

Hledání kořenů

Úloha: Pro danou funkci $f(x)$ máme najít číslo r tak, aby $f(r) = 0$.

Pozor, počítač totiž kořen nepozná! Má jistou přesnost výpočtu $\delta > 0$ a prohlásí $f(r) = 0$ pokaždé, když $|f(x)| < \delta$. Není ovšem zaručeno, že opravdu máme kořen, to je problém např. při rozkladu polynomů. S tím se v zásadě nedá nic dělat, prostě to bereme na vědomí a většinou se s tím smíříme.

Zde se podíváme na metody iterační, tedy začneme nějakým odhadem x_0 a postupně jej vylepšujeme postupem $x_k \mapsto x_{k+1}$, dokud nejsme spokojeni. Tuto spokojenost máme zadánu jako toleranci ε . Stačí najít nějaké číslo \hat{r} splňující $|r - \hat{r}| < \varepsilon$. Problém je, že kořen r neznáme, takže ani neumíme spočítat, o kolik se od něj odhad liší.

Proto se často jako náhrada používá požadavek, aby $|f(\hat{r})| < \varepsilon$, což sice není úplně přesně to pravé, zato to umíme snadno testovat. Tyto dvě podmínky jsou (někdy) svázané.

Fakt.

Nechť f je funkce diferencovatelná na okolí U bodu r . Předpokládejme, že pro nějaké $m_1 > 0$ platí $m_1 \leq |f'|$ na U . Jestliže je r kořen funkce f , pak pro $\hat{r} \in U$ platí $|r - \hat{r}| \leq \frac{1}{m_1} |f(\hat{r})|$.

Důkaz: Věta o střední hodnotě dává jisté $\xi \in U$ tak, aby $\frac{f(\hat{r}) - f(r)}{\hat{r} - r} = f'(\xi)$ neboli (díky $f(r) = 0$)

$$f(\hat{r}) = f'(\xi)(\hat{r} - r) \implies |f(\hat{r})| = |f'(\xi)| \cdot |\hat{r} - r| \geq m_1 |\hat{r} - r|.$$

Zbytek je jasný. □

Kdy toto může selhat? Pokud je f' spojitá funkce a $f'(r) \neq 0$, pak už existuje okolí $U(r)$ a konstanta m_1 tak, jak to potřebujeme. Kritická situace tedy je $f'(r) = 0$, což je případ vícenásobného kořene.

1. Iterační metody, řád metody

Nejprve se podíváme na iteraci obecněji. Zajímá nás situace, kdy $x_{k+1} = F(x_k)$, kde F je nějaká funkce (třeba zadaná popisem algoritmu). Základní požadavek samozřejmě je, aby byla posloupnost $\{x_k\}$ konvergentní, a také chceme, aby limita dávala kořen (některý z možných). Toto bývá u většiny schémat relativně snadné zajistit a dokázat.

Nás ovšem z pohledu praktického zajímá, jak rychle se x_k k té limitě r přibližují, tedy „rychlost konvergence metody“. Pro danou metodu hledáme odhad chyby $|r - x_{k+1}| \leq K|r - x_k|^a$, tomu pak říkáme **řád metody**.

Pokud uvažujeme chybu ve tvaru 10^{-k-1} , tedy situaci, kdy máme k správných míst, tak z odhadu zjistíme, jaká je zaručená přesnost další iterace. Pokud přimhouříme oko nad tím K (často je okolo jedničky), tak se dá (zhruba) říct, že řád metody a udává, kolikrát se každou iterací zvětší počet správných desetinných míst odhadu, jakmile už jednou nějaké správné máme. Mluvíme tedy o dost rychlé konvergenci, již řád $a = 2$ zhruba řečeno znamená, že každou iterací zdvojnásobíme počet správných desetinných míst!

Někdy se také zadaří najít odhad ve tvaru $|r - x_{k+1}| \leq k|x_{k+1} - x_k|$. To je velice užitečné, protože číslo $|x_{k+1} - x_k|$ máme samozřejmě k dispozici, takže dokážeme spolehlivě najít odhad kořene r se zadanou přesností, tedy umíme zaručit, že $|r - x_k| < \varepsilon$.

1a. Metoda půlení intervalů (metoda bisekce, Bisection method)

Tato metoda vychází z vlastnosti mezihodnoty pro spojitou funkci.

Fakt.

Nechť f je funkce na intervalu $\langle a, b \rangle$. Jestliže $f(a) \cdot f(b) < 0$ (tj. $f(a)$ a $f(b)$ mají rozdílná znaménka) a f je spojitá na $\langle a, b \rangle$, pak f musí mít v intervalu $\langle a, b \rangle$ kořen.

Toto se spojí s pozorováním, že když vezmeme interval, na kterém funkce mění znaménko, a rozdělíme jej na poloviny, tak alespoň jedna z těchto polovin zase musí mít změnu znamének na krajích, teď už máme ovšem kořen vymezen s poloviční chybou. Toto se dá opakovat, dokud není kořen vymezen dostatečně přesně.

Algoritmus (metoda bisekce pro hledání kořene funkce f).

Zadána funkce f spojitá na intervalu $\langle a, b \rangle$ a tolerance ε . Předpoklad: Znaménka $f(a)$ a $f(b)$ jsou různá.

0. Zavedeme $x_0 = a$, $y_0 = b$. Nechť $k = 0$.

1. Předpoklad: znaménka $f(x_k)$ a $f(y_k)$ jsou různá.

Nechť $m_k = \frac{1}{2}(x_k + y_k)$.

2. Jestliže $f(m_k) = 0$ nebo $|y_k - x_k| < \varepsilon$, tak konec, výstup je m_k .

Jestliže jsou znaménka $f(x_k)$ a $f(m_k)$ různá, nechť $x_{k+1} = x_k$, $y_{k+1} = m_k$, zvýšíme k o jedničku a jdeme zpět na krok 1.

Jestliže jsou znaménka $f(m_k)$ a $f(y_k)$ různá, nechť $x_{k+1} = m_k$, $y_{k+1} = y_k$, zvýšíme k o jedničku a jdeme zpět na krok 1.

△

V praxi samozřejmě $f(m_k) = 0$ ve skutečnosti znamená $|f(m_k)| < \delta$, ale i tak je šance na takto šťastnou trefu malá. Výpočet tedy typicky končí dosažením žádané přesnosti odhadu. Jako odhad bychom mohli brát libovolné číslo mezi x_k a y_k , to m_k se nabízí. Alternativní podmínka ukončení je $|f(m_k)| < \varepsilon$, pro jistotu je možné vyžadovat obojí.

Věta.

Nechť spojitá funkce f na $\langle a, b \rangle$ splňuje $f(a) \cdot f(b) < 0$. Pak posloupnost $\{m_k\}$ generovaná metodou bisekce se vstupními hodnotami $x_0 = a$, $y_0 = b$ konverguje ke kořeni r funkce f .

Metoda bisekce je lineárního řádu.

Lineární metody jsou pro svou konvergenci závislé na konstantě K , zde $K = \frac{1}{2}$. Ze známých metod je tato ve skupině těch nejpomalejších. Obvykle nás zajímá, jak rychle při iteraci přibývají desetinná místa, která jsou již dobře, jinými slovy jak rychle se chyba zmenší desetkrát. U metody bisekce dostáváme jedno nové správné místo s každou iterací, ovšem ve dvojkové soustavě, my bychom rádi spíš desítkovou. Obecně vycházíme-li z chyby e_k , pak $e_{k+1} = \frac{1}{2}e_k$, $e_{k+2} = \frac{1}{4}e_k$, $e_{k+3} = \frac{1}{8}e_k$, $e_{k+4} = \frac{1}{16}e_k$. Vidíme, že ani tři iterace nestačily na vytvoření nového správného desetinného místa. Můžeme zkusit položit $2^k = 10$ a zjistíme, že v zásadě (v dlouhodobém průměru) je třeba $\log_2(10) \sim 3.32\dots$ iterací na desetinné místo. Obdobnou informaci získáme, když si všimneme, že $2^{10} = 1012 \sim 10^3$, tedy deseti iteracemi získáme 3 zaručená místa. Odtud pak tradiční tvrzení, že bisekce potřebuje cca $3\frac{1}{3}$ iterací na desetinné místo.

Na druhou stranu je metoda bisekce zcela spolehlivá a navíc snadno testujeme, zda je odhad dostatečně blízko skutečného kořene. Proto je tato metoda přes svou jednoduchost základem. Často se používá jako metoda startovací, kdy si kořen lokalizujeme do malého prostoru, kde jej pak výrazně zpřesníme metodou rychlejší.

1b. Metoda regula falsi (falešná pozice, false position)

I zde budeme interval opakovaně dělit na části tak, aby se střídala znaménka, ale nebudeme púlit. V k -tém kroku metody máme k dispozici hodnoty $f(x_k)$ a $f(y_k)$ budeme rozdělovat tento interval v poměru daném dotýčnými hodnotami.

Jinak řečeno (ale se stejným výsledkem), známými body grafu $(x_k, f(x_k))$ a $(y_k, f(y_k))$ proložíme úsečku a její průnik s osou x bude dobrým odhadem kořene, který použijeme pro další rozhodování. Rovnice dotýčné přímky je

$$y = f(x_k) + \frac{f(y_k) - f(x_k)}{y_k - x_k}(x - x_k).$$

Osu x protne v bodě

$$x = \frac{(f(y_k) - f(x_k))x_k - f(x_k)(y_k - x_k)}{f(y_k) - f(x_k)} = \frac{x_k f(y_k) - y_k f(x_k)}{f(y_k) - f(x_k)}.$$

V ideálním případě by tento bod měl být poblíž hledaného kořene.

Algoritmus (metoda regula falsi pro hledání kořene funkce f).

Zadána funkce f spojitá na intervalu $\langle a, b \rangle$ a tolerance ε . Předpoklad: znaménka $f(a)$ a $f(b)$ jsou různá.

0. Zavedeme $x_0 = a$, $y_0 = b$. Nechť $k = 0$.

1. Předpoklad: znaménka $f(x_k)$ a $f(y_k)$ jsou různá.

Nechť $r_k = \frac{x_k f(y_k) - y_k f(x_k)}{f(y_k) - f(x_k)}$.

2. Jestliže $|f(r_k)| < \varepsilon$, tak konec, výstup je r_k .

Jestliže jsou znaménka $f(x_k)$ a $f(r_k)$ různá, nechť $x_{k+1} = x_k$, $y_{k+1} = r_k$, zvýšíme k o jedničku a jdeme zpět na krok 1.

Jestliže jsou znaménka $f(r_k)$ a $f(y_k)$ různá, nechť $x_{k+1} = r_k$, $y_{k+1} = y_k$, zvýšíme k o jedničku a jdeme zpět na krok 1.

△

Proč jsme nečekali, až $|y_k - x_k| < \varepsilon$? Protože k tomu vůbec nemusí dojít! To ale neznamená, že by nekonvergovala posloupnost $\{r_k\}$. Abychom zajistili konvergenci, je třeba mít vhodnou kombinaci konvexity a monotonie funkce.

Věta.

Předpokládejme, že funkce f je taková, že $f(a)f(b) < 0$ a f'' nemění na intervalu $\langle a, b \rangle$ znaménko. Uvažujme posloupnosti $\{x_k\}$, $\{y_k\}$ a $\{r_k\}$ vytvořené pomocí metody regula falsi se vstupními hodnotami $x_0 = a$, $y_0 = b$. Pak posloupnost $\{r_k\}$ konverguje ke kořeni r funkce f . Konvergence metody regula falsi je lineární.

Podobně jako metoda bisekce, i regula falsi konverguje nejpomaleji u ostře zalomených funkcí, kdy zlom je těsně pod osou x . Obecně není regula false lepší než bisekce, často je horší (i výrazně).

1c. Metoda sečen:

V metodě regula falsi jsme ze znalosti dvou bodů grafu získávali odhad, kde by tak mohl ležet kořen. Nabízí se nápad tuto znalost využít přímo. Nedostáváme pak pro kořen nový rozsah, ale bodový odhad, budeme tedy pracovat s jednou posloupností, u které vždy pomocí dvou posledních známých bodů vygenerujeme nový. Evidentně budeme na inicializaci potřebovat dva odhady, tentokrát bez požadavků na znaménka.

Vzorec pro nový bod se tedy odvozuje již známým vzorcem

$$l(x) = f(x_{k-1}) + \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}(x - x_k)$$

pro spojnicí dvou bodů grafu funkce daných hodnotami x_{k-1} , x_k .

Algoritmus (metoda sečen pro hledání kořene funkce f).

Zadána funkce f spojitá na \mathbb{R} a tolerance ε .

0. Zvolíme x_0 , x_1 . Nechť $k = 1$.

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

Jestliže $|f(x_{k+1})| < \varepsilon$, algoritmus končí, výstup je x_{k+1} .

Jinak zvýšíme k o jedničku a jdeme zpět na krok 1.

△

Někdy se také končí, pokud $|x_{k+1} - x_k| < \varepsilon$. Je nutno hlídat dva kritické stavy. Pokud se stane, že $f(x_{k+1}) = 0$, pak bychom následně dostali $x_{k+2} = x_{k+1}$ a máme problém s dělením nulou, je to ovšem vlastně varianta dobrá, protože jsme našli kořen. Pokud bychom chtěli být formální, dodefinujeme všechna další x_j pro $j > k + 1$ jako $x_j = r$ a dostáváme posloupnost, která má stejné vlastnosti jako v obecném běhu metody.

Druhý problém je situace, kdy $f(x_k) = f(x_{k-1})$. To je ovšem problém fatální a ukazuje, že metoda nemusí konvergovat. Dokonce může selhat i v případě, kdy takovéto problémy s během nenastanou, klidně se může stát, že algoritmus generuje nová x_k , ale k ničemu to není. Hodně záleží na průběhu funkce, opět to chce vhodnou kombinaci konvexity a monotonie.

Věta.

Jestliže je funkce f spojitá a konvexní či konkávní na intervalu $\langle a, b \rangle$ a $f(a)f(b) < 0$, pak pro libovolné hodnoty x_0, x_1 zvolené tak, aby $f(x_0) > 0, f(x_1) > 0$ pro konvexní případ a $f(x_0) < 0, f(x_1) < 0$ pro konkávní případ, posloupnost $\{x_k\}$ generovaná metodou sečen konverguje ke kořeni r funkce f .

Nyní se podíváme na rychlost konvergence.

Věta.

Nechť je funkce f dvakrát spojitě diferencovatelná na nějakém okolí U svého kořene r . Nechť $\{x_k\}$ je posloupnost generovaná metodou sečen taková, že $x_k \rightarrow r$. Jestliže je kořen r jednoduchý, pak existuje konstanta C a $N \in \mathbb{N}$ takové, že $|r - x_{k+1}| \sim k|r - x_k|^\alpha$ pro $k \geq N$, kde $\alpha = \frac{1+\sqrt{5}}{2}$.

Číslo $\alpha \sim 1.6$ je hodnota zlatého řezu a vidíme, že u jednoduchých kořenů je metoda sečen výrazně lepší než lineární. Pokud je r vícenásobný, tak už se chyba chová lineárně.

Metoda sečen má problém, že když se blížíme ke kořeni, tak se odečítají téměř shodná čísla, tudíž hrozí nebezpečí, že se teoreticky dobrá konvergence neguje vzrůstem zaokrouhlovacích chyb. Je třeba být ostražitý.

1d. Newtonova metoda (metoda tečen, Newton method)

Tato metoda používá jedinou posloupnost. Máme-li bod x_k , sestrojíme v něm tečnu ke grafu,

$$y = f(x_k) + f'(x_k)(x - x_k).$$

Její průsečík s osou x bude další odhad kořene.

Algoritmus (Newtonova metoda pro hledání kořene funkce f).

Zadána funkce f diferencovatelná na intervalu \mathbb{R} a tolerance ε .

0. Zvolíme x_0 . Nechť $k = 0$.

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

Jestliže $|x_{k+1} - x_k| < \varepsilon$ nebo také $|f(x_{k+1})| < \varepsilon$, tak skončíme, výstup je x_{k+1} .

Jinak zvýšíme k o jedničku a jdeme zpět na krok **1**.

△

Podobně jako u metody sečen, i zde si musíme hlídat problémy (jmenovitě $f'(x_k) = 0$) a metoda nemusí konvergovat, pokud tvar funkce není zrovna ideální. I zde pomůže rozumná kombinace monotonie a konvexity/konkávity.

Věta.

Nechť f je funkce na intervalu $\langle a, b \rangle$ taková, že $f(a) \cdot f(b) < 0$. Předpokládejme, že f je na $\langle a, b \rangle$ dvakrát spojitě diferencovatelná, $f' \neq 0$ a f'' na tomto intervalu nemění znaménko.

Jestliže je $x_0 \in \langle a, b \rangle$ zvoleno tak, aby $f(x_0) \cdot f''(x_0) > 0$, pak posloupnost $\{x_n\}$ daná Newtonovou metodou konverguje k nějakému kořeni r funkce f na intervalu $\langle a, b \rangle$.

Protože funkce f' má vlastnost mezihodnoty, podmínka z věty znamená, že buď $f' > 0$ na $\langle a, b \rangle$, tedy funkce je vždy rostoucí, nebo je naopak vždy klesající.

Pokud už Newtonova metoda konverguje, pak (s trochou štěstí) velice rychle.

Věta.

Nechť r je kořen funkce f , která je dvakrát spojitě diferencovatelná na okolí U bodu r , pro které existují $m_1 > 0$ a M_2 splňující $|f'| \geq m_1$ a $|f''| \leq M_2$ na U . Uvažujme posloupnost $\{x_k\}$ generovanou Newtonovou metodou. Pak pro všechna $x_{k+1}, x_k \in U$ platí $|r - x_{k+1}| \leq \frac{M_2}{m_1} |r - x_k|^2$ a $|r - x_{k+1}| \leq \frac{M_2}{2m_1} |x_{k+1} - x_k|^2$.

Důkaz: Vzorec z definice se pomocí $x_k = r - e_k$ přepíše na

$$e_{k+1} = e_k + \frac{f(x_k)}{f'(x_k)}.$$

Podle věty o střední hodnotě existuje mezi x_k a r číslo ξ splňující $f(x_k) - f(r) = f'(\xi)(x_k - r)$, ale protože r je kořen, dostáváme $f(x_k) = -f'(\xi)e_k$. Po dosazení

$$e_{k+1} = e_k \left(1 - \frac{f'(\xi)}{f'(x_k)} \right) = e_k \frac{f'(x_k) - f'(\xi)}{f'(x_k)}.$$

Opět věta o střední hodnotě, najdeme μ tak, aby $f'(x_k) - f'(\xi) = f''(\mu)(x_k - \xi)$. Protože ξ je mezi x_k a r , je $|x_k - \xi| \leq |e_k|$, tedy

$$e_{k+1} = e_k \frac{f''(\mu)}{f'(x_k)}(x_k - \xi) \implies |e_{k+1}| \leq |e_k|^2 \frac{|f''(\mu)|}{|f'(x_k)|}.$$

Odtud pak vyplýne odhad řádu metody.

Druhý odhad: Podle definice posloupnosti z Newtonovy metody platí $f(x_k) + f'(x_k)(x_{k+1} - x_k) = 0$, díky čemuž se zajímavě zjednoduší Taylorův rozvoj se středem x_k s Lagrangeovým tvarem zbytku:

$$f(x_{k+1}) = f(x_k) + f'(x_k)(x_{k+1} - x_k) + \frac{1}{2}f''(\xi)(x_{k+1} - x_k)^2 = 0 + \frac{1}{2}f''(\xi)(x_{k+1} - x_k)^2.$$

Toto se dosadí do univerzálního odhadu chyby výše a je to. □

Již dříve jsme rozebrali, že dolní odhad pro $|f'|$ souvisí s násobností kořene. Pokud je tedy kořen jednoduchý, pak v okamžiku, kdy se již Newtonova metoda dostane blízko ke kořeni, začne být konvergence velice rychlá, Newtonova metoda je v této situaci kvadratického řádu. Zhruba řečeno se při každé iteraci zdvojnásobí počet korektně odhadnutých desetinných míst.

Druhý odhad nám umožňuje čekat při iteraci, dokud nenastane $|r - x_k| < \varepsilon$.

Na kvadraticou konvergenci už nelze spoléhat, pokud nejsme blízko kořene, proto se často používá nějaká startovací metoda (třeba bisekce). Dalším problémem jsou vícenásobné kořeny, kde konvergence padá na lineární.

Příklad:

1. Chceme-li najít \sqrt{A} , pak volba $f(x) = x^2 - A$ a Newtonova metoda dávají rekurentní vzorec $x_{k+1} = \frac{x_k}{2} + \frac{A}{2x_k}$, který velice rychle konverguje.

2. Počítání inverze $\frac{1}{a}$ je výpočetně náročné, zajímavou alternativou je najít kořen funkce $f(x) = \frac{1}{x} - a$, vychází $x_{k+1} = x_k(2 - ax_k)$. Tomuto se říká Newtonovo-Raphstonovo dělení. Z výpočetního pohledu se lépe chová $x_{k+1} = x_k + x_k(1 - ax_k)$. Vyplatí se také naškálovat problém tak, aby $a \in \langle 0.5, 1 \rangle$, pak se doporučuje iniciační hodnota $x_0 = \frac{48}{17} - \frac{32}{17}a$.

△

Poznamenejme, že Newtonova metoda funguje i u komplexních funkcí. Pokud chceme najít komplexní kořen reálné funkce, je nutno vzít x_0 s nenulovou imaginární částí.

2. Pevné body

Je-li dána funkce φ , pak jejím pevným bodem rozumíme libovolné číslo x_f splňující $\varphi(x_f) = x_f$. Souvislost s tématem je zjevná, pevný bod je totiž kořenem rovnice $\varphi(x) - x = 0$, naopak kořen funkce f lze hledat například zkoumáním pevného bodu $f(x) + x = x$. Převodů mezi těmito dvěma pojmy je více a ještě uvidíme, že se snažíme o převod s co nejvhodnějším výsledkem.

Pojem pevného bodu je obecnější než pojem kořenu, protože pevný bod lze zavést pro libovolné zobrazení z množiny M do M . Jde o důležité téma v mnoha oblastech matematiky, zajímavé je to například při zobrazení mezi vícedimenzionálními prostory, podívejme se tedy na nejoblíbenější metodu nalézání pevného bodu.

2a. Metoda prosté iterace

Začneme číslem x_0 , dosazením získáme $x_1 = \varphi(x_0)$, dalším dosazením $x_2 = \varphi(x_1)$ atd. Pokud se výsledky stabilizují, pak máme $x_{k+1} \sim x_k$ neboli $\varphi(x_k) \sim x_k$ a tím (doufáme) i skoro hledané číslo x_f . Je neuvěřitelné, že to občas funguje.

Věta.

Nechť φ je funkce, $x_0 \in \mathbb{R}$ a $x_{k+1} = \varphi(x_k)$ pro $i \in \mathbb{N}$. Jestliže $x_k \rightarrow x_f$ a φ je spojitá v x_f , pak $\varphi(x_f) = x_f$.

Důkaz: Z předpokladu $x_k \rightarrow x_f$ dostáváme také $x_{k+1} \rightarrow x_f$. Protože $x_{k+1} = \varphi(x_k)$, můžeme počítat

$$x_f = \lim_{k \rightarrow \infty} (x_{k+1}) = \lim_{k \rightarrow \infty} (\varphi(x_k)) = \varphi\left(\lim_{k \rightarrow \infty} (x_k)\right) = \varphi(x_f).$$

Vytáhnutí funkce φ z limity bylo korektní, protože je spojitá. □

Tato věta platí velice obecně, kdykoliv pracujeme se spojitým zobrazením z nějaké množiny do téže množiny, na které lze zavést pojem limity. Problém samozřejmě je, jak donutit vznikající posloupnost ke konvergenci. Než se na to podíváme blíže, ukážeme si geometrické souvislosti.

Poznámka: Funkci si často představujeme pomocí jejího grafu. Pevný bod neboli řešení rovnice $\varphi(x) = x$ je vlastně průsečíkem dvou grafů, $y = \varphi(x)$ a $y = x$, tedy hledáme průsečík grafu dané funkce a hlavní diagonály.

Graficky prostá iterace funguje jednoduše. Iniciace: Najdeme bod odpovídající x_0 na diagonále. Iterace pak vypadá následovně: Ze známého bodu na diagonále se svisle dostaneme na graf φ , z tohoto bodu pak vodorovně na diagonálu a máme x_1 . Zase svisle na φ , vodorovně na diagonálu a máme x_2 , a tak pořád dokola, čekáme, zda se takto dokážeme přiblížit k průsečíku grafu φ s hlavní diagonálou.

△

Když si toto vyzkoušíme s několika grafy, tak brzy zjistíme, že konvergence závisí na tvaru. Je ovšem velice těžké vystihnout, které tvary jsou příhodné.

Příklad: Podíváme se na nejjednodušší funkce, které lze za φ vzít, tedy mocniny, a budeme uvažovat počáteční iterace $x_0 > 0$.

Varianta 1. $\varphi(x) = x^a$ vede na rovnici $x = x^a$, to je moc jednoduché. Uvažujme tedy $\varphi(x) = x^a$ pro $a > 0$. Hledáme tedy x_f splňující $x^a = x$ neboli nulu či jedničku. Jak vypadá vzniklá posloupnost? Takto:

$$\{x_0, x_0^a, x_0^{a^2}, x_0^{a^3}, \dots\}$$

neboli $x_k = (x_0)^{a^k}$. Jak se chová? Rozhodně je monotonní, dále záleží na a .

1a. Pro $0 < a < 1$ je $a^k \rightarrow 0$, tedy $x_k \rightarrow x_0^0 = 1$ pro $x_0 > 0$.

1b. Pro $a > 1$ je $a^k \rightarrow \infty$, tedy $x_k \rightarrow x_0^\infty$ a záleží na volbě x_0 . Pro $|x_0| < 1$ posloupnost $\{x_k\}$ konverguje k nule, pro $x_0 > 1$ konverguje do nekonečna.

Varianta 2. Uvažujme teď $\varphi(x) = \frac{1}{x^a}$ pro $a > 0$. Řešením rovnice $\frac{1}{x^a} = x$ je teď jen jednička. Pro $x_0 > 0$ dostáváme posloupnost

$$\left\{x_0, \frac{1}{x_0^a}, x_0^{a^2}, \frac{1}{x_0^{a^3}}, x_0^{a^4}, \dots\right\}.$$

Jak se chová? Pro $x_0 \neq 1$ jsou to čísla střídavě větší a menší než 1, tedy tato posloupnost určitě nebude monotonní.

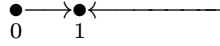
2a. Pro $0 < a < 1$ je $a^k \rightarrow 0$, tedy $(x_0)^{a^k} \rightarrow x_0^0 = 1$ pro $x_0 > 0$, $x_0 \neq 1$. Proto $x_k \rightarrow 1$.

2b. Pro $a > 1$ je $a^k \rightarrow \infty$, tedy $(x_0)^{a^k} \rightarrow x_0^\infty$ a pro volbu $x_0 \neq 1$ bude posloupnost x_k střídavě utíkat k nule a do nekonečna.

△

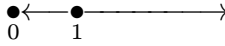
Poznámka: Jedna ze zajímavých oblastí matematiky, které se ale u pevných bodů nebudeme věnovat, je stabilita řešení. Rovnice $\varphi(x) = x$ popisuje jakýsi vyvíjející se systém a pevné body jsou body rovnováhy (ekvilibria). Jak ale víme z fyziky, existují body stabilní a nestabilní, což se pozná podle toho, jak se systém zachová, pokud se z rovnovážné polohy trochu vychýlíme. Informace se dá rychle vyