



NUMERICKÉ ALGORITMY

TOMÁŠ DVOŘÁK

Numerická matematika se zabývá konstruktivními postupy, pomocí nichž lze získat řešení matematických problémů s libovolnou přesností po provedení konečného počtu aritmetických operací.

Numerické výpočty jsou zpravidla zatíženy chybami: Jde o *chyby ve vstupních údajích* (dány např. nepřesností naměřených dat), *zaokrouhlovací chyby* (dány konečnou reprezentací čísel v paměti počítače) a *chybu metody* (numerická metoda neposkytne přesné řešení, ale pouze jeho aproximaci).

Metody lze rozdělit do dvou skupin. *Přímá metoda* dává algoritmus teoreticky přesného řešení, tj. výsledek je přesný za předpokladu, že všechny výpočty jsou provedeny přesně (příklad: vzorec pro výpočet kořenů kvadratické rovnice). *Iterační metoda* konstruuje posloupnost $x_0, x_1, \dots, x_k, \dots$ částečných aproximací přesného řešení. Zde je vždy důležité zjistit, za jakých podmínek posloupnost $\{x_k\}_{k=0}^{\infty}$ konverguje k přesnému řešení dané úlohy (hovoříme o *konvergenci metody*).

1. ŘEŠENÍ NELINEÁRNÍCH ROVNIC.

Problém: Nalézt řešení rovnice $f(x) = 0$ na intervalu $\langle a, b \rangle$, kde f je reálná funkce, definovaná na $\langle a, b \rangle$. Řešením rovnice je číslo $x^* \in \langle a, b \rangle$ splňující $f(x^*) = 0$.

Obecné schema iterační metody:

```
zvol počáteční aproximaci  $x_0 \in \langle a, b \rangle$ ;  
k:=0;  
repeat k:=k+1;  
    pomocí iteračního vzorce spočti  $x_k$ ;  
until  $|x^* - x_k| < \epsilon$ .  
return  $x_k$ .
```

Výstupem je tedy přibližné řešení, určené s předem zvolenou přesností ϵ .

Myšlenkou *Newtonovy metody* je nahrazení funkce f tečnou (ke grafu funkce f). Je-li x_k již známá aproximace kořene x^* , pak další aproximaci x_{k+1} obdržíme jako průsečík osy x a tečny ke grafu funkce f v bodě $[x_k, f(x_k)]$. Protože rovnice tečny má tvar $y = f'(x_k)(x - x_k)$, položíme-li $y = 0$, obdržíme iterační vzorec

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Postačující podmínky pro konvergenci Newtonovy metody jsou náročné: Je třeba, aby

- (1) interval $\langle a, b \rangle$ obsahoval jediné řešení x^* rovnice $f(x)$,
- (2) první i druhá derivace funkce f byly na $\langle a, b \rangle$ nenulové a zachovávaly tam znaménko,
- (3) počáteční iterace $x_0 \in \langle a, b \rangle$ splňovala podmínku $f(x_0)f''(x_0) > 0$.

Pro odhad chyby lze použít vztahu

$$|x_k - x^*| \leq \frac{M}{2m}(x_k - x_{k-1})^2, \quad \text{kde } M = \max_{x \in \langle a, b \rangle} |f''(x)|, \quad m = \min_{x \in \langle a, b \rangle} |f'(x)|.$$

Protože uvedené podmínky se obtížně ověřují, v praxi se ukončení výpočtu testuje některým z těchto empirických kritérií:

- (1) $|x_k - x_{k-1}| < \epsilon$,
- (2) $|f(x_k)| < \epsilon$.

Z podmínky (1) ani (2) obecně neplyne, že chyba aproximace je menší než ϵ (tj. že $|x_k - x^*| < \epsilon$), proto říkáme, že jsme řešení našli s *tolerancí* ϵ .



2. SOUSTAVY LINEÁRNÍCH ROVNIC

Problém: Nalézt řešení soustavy

$$Ax = b$$

n lineárních rovnic o n neznámých. $A = (a_{ij})_{i,j=1}^n$ je reálná čtvercová matice řádu n , $x = (x_1, \dots, x_n)$ vektor neznámých a $b = (b_1, \dots, b_n)$ reálný vektor pravých stran. Budeme předpokládat, že matice A je regulární (tj. $\det A \neq 0$). Z tohoto předpokladu vyplývá, že soustava má právě jedno řešení.

Metoda LU -rozkladu je založena na myšlence vyjádřit A jako součin matic $A = LU$, kde L je dolní a U horní trojúhelníková matice (tj. $L = (l_{ij})_{i,j=1}^n$, $l_{ij} = 0$ pro $i < j$, a $U = (u_{ij})_{i,j=1}^n$, $u_{ij} = 0$ pro $i > j$). Pak lze soustavu zapsat ve tvaru

$$LUx = b$$

a získat její řešení tak, že

- (1) vyřešíme soustavu $Ly = b$,
- (2) vyřešíme soustavu $Ux = y$.

Vyřešit obě soustavy je snadné, v prvním případě postupným dosazováním od první rovnice počínaje (přímá substitute), ve druhém pak postupným dosazováním od poslední rovnice (zpětná substitute).

