Riešenie systému nelineárnych rovníc

$$\vec{F}(\vec{x}) = \begin{pmatrix} f_1(\vec{x}) \\ f_2(\vec{x}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Newtonova metóda

$$0 = \vec{F}(\vec{x}*) = \vec{F}(\vec{x}_n) + \vec{F}\,'(\vec{x}_n)(\vec{x}* - \vec{x}_n) + \ldots$$

(Taylorov rozvoj pre funkciu viac premenných)

$\vec{F}\,'(\vec{x})$ je jakobián transformácie:

$$J = \vec{F}\,'(\vec{x}) = \begin{pmatrix} \partial_x f_1 & \partial_y f_1 \\ \partial_x f_2 & \partial_y f_2 \end{pmatrix}$$

v každom kroku sa vypočíta veľkosť opravy (lineárny systém)

a pripočíta sa k aktuálnemu vektoru riešenia

$$\vec{F}\,'(\vec{x}_n)d\vec{x}_n = -\vec{F}(\vec{x}_n)$$

$$\vec{x}_{n+1} = \vec{x}_n + d\vec{x}_n$$

Riešenie systému nelineárnych rovníc

Newtonova metóda

zadefinujeme si $f_1(\vec{x}),\ f_2(\vec{x}),\ J$

počiatočný odhad – vektor (1, 3)

V cykle riešime

$$J d\vec{x}_n = -\vec{F}(\vec{x}_n)$$

a opravujeme aktuálny vektor riešenia

$$\vec{x}_{n+1} = \vec{x}_n + d\vec{x}_n$$

po troch iteráciách máme výsledok na 6 platných cifier

program vo výpočtovom prostredí Mathematica

```
f1[x_, y_] := Sin[x] Sin[y] + y / 2 - x

f2[x_, y_] := y^2 - 4 x^2 + Cos[x] Cos[y]

J[x_, y_] = ( ∂x f1[x, y]   ∂y f1[x, y]
              ∂x f2[x, y]   ∂y f2[x, y] );

J[x, y] // MatrixForm

( -1 + Cos[x] Sin[y]        1/2 + Cos[y] Sin[x]
  -8 x - Cos[y] Sin[x]      2 y - Cos[x] Sin[y] )

Module[{x, y},
 x = 1.; y = 3.;
 Do[
  r = Solve[J[x, y].{dx, dy} == -{f1[x, y], f2[x, y]}, {dx, dy}];
  x = x + dx /. r[[1]];
  y = y + dy /. r[[1]];
  Print["x = ", x, ",  y = ", y],
  {i, 1, 4}];
]

x = 1.6556,  y = 3.03943

x = 1.57212,  y = 3.14108

x = 1.5708,  y = 3.14159

x = 1.5708,  y = 3.14159

N[{Pi / 2, Pi}, 8]

{1.5707963, 3.1415927}
```

lineárne systémy sa vyskytujú veľmi často,
je vhodné mať pre ne špeciálne prispôsobené efektívne metódy riešenia

priame metódy: eliminovanie, vyjadrenie
(použiteľné pre relatívne malé systémy)

iteratívne metódy: dostatočne presná aproximácia
(presnosť určuje potrebný počet iterácií)

$$Ax = b$$

A je štvorcová nesingulárna matica

Gaussova eliminácia

najdôležitejší predstaviteľ priamych metód je Gaussova eliminácia

postup:

1. ekvivalentnými úpravami získať systém
s hornou **trojuholníkovou** maticou (priamy chod)
2. postupne vypočítať neznáme (spätný chod)

veľmi užitočné sa ukazuje **vyberať hlavné prvky**
(prvky na diagonále) v každom kroku tak, aby boli čo najväčšie
– dosiahnuť to možno výmenou riadkov pred každým eliminovaním
(čiastočný výber hlavného prvku),
– zaručuje to, že riešenie neznehodnotia zaokrúhľovacie chyby

bez výberu hlavných prvkov je Gaussova eliminácia použiteľná
len pre špeciálne matice:
rýdzo diagonálne dominantná, pozitívne definitná

Gaussova eliminácia – LU dekompozícia

v prípadoch, kedy je potrebné riešiť sústavu opakovane pre rôzne pravé strany: LU dekompozícia

priamy chod: získanie hornej a dolnej trojuholníkovej matice a permutačnej matice

$$Ax = b$$

$$PAx = Pb$$

$$PAx = L\underbrace{Ux}_{y} = \underbrace{Pb}_{z} \qquad Ly = z$$

spätný chod: najskôr vypočítame y, potom hodnoty neznámej x riešením sústav

$$Ly = z \qquad Ux = y$$

výpočet riešenia pre viaceré pravé strany je možné urobiť analogicky ako v prípade jednej – vektory budú teraz matice

