $*$indx, double $*$d)

Matrice dobivene LU dekompozicijom su zapisane kompaktno (matrica C na dijagonali i iznad dijagonale, matrica B ispod dijagonale) u polje umjesto originalne matrice a.

RJEŠAVANJE SUSTAVA LINEARNIH JEDNADŽBI

- koristeći LU dekompoziciju, sustav linearnih jednadžbi se može jednostavno riješiti

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= w_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= w_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= w_3 \\
a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= w_4
\end{aligned}$$

- zapis u matričnom obliku:

$$\mathbf{A}\mathbf{x} = \mathbf{w}$$

- Primjenom LU dekompozicije,

$$\mathbf{A}\mathbf{x} \equiv \mathbf{B}\mathbf{C}\mathbf{x} = \mathbf{w}$$

- jednadžba se može rješavati u dva koraka,

$$\mathbf{B}\mathbf{y} = \mathbf{w} \quad \Longrightarrow \quad \mathbf{y}$$

$$\mathbf{C}\mathbf{x} = \mathbf{y} \quad \Longrightarrow \quad \mathbf{x}$$

RJEŠAVANJE SUSTAVA LINEARNIH JEDNADŽBI

- u slučaju 4x4 matrice,

$$\begin{aligned}
y_1 &= w_1\\
b_{21}y_1 + y_2 &= w_2\\
b_{31}y_1 + b_{32}y_2 + y_3 &= w_3\\
b_{41}y_1 + b_{42}y_2 + b_{43}y_3 + y_4 &= w_4
\end{aligned} \quad \Longrightarrow \quad \mathbf{y}$$

$$\begin{aligned}
c_{11}x_1 + c_{12}x_2 + c_{13}x_3 + c_{14}x_4 &= y_1\\
c_{22}x_2 + c_{23}x_3 + c_{24}x_4 &= y_2\\
c_{33}x_3 + c_{34}x_4 &= y_3\\
c_{44}x_4 &= y_4
\end{aligned} \quad \Longrightarrow \quad \mathbf{x}$$

RJEŠAVANJE SUSTAVA LINEARNIH JEDNADŽBI

- sumarni pregled algoritma za rješavanje sustava jednadžbi:

1) Definirati matricu A i vektor w sa odgovarajućim dimenzijama

2) LU dekompozicija matrice → može se koristiti funkcija

$$\text{ludcmp(double } **\text{a, int n, int } *\text{indx, double } *\text{d)}$$

3) Primjena LU dekomponirane matrice u rješavanju sustava jednadžbi sa ciljem određivanja nepoznanica x
→ može se koristiti funkcija

$$\text{lubksb(double } **\text{a, int n, int } *\text{indx, double } *\text{w)}$$

Funkcija koristi LU dekomponiranu matricu a i vektor w, i vraća rješenje x u polju gdje je na ulazu bio pohranjen vektor w

ZADATAK 7

- Napišite program koji izvodi LU dekompoziciju proizvoljne matrice *A* (dozvoljeno koristenje funkcija iz Num. Recipes).
- 1) Ispisati matricu *A* i matrice dobivene LU dekompozicijom i provjeriti na jednostavnom primjeru (npr. 3x3 matrica) da je dekompozicija dobro napravljena.
- 2) Koristeći LU dekompoziciju, izračunati determinantu matrice *A*, provjeriti rezultat na primjeru (npr. 3x3 matrica)
- 3) Primjeniti LU dekompoziciju u rješavanju sustava n linearnih jednadžbi sa n nepoznanica (dozvoljeno je korištenje funkcije lubksb iz Numerical Recipes). Treba provjeriti da dobiveno rješenje doista odgovara polaznim jednadžbama.