Jak nalézt LU -rozklad matice A ? Nejprve zapíšeme matici ve tvaru

$$A = \begin{bmatrix} a_{11} & w^T \\ v & A' \end{bmatrix},$$

kde w^T je řádkový a v sloupcový vektor délky $n-1$ a A' podmatice řádu $n-1$. Budeme se snažit navrhnout rekurzivní proceduru LU -rozkladu matice A . S použitím lineární algebry odvodíme

$$A = \begin{bmatrix} 1 & 0 \\ \frac{v}{a_{11}} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & A' - \frac{vw^T}{a_{11}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{v}{a_{11}} & I_{n-1} \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & L'U' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{v}{a_{11}} & L' \end{bmatrix} \begin{bmatrix} a_{11} & w^T \\ 0 & U' \end{bmatrix},$$

kde I_{n-1} je jednotková matice řádu $n-1$ (na hlavní diagonále má jedničky a všude jinde samé nuly), a L' a U' jsou dolní a horní trojúhelníková matice, pro něž $L'U' = A' - \frac{vw^T}{a_{11}}$. Jak tyto matice nalezneme? Rekurzivní aplikací odvozeného vztahu, kde roli matice A bude hrát matice $A' - \frac{vw^T}{a_{11}}$. Tento postup dává následující algoritmus, v němž je již ovšem rekurze nahrazena iterací.

LU -rozklad(A):

```
for k:=1 to n do  $u_{kk}:=a_{kk}$ ;  
    for i:=k+1 to n do  $l_{ik}:=a_{ik}/u_{kk}$ ;  
         $u_{ki}:=a_{ki}$   
    od  
    for i:=k+1 to n do for j:=k+1 to n  
        do  $a_{ij}:=a_{ij} - l_{ik}u_{kj}$  od  
    od  
return  $L = (l_{ij})_{i,j=1}^n$ ,  $U = (u_{ij})_{i,j=1}^n$  (nulové trojúhelníky obou matic nejsou inicializovány).
```

Algoritmus selže v případě, že pro některé k bude $u_{kk} = 0$ (dělení nulou). Tuto nesnáz lze však snadno odstranit: Najdeme $m > k$ tak, že $u_{km} \neq 0$ (takový prvek existuje, protože matice A je regulární). Pak vyměníme m -tý a k -tý řádek a pokračujeme v provádění algoritmu.

3. SOUSTAVY NELINEÁRNÍCH ROVNIC

Problém: Nalézt řešení soustavy

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, \dots, x_n) &= 0 \end{aligned}$$



na n -rozměrném intervalu $R_n = \langle a_1, b_1 \rangle \times \dots \times \langle a_n, b_n \rangle$.

S použitím vektorové symboliky lze soustavu ekvivalentně zapsat ve tvaru $F(\mathbf{x}) = \mathbf{0}$, kde $F = (f_1, \dots, f_n)^T$ je zobrazení z R^n do R^n , $\mathbf{x} = (x_1, \dots, x_n)$ vektor neznámých a $\mathbf{0}$ n -rozměrný nulový vektor.

Newtonova metoda řešení systému nelineárních rovnic je (podobně jako v případě jedné rovnice) založena na iteračním vzorci

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [F'(\mathbf{x}^{(k)})]^{-1} F(\mathbf{x}^{(k)}),$$

kde $F'(\mathbf{x})$ je derivace zobrazení F , tj. matice (tzv. Jacobiova matice zobrazení F), jejímiž prvky jsou parciální derivace funkcí f_i podle proměnných x_j :

$$F'(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Exponent -1 pak značí inverzní matici. Protože výpočet podle uvedeného vzorce by byl obtížný, zavedeme místo toho vektor přírůstků $\Delta^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ ($\Delta^{(k)} = (\Delta_1^{(k)}, \dots, \Delta_n^{(k)})$), a dosazením do iteračního vzorce obdržíme soustavu

$$(*) \quad F'(\mathbf{x}_k) \Delta^{(k)} = -F(\mathbf{x}_k),$$

n lineárních rovnic o n neznámých $\Delta_1^{(k)}, \dots, \Delta_n^{(k)}$ (kterou umíme vyřešit!).

Algoritmus:

Zvol počáteční aproximaci $\mathbf{x}_{(0)} \in R_n$ "blízko" předpokládanému řešení;

$k:=0$;

repeat řešením soustavy (*) urči vektor $\Delta^{(k)}$;

$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} + \Delta^{(k)};$$

$k:=k+1$;

until $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| < \epsilon$ or $\det[F'(\mathbf{x}^{(k)})]$ je blízký nule.

Symbol $\|\mathbf{x}\|$ značí normu vektoru \mathbf{x} , např. $\|(x_1, \dots, x_n)\| = \max\{x_1, \dots, x_n\}$. Je-li determinant matice $F'(\mathbf{x}^{(k)})$ blízký nule, je vhodné výpočet ukončit, protože výsledek by byl vlivem zaokrouhlovacích chyb zcela znehodnocen.

