Reševanje nelinearnih enačb

Uvod

- Ogledali si bomo metode za reševanje enačb oblike

$$f(x) = 0, \quad f : I \subseteq \mathbb{R} \to \mathbb{R}$$

ter posplošitev na reševanje sistemov oblike

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}, \quad \boldsymbol{F} : D \subseteq \mathbb{R}^d \to \mathbb{R}^d.$$

- Številu x, za katerega je $f(x) = 0$, rečemo ničla funkcije f ali rešitev enačbe $f(x) = 0$.

Primer

Denimo, da je 150m dolga tračnica trdno vpeta na obeh koncih. Zaradi segrevanja se raztegne za 1 *cm in usloči v obliki krožnega loka. Za koliko se sredina tračnice dvigne od tal?*

Na splošno o obstoju in številu (realnih) rešitev enačbe $f(x) = 0$ ne moremo reči ničear:

- enačba $\exp(-x) - x = 0$ ima natanko eno rešitev,
- enačba $x - \tan x = 0$ ima neskončno rešitev,
- enačba $x^3 - 6x^2 + 11x - 6 = 0$ ima natanko tri rešitve,
- enačba $x^4 + 1 = 0$ nima nobene rešitev.

Izrek

Naj bo $f : [a, b] \to \mathbb{R}$ zvezna funkcija za katero je $f(a) f(b) < 0$. Potem ima f na (a, b) vsaj eno ničlo.

Izrek je osnova za prvo metodo za iskanje ničle funkcije, t.i. bisekcijo.

Bisekcija

```
function nicla=bisekcija(f,a,b,eps)
%BISEKCIJA vrne niclo funkcije po metodi bisekcije
%nicla=BISEKCIJA(f,a,b,eps)
%f je funkcija, katere niclo iscemo
%a in b sta krajisci intervala, eps je toleranca

while abs(b-a)>eps
  c= a+(b-a)/2; %bolje kot c=(a+b)/2!!!
  if sign(feval(f,a))==sign(feval(f,c))
    a=c;
  else
    b=c;
  end
end
nicla=a+(b-a)/2;
```

Nekaj komentarjev:

- Namesto očitnega izraza $c = (a + b)/2$, je bolje računati $c = a + (b - a)/2$ (prekoračitev obsega ali pa c v določeni aritmetiki lahko pade iz $[a, b]$).
- Toleranca ε mora biti smiselno izbrana (ne sme biti premajhna).
- Bolje je preverjati predznake kot računati vrednosti funkcije.
- Če je ničel na (a, b) več, dobimo le eno od njih.
- Na vsakem koraku se interval, kjer je ničla, razpolovi. Po k korakih je interval, kjer je ničla, dolg

$$\frac{|b - a|}{2^k}.$$

Za natančnost ε potrebujemo torej

$$k \geq \log_2 \frac{|b - a|}{\varepsilon}$$

korakov.

Navadna iteracija

Pri metodi navadne iteracije osnovno enačbo

$$f(x) = 0$$

preoblikujemo v ekvivalentno

$$x = g(x)$$

in izvajamo iteracijo

$$x_{r+1} = g(x_r)$$

pri izbranem začetnem približku x_0.
Nekatere možne izbire za iteracijsko funkcijo g so denimo

- $g(x) = x - f(x)$,
- $g(x) = x - C\,f(x), C \neq 0$,
- $g(x) = x - h(x)\,f(x), h(x) \neq 0$.

Primer

Poiščimo kakšno iteracijsko funkcijo g za reševanje enačbe

$$p(x) = x^3 - 5\,x + 1 = 0.$$

Možnosti so

a)
$$g_1(x) = \frac{x^3+1}{5}$$

b)
$$g_2(x) = \sqrt[3]{5x-1}$$

c)
$$g_3(x) = \frac{1}{5-x^2}$$

Konvergenco iteracije $x_{r+1} = g_i(x_r)$ pri različnih začetnih približkih x_0 lahko raziščemo s pomočjo programa `iter.m`.

```
function nicla=iter(g,x0,eps,N)
%ITER vrne resitev nelinearne enacbe
%z navadno iteracijo
%nicla=ITER(g,x0,eps,N)
%g je iteracijska funkcija
%x0 je zacetni priblizek, eps je toleranca
%N je maksimalno stevilo korakov

k=0;
razlika=inf;
while (razlika>eps) && (k<N)
  x1=feval(g,x0)
  razlika=abs(x1-x0);
  k=k+1;
  x0=x1;
end
nicla=x1;
```

Konvergenco navadne iteracije opisuje naslednji izrek.