LU dekompozícia

zadefinujeme si maticu

permutačný vektor

hľadanie pivota a zámena riadkov

vlastné eliminovanie

ukladanie prvkov dolnej trojuholníkovej matice

```
Module[{A, b, n, m, max, im, tmp},
```

$$A = \begin{pmatrix} -0.4 & -0.95 & -0.4 & -7.34 \\ 0.5 & -0.3 & 2.15 & -2.45 \\ -2 & 4 & 1 & -3 \\ -1 & 5.5 & 2.5 & 3.5 \end{pmatrix};$$

```
 p = {1, 2, 3, 4};
 n = 4;
 Do[
  max = Abs[A[[k, k]]]; im = k;
  Do[If[Abs[A[[j, k]]] > max, max = Abs[A[[j, k]]]; im = j, False], {j, k + 1, n}];
  Do[tmp = A[[k, j]]; A[[k, j]] = A[[im, j]]; A[[im, j]] = tmp, {j, 1, n}];
  tmp = p[[k]]; p[[k]] = p[[im]]; p[[im]] = tmp;
  Do[
   m = A[[i, k]] / A[[k, k]];
   Do[A[[i, j]] = A[[i, j]] - m A[[k, j]], {j, k + 1, n}];
   A[[i, k]] = m,
   {i, k + 1, n}],
  {k, 1, n - 1}];
 Print[A // MatrixForm]; Print[p]]
```

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ -0.25 & 0.2 & 1 & 0 \\ 0.2 & -0.5 & 0.2 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} -2 & 4 & 1 & -3 \\ \frac{1}{2} & 3.5 & 2. & 5. \\ -0.25 & 0.2 & 2. & -4.2 \\ 0.2 & -0.5 & 0.2 & -3.4 \end{pmatrix} \leftarrow U = \begin{pmatrix} -2 & 4 & 1 & -3 \\ 0 & 3.5 & 2 & 5 \\ 0 & 0 & 2 & -4.2 \\ 0 & 0 & 0 & -3.4 \end{pmatrix}$$

{3, 4, 2, 1}

Potom pomocou p vypočítame z (poprehadzujeme prvky b)

a z vyriešených sústav $Ly = z$ $\quad Ux = y$ získame vyjadrením x

aplikácie:

výpočet determinantu (súčin diagonálnych prvkov, znamienko)

výpočet inverznej matice (b bude jednotková matica)

špeciálne algoritmy pre riedke matice (pásová matica – tridiagonálna)

vo všeobecnosti dostaneme použitím Gaussovej eliminácie

malé rezíduum: $r = Ax - b$

nemusí to však znamenať, že riešenie je blízke presnému riešeniu

vzťah: veľkosť rezídua – veľkosť chyby riešenia možno charakterizovať

pomocou čísla podmienenosti matice: $\kappa(A)$

Podmienenosť sústavy

podmienenosť sústavy je ekvivalentná podmienenosti matice sústavy

číslo podmienenosti:

$$\kappa(A) = \frac{M}{m}$$

maximálne a minimálne relatívne predĺženie pri zobrazení vektora maticou A

$$M = \max \frac{\|Ax\|}{\|x\|}$$

$$m = \min \frac{\|Ax\|}{\|x\|}$$

pokiaľ poznáme číslo podmienenosti,

umožňuje nám to odhadnúť

relatívnu chybu riešenia

Podmienenosť sústavy

$$A(x+\Delta x)=b+\Delta b \quad \Rightarrow \quad A\Delta x=\Delta b$$

$$M \geq \frac{\|b\|}{\|x\|} \qquad m \leq \frac{\|\Delta b\|}{\|\Delta x\|}$$

$$\kappa(A)=\frac{M}{m} \geq \frac{\|b\|}{\|x\|}\left(\frac{\|\Delta b\|}{\|\Delta x\|}\right)^{-1}$$

$$\frac{\|\Delta b\|}{\|b\|}\kappa(A)=\frac{\|r\|}{\|b\|}\kappa(A) \geq \frac{\|\Delta x\|}{\|x\|}$$

takže malá hodnota čísla podmienenosti
spôsobuje, že malé reziduum zabezpečí malú
relatívnu chybu riešenia

sústavu považujeme za dobre podmienenú, ak $\kappa(A)<100$

podmienka porušená – sústava zle podmienená