4. APROXIMACE FUNKCÍ

Polynomiální interpolace. Je dán soubor dat $(x_0, y_0), \dots, (x_n, y_n)$, kde $x_i \neq x_j$ pro $i \neq j$ (lze např. předpokládat, že y_i jsou naměřené hodnoty neznámé funkce f v uzlových bodech x_i). Úlohou polynomiální interpolace je nalézt polynom $p(x)$ stupně nejvýše n , splňující

$$p(x_0) = y_0, p(x_1) = y_1, \dots, p(x_n) = y_n.$$

Klasické řešení podává *Lagrangeův interpolační vzorec*:

$$p(x) = \sum_{0 \leq j \leq n} y_j \prod_{\substack{0 \leq i \leq n \\ i \neq j}} \frac{x - x_i}{x_j - x_i}.$$

Aproximace metodou nejmenších čtverců. Je dán soubor dat $(x_0, y_0), \dots, (x_n, y_n)$. Hledáme funkci $F(x)$ tak, aby součet

$$R = \sum_{i=0}^n (F(x_i) - y_i)^2$$

byl minimalizován (minimalizujeme *součet čtverců odchylek*, odtud název metody). Funkci F budeme hledat ve tvaru lineární kombinace $F(x) = \sum_{j=0}^m c_j f_j(x)$ vhodně zvolených "základních funkcí" f_0, \dots, f_m .



Nutnou podmínkou pro existenci minima funkce R je

$$\frac{\partial R}{\partial c_0} = 0, \frac{\partial R}{\partial c_1} = 0, \dots, \frac{\partial R}{\partial c_m} = 0.$$

Po dosazení za R a výpočtu parciálních derivací $\frac{\partial R}{\partial c_j}$ získáme soustavu

$$\sum_{j=0}^m c_j \sum_{i=0}^n f_j(x_i) f_k(x_i) = \sum_{i=0}^n y_i f_k(x_i) \quad k = 0, \dots, m,$$

$m + 1$ lineárních rovnic o $m + 1$ neznámých c_0, \dots, c_m , zvanou *soustava normálních rovnic*. Položíme-li nyní

$$A = \begin{bmatrix} f_0(x_0) & f_1(x_0) & \dots & f_m(x_0) \\ f_0(x_1) & f_1(x_1) & \dots & f_m(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ f_0(x_n) & f_1(x_n) & \dots & f_m(x_n) \end{bmatrix},$$

soustava normálních rovnic má tvar

$$A^T A c = A^T y,$$

kde $y = (y_0, \dots, y_n)$ a $c = (c_0, \dots, c_m)$ je vektor neznámých.

Algoritmus

sestav matici A ;

vyřeš systém lineárních rovnic $A^T A c = A^T y$;

return $F(x) = c_0 f_0 + c_1 f_1 + \dots + c_m f_m$.

Zbývá zodpovědět otázku, jak zvolit základní funkce f_0, \dots, f_m ? Populární volbou je $f_j(x) = x^j$, což znamená, že $F(x) = c_0 + c_1 x + \dots + c_m x^m$ je polynom stupně m .

5. NUMERICKÁ INTEGRACE

Problém: Vyčíslit integrál $\int_a^b f(x) dx$, kde f je spojitá reálná funkce na intervalu $\langle a, b \rangle$.

Metody pro přibližný výpočet integrálu pracují tak, že hodnoty funkce f počítáme jen v konečně mnoha uzlových bodech $a = x_0 < x_1 < \dots < x_n = b$. $\int_a^b f(x) dx$ pak nahradíme aproximací tvaru

$$\sum_{i=0}^n w_i f(x_i)$$

(tzv. *kvadrurní vzorec*). Nejběžnější způsob odvození kvadrurních vzorců spočívá v nahrazení integrované funkce interpolačním polynomem a integrací tohoto polynomu. Podle toho, zda volíme polynom stupně 0, 1, či 2, obdržíme *obdélníkové pravidlo* O_n , *lichoběžníkové pravidlo* L_n a *Simpsonovo pravidlo* S_n (zde je třeba, aby n bylo sudé).

Za předpokladu, že $x_i - x_{i-1} = h$ pro každé $i \in \{1, \dots, n\}$, mají kvadrurní vzorce tento tvar:

$$O_n = h \left(f(x_0 + \frac{h}{2}) + f(x_1 + \frac{h}{2}) + \dots + f(x_{n-1} + \frac{h}{2}) \right)$$

$$L_n = \frac{h}{2} (f(x_0) + 2f(x_1) + \dots + 2f(x_{n-1}) + f(x_n))$$

$$S_n = \frac{h}{3} (f(x_0) + 4f(x_1) + 2f(x_2) + \dots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)), \quad n \text{ sudé.}$$

LITERATURA

1. R. Černá, M. Machalický, J. Vogel, Č. Zlatník, *Základy numerické matematiky a programování*, SNTL, Praha, 1987.
2. D. E. Knuth, *The Art of Computer Programming, vol. 2: Seminumerical Algorithms*, Addison-Wesley, Reading, MA, 1969, 2. vydání, 1981.
3. W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, 1988.
4. E. Vitásek, *Numerické metody*, SNTL, Praha, 1987.