Izrek

Naj iteracijska funcija g na intervalu $I = [\alpha - \delta, \alpha + \delta]$ zadošča pogoju

$$|g(x) - g(y)| \leq m\,|x - y|\,, \quad x, y \in I,\ 0 \leq m < 1.$$

Potem za vsak $x_0 \in I$ zaporedje

$$x_{r+1} = g(x_r), \quad r = 0, 1, \ldots$$

konvergira k α. Velja

$$|x_r - \alpha| \leq m^r\,|x_0 - \alpha|$$

in

$$|x_{r+1} - \alpha| \leq \frac{m}{1 - m}\,|x_r - x_{r-1}|\,.$$

Posledica

Naj bo $g(\alpha) = \alpha$ in g zvezno odvedljiva pri α. Če je $|g'(\alpha)| < 1$, potem obstaja taka okolica I za α, da za vsak $x_0 \in I$ zaporedje

$$x_{r+1} = g(x_r), \quad r = 0, 1, \ldots$$

konvergira k α.

O hitrosti konvergence v bližini α odloča število $g'(\alpha)$.

Definicija

Naj zaporedje $(x_r)_{r=0}^{\infty}$ konvergira k α. Pravimo, da je red konvergence enak p, če obstajata konstanti $C_1, C_2 > 0$, da za dovolj pozne člene zaporedja velja

$$C_1 |x_r - \alpha|^p \leq |x_{r+1} - \alpha| \leq C_2 |x_r - \alpha|^p .$$

Lema

Naj bo iterativna funkcija g v okolici negibne točke $\alpha = g(\alpha)$ p-krat zvezno odvedljiva in naj bo $\left|g^{(k)}(\alpha)\right| = 0$ za $k = 1, 2, \ldots, p-1$ ter $g^{(p)}(\alpha) \neq 0$. Potem ima iterativna metoda $x_{r+1} = g(x_r)$, $r = 0, 1, \ldots,$ v bližini rešitve α red konvergence p.

Posebni primeri konvergence so:

- $p = 1$: linearna (na vsakem koraku pridobimo konstantno mnogo novih točnih decimalk),
- $p = 2$: kvadratična (na vsakem koraku se število točnih decimalk podvoji),
- $p = 3$: kubična (na vsakem koraku se število točnih decimalk potroji),
- $1 < p < 2$: superlinearna (hitrejša od linearne in počasnejša od kvadratične).

Primer

Ena od možnih iteracijskih funkcij za računanje $\sqrt{a}$, $a > 0$, *je*

$$g(x) = \frac{x^2 + a}{2\,x}.$$

Očitno je $g(\sqrt{a}) = \sqrt{a}$, $g'(\sqrt{a}) = 0$ *in* $g''(\sqrt{a}) \neq 0$. *Iterativna metoda* $x_{r+1} = g(x_r)$ *ima torej v bližini* $\sqrt{a}$ *kvadratično konvergenco.*
Izberimo $a = 10$ *in* $x_0 = 3$. *Potem je*

r	x_r
0	3.00000000
1	3.16666667
2	3.16228070
3	3.16227766

Z redečo barvo so označene točne deimalke, ki se na vsakem koraku približno podvojijo, kar potrjuje kvadratično konvergenco.

Tangentna metoda

Ideja za tangentno ali Newtonovo metodo je preprosta: nov približek je presečišče tangente v prejšnjem približku z abscisno osjo.
Če torej rešujemo $f(x) = 0$, se zaporedje glasi

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)}, \quad r = 0, 1, \ldots$$

Očitno je primer navadne iteracije

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Za konvergenco tangentne metode velja:

- Če je ničla α funkcije f enostavna, je konvergenca vsaj kvadratična. Kvadratična je, če je $f''(\alpha) \neq 0$, sicer je vsaj kubična.
- Če je α m-kratna ničla funkcije f, je konvergenca vsaj linearna.

Izrek

Naj bo α enostavna ničla dvakrat zvezno odvedljive funkcije f. Potem obstaja okolica I točke α in konstanta C, da tangentna metoda konvergira za vsak $x_0 \in I$ in približki x_r zadoščajo oceni

$$|x_{r+1} - \alpha| \leq C(x_r - \alpha)^2.$$

Izrek

Naj bo f na $I = [a, \infty)$ dvakrat zvezno odvedljiva, naraščajoča in konveksna funkcija, ki ima ničlo $\alpha \in I$. Potem je α edina ničla funkcije f na I in za vsak $x_0 \in I$ tangentna metoda konvergira k α.

