6.1. Úvod do matematických algoritmov

Matematické postupy tvoria úplný základ pri algoritmickom myslení. Jeden zo základných vedeckých výpočtov je riešenie sústavy lineárnych rovníc. Využíva sa nielen v matematike, ale aj vo fyzike, a iných pridružených odvetviach.

6.1.1. Gaussova eliminácia (Gaussian elimination)

Základ tohto algoritmu na riešenie sústavy rovníc je jednoduchý a menil sa, od svojho vzniku pred 150 rokmi, len nepatrne. Algoritmus je veľmi dobre pochopený, hlavne v posledných 20 rokoch, takže môžeme presnosti jeho výsledkov spoľahlivo dôverovať. Vo všeobecnosti, chceme vyriešiť nasledovnú sústavu N rovníc o N neznámych:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1N}x_N &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2N}x_N &= b_2 \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ a_{N1}x_1 + a_{N2}x_2 + \ldots + a_{NN}x_N &= b_N \end{aligned}$$

Čo sa zapisuje v podobe matíc nasledovne:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}$$

Alebo jednoduchšie symbolickým zápisom $A \cdot x = b$, kde A reprezentuje koeficienty neznámych v rovniciach, x reprezentuje premenné a b reprezentuje pravé strany rovníc. Keďže sa pri ďalšom výpočte riadky A menia súčasne so zodpovedajúcimi prvkami v b, je pohodlné si pamätať b ako (N+**1**)-vý stĺpec matice A. Zhrňme si teraz operácie, ktoré môžeme vykonávať na takýchto rovniciach bez zmeny výslednej množiny riešení:

- *Výmena rovníc*. Zrejme poradie, v ktorom sú rovnice napísané neovplyvňuje riešenia danej sústavy. V maticovej reprezentácií to zodpovedá vzájomnej výmene dvoch riadkov matice.
- *Premenovanie neznámych*. Toto zodpovedá vzájomnej výmene dvoch stĺpcov v maticovej reprezentácií. Ak sú stĺpce i a j vymenené, tak aj neznáme x_i a x_j musíme vymeniť.
- *Vynásobenie rovníc nenulovou konštantou*. Znovu, v maticovej reprezentácií, to zodpovedá vynásobením hodnôt v riadku rovnakou nenulovou konštantou.
- *Sčítanie dvoch rovníc a zámena jednej z nich týmto súčtom*. Zrejme, takéto kombinovanie rovnic nám nič nepokazí, pretože ľahko vidieť, že ak takto zmenená sústava má riešenie, tak aj pôvodná má riešenie a naopak.

Gaussova eliminácia prebieha v dvoch fázach:

1. *Predná eliminácia* (forward-elimination), kde pôvodný systém rovníc transformujeme systematickým odstraňovaním neznámych z rovníc tak, aby (v matici A) koeficienty pod hlavnou diagonálou zostali nulové.
2. *Spätná substitúcia* (backward-substitution), kde sú hodnoty neznámych dopočítané z redukovanej trojuholníkovej matice vytvorenej v prvej fáze.

6.1.2. Priebeh Gaussovej eliminácie

Prvá fáza – *predná eliminácia* – prebieha nasledovne: najprv, eliminujeme prvú neznámu vo všetkých okrem prvej rovnice – pripočítame vhodný násobok prvej rovnice k ostatným rovniciam. Potom eliminujeme druhú neznámu vo všetkých okrem prvých dvoch rovniciach tak, že pripočítame vhodný násobok druhej rovnice ku každej rovnici od tretej až po poslednú, atď.

Všeobecne, na eliminovanie ***i***-tej neznámej v ***j***-tej rovnici (pre ***j*** idúce od ***i*+1** do ***N***), od ***j***-tej rovnice odpočítame $\frac{a_{ji}}{a_{ij}}$ násobok ***i***-tej rovnice. Problém nastáva, keď je $a_{ij} = 0$, vtedy môže nastať delenie nulou. Toto jednoducho opravíme tak, že ***i***-ty riadok vymeníme s niektorým ďalším riadkom (od ***i*+1** do ***N***) tak, aby bol prvok a_{ij} nenulový. Ak neexistuje takýto ďalší riadok, naša matica je singulárna, tzv. neexistuje jednoznačné riešenie. Prvok a_{ij}, ktorý eventuálne použijeme na elimináciu nenulových prvokov v ***i***-tom stĺpci pod diagonálou nazývame *pivot*. Ukazuje sa, že pri výbere *pivota* je vhodné nájsť nielen nejaký nenulový prvok, ale zároveň prvok najväčší v absolútnej hodnote. Hodnota pivota sa totiž používa pri následných elimináciách a vzniknuté koeficienty pri násobení by mohli byť priveľmi veľké, čo by spôsobilo, že ďalšie hodnoty v matici by boli nepresné v dôsledku zaokrúhľovacích chýb (round-off-error).

V druhej fáze – *spätná substitúcia* – jednoducho, začínajúc od neznámej x_N smerom naspäť (až ku x_1), postupne počítame hodnoty neznámych z už vypočítaných hodnôt neznámych. Na zistenie hodnoty neznámej x_i nám postačujú hodnoty neznámych $x_{i+1}, ..., x_N$.

```
var N:integer;
    a:array[1..100,1..100] of real;
    x:array[1..100] of real;

procedure eliminuj;
var i,j,k,max:integer;
    t:real;
begin
  for i:=1 to N do
   begin
     max:=i;
     for j:=i+1 to N do
       if abs(a[j,i]) > abs(a[max,i]) then
         max:=j;
     for k:=i to N+1 do
      begin
       t:=a[i,k]; a[i,k]:=a[max,k]; a[max,k]:=t;
      end;
     for j:=i+1 to N do
       for k:=N+1 downto i do
         a[j,k]:=a[j,k]-a[i,k]*a[j,i]/a[i,i];
   end;
end;

procedure substituuj;
var j,k:integer;
    t:real;
begin
  for j:=N downto 1 do
   begin
     t:=0.0;
     for k:=j+1 to N do
       t:=t+a[j,k]*x[k];
     x[j]:=(a[j,N+1]-t)/a[j,j];
   end;
end;
```

```
double a[101][101], x[101];
int N;

void eliminuj(void)
{
  int i,j,k,max;
  double t;

  for (i = 1; i <= N; i++)
  {
    max = i;
    for (j = i+1; j <= N; j++)
      if (fabs(a[j][i]) > fabs(a[max][i]))
        max = j;
    /* výmena riadku */
    for (k = i; k <= N+1; k++)
    { t = a[i][k]; a[i][k]=a[max][k]; a[max][k]=t; }
    /* eliminácia */
    for (j = i+1; j <= N; j++)
      for (k = N+1; k >= i; k--)
        a[j][k] -= a[i][k]*a[j][i]/a[i][i];
  }
}

void substituuj(void)
{
  int j,k;
  double t;
  for (j = N; j >= 1; j--)
  {
    t = 0.0;
    for (k = j+1; k <= N; k++)
      t += a[j][k]*x[k];
    x[j] = (a[j][N+1]-t)/a[j][j];
  }
}
```

Gaussova eliminácia pracuje v čase $O(N^3)$, a pamäti $O(N^2)$ – koeficienty neznámych v rovniciach.