Sekantna metoda

Pri tangentni metodi poleg vrednosti potrebujemo tudi odvod. Če ta ni na voljo, ali ga težko računamo, ga aproksimiramo z diferenčnim kvocientom

$$\frac{f(x_r) - f(x_{r-1})}{x_r - x_{r-1}}.$$

Tako dobimo sekantno metodo

$$x_{r+1} = x_r - \frac{f(x_r)(x_r - x_{r-1})}{f(x_r) - f(x_{r-1})}, \quad r = 0, 1, \ldots$$

Izkaže se, da je red konvergence zanjo $p \approx 1.62$ (superlinearna).

Naštejmo še nekaj metod, ki se pogosteje uporabljajo:

- Mullerjeva metoda (namesto sekante skozi zadnji dve točki vzamemo parabolo skozi zadnje tri točke),
- inverzna interpolacija (inverzno funkcijo funkcije f aproksimiramo s parabolo in za nov približek vzamemo vrednost aproksimanta v 0),
- metoda (f, f', f'') (pri izračunu novega približka uporabimo vrednost, odvod in drugi odvod),
- kombinirana Brentova metoda (kombiniramo inverzno interpolacijo, bisekcijo in sekantno metodo; uporablja jo funkcija `fzero` v Matlabu).

Polinomske enačbe

- Omejimo se sedaj na reševanje enačbe $p(x) = 0$, kjer je $p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$ polinom stopnje $\leq n$.
- Lahko uporabimo katerokoli od omenjenih metod, toda pri tem ne izkoristimo dejstva, da je p polinom.
- Ponavadi potrebujemo vse ničle polinoma p.
- Ko eno (denimo x_k) izračunamo, z deljenjem $p(x)/(x - x_k)$ dobimo polinom q nižje stopnje in nadaljujemo.
- Pri tem je v splošnem natančnost odvisna od vrstnega reda izločanja.

- Druga pot je, da sestavimo matriko

$$A_n = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & & 1 \\ -a_0/a_n & -a_1/a_n & \cdots & -a_{n-1}/a_n \end{bmatrix}$$

  in izračunamo njene lastne vrednosti.
- Lahko pa uporabimo posebne metode za reševanje polinomske enačbe: Laguerrova, Bairstrow-Hitchcockova, Jenkin-Traubova, Ehrlich-Alberthova, Durand-Kernerjeva,...

Laguerreova metoda

- Ponavljamo iteracijo:

$$S_1 = \frac{p'(z_r)}{p(z_r)}, \quad S_2 = \frac{p'(z_r)^2 - p(z_r)p''(z_r)}{p(z_r)^2}$$

$$z_{r+1} = z_r - \frac{n}{S_1 \pm \sqrt{(n-1)(n\,S_2 - S_1^2)}}$$

- Predznak izberemo tako, da ima imenovalec čim večjo absolutno vrednost.

Izrek

Če ima polinom p same realne ničle, potem za poljubni začetni približek Laguerreova metoda konvergira k levemu ali desnemu najbližjemu korenu, pri čemer si mislimo, da sta pozitivni in negativni krak realne osi povezana v neskončnosti. V primeru enostavne ničle je red konvergence v bližini ničle kubičen.

```
function z=laguerre(p,z0,eps,maxit)
%LAGUERRE poisce niclo polinoma z Laguerreovo metodo
%z=LAGUERRE(p,z0,eps,maxit)
%p je seznam koeficientov polinoma
%p=[a_n,a_{n-1},...,a_0]
%z0 je zacetni priblizek, eps je natancnost
%maxit je maksimalno stevilo iteracij

n=length(p)-1;
dp=polyder(p);
ddp=polyder(dp);
napaka=eps+1;
k=0;
```

```
while napaka>eps && k<maxit
  pz=polyval(p,z0);
  dpz=polyval(dp,z0);
  ddpz=polyval(ddp,z0);
  S1=dpz/pz;
  S2=(dpz^2-pz*ddpz)/pz^2;
  S3=sqrt((n-1)*(n*S2-S1^2));
  if abs(S1-S3)>abs(S1+S3)
    z1=z0-n/(S1-S3);
  else
    z1=z0-n/(S1+S3);
  end
  napaka=abs(z1-z0);
  z0=z1;
  k=k+1;
end
z=z1;
```

- Metoda deluje tudi za kompleksne ničle.
- Ni pa nujno konvegentna za vsak začetni približek.
- Ko eno ničlo izračunamo, jo lahko z deljenjem (Horner!) izločimo (postopku rečemo redukcija).
- Izločamo lahko korene od največjega (po absolutni vrednosti), do najmanjšega (direktna redukcija) ali od najmanjšega do največjega (obratna redukcija), ali pa kombiniramo (kombinirana redukcija).

Reševanje sistemov nelinearnih enačb

- Rešujemo sistem

$$
\begin{aligned}
f_1(x_1, x_2, \ldots, x_n) &= 0\\
f_2(x_1, x_2, \ldots, x_n) &= 0\\
&\vdots\\
f_n(x_1, x_2, \ldots, x_n) &= 0
\end{aligned}
$$

  kjer so f_i v splošnem nelinearne funkcije.

- Pisali bomo krajše $\boldsymbol{F}(\boldsymbol{x}) = \mathbf{0}$, kjer je $\boldsymbol{F} = (f_1, f_2, \ldots, f_n)^\top$, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^\top$ in $\mathbf{0} = (0, 0, \ldots, 0)^\top$.

- Najpreprostejša metoda je navadna ali Jacobijeva iteracija
- Sistem $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}$ prepišemo v $\boldsymbol{x} = \boldsymbol{G}(\boldsymbol{x})$.
- Izvajamo iteracijo

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{G}\left(\boldsymbol{x}^{(k)}\right), \quad k = 0, 1, \ldots$$

- Metoda konvergira, če sta $\boldsymbol{G}$ in $\boldsymbol{x}^{(0)}$ ustrezno izbrana.

Izrek

Če obstaja območje $\Omega \subseteq \mathbb{R}^n$ z lastnostima

a) $x \in \Omega \Rightarrow \mathbf{G}(\boldsymbol{x}) \in \Omega$,

b) $x \in \Omega \Rightarrow \rho(J_\mathbf{G}(\boldsymbol{x}) \leq q < 1$, *kjer je* $J_\mathbf{G}(\boldsymbol{x})$ *Jacobijeva matrika*

$$J_\mathbf{G}(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial g_1(\boldsymbol{x}}{\partial x_1} & \cdots & \frac{\partial g_1(\boldsymbol{x}}{\partial x_n} \\ \vdots & \vdots & \\ \frac{\partial g_n(\boldsymbol{x}}{\partial x_1} & \cdots & \frac{\partial g_n(\boldsymbol{x}}{\partial x_n} \end{bmatrix}$$

in ρ *spektralni radij matrike, potem ima* $\mathbf{G}(\boldsymbol{x}) = \boldsymbol{x}$ *v* Ω *natanko eno rešitev* $\boldsymbol{\alpha}$, *ki je limita zaporedja* $\boldsymbol{x}^{(k+1)} = \boldsymbol{G}\left(\boldsymbol{x}^{(k)}\right), \quad k = 0, 1, \ldots,$ *za poljuben začetni približek* $\boldsymbol{x}^{(0)}$.

Posledica

Za konvergenco je dovolj

a) $x \in \Omega \Rightarrow \mathbf{G}(\boldsymbol{x}) \in \Omega$,

b) $\sum_{j=1}^{n} \left| \frac{\partial g_i(\boldsymbol{x})}{\partial x_k} \right| \leq m < 1, i = 1, 2, \ldots, n.$

Obenem velja

$$\left\| \boldsymbol{x}^{(k)} - \boldsymbol{\alpha} \right| \leq \frac{m^k}{1 - m} \left\| \boldsymbol{x}^{(1)} - \boldsymbol{x}^{(0)} \right\|_\infty .$$

Newtonova metoda

- Pri Newtonovi metodi iteriramo

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - J_{\mathbf{F}}\left(\mathbf{x}^{(k)}\right)^{-1} \mathbf{F}\left(\mathbf{x}^{(k)}\right), \quad k = 0, 1, \ldots$$

- V praksi seveda ne računamo inverza $J_{\mathbf{F}}\left(\mathbf{x}^{(k)}\right)^{-1}$, ampak rešujemo sistem

$$J_{\mathbf{F}}\left(\mathbf{x}^{(k)}\right)\left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right) = -\mathbf{F}\left(\mathbf{x}^{(k)}\right).$$

- Čeprav obstajajo izreki o konvergenci Newtonove metode (denimo Kanotorovičev), je njihove predpostavke v praksi težko preveriti. Konvergenco ponavadi zagovoti že dober začetni približek.

- Kadar je sistem enačb velik, je z Newtonovo metodo veliko dela.
- Pohitrimo jo lahko tako, da Jacobijevo matriko na novo računamo samo na vsakih nekaj korakov.
- Taki metodi rečemo kvazi-Newtonova metoda.
- Znane so tudi variante, ko Jacobijevo matriko ocenimo brez poznavanja paricalnih odvodov (Broydenova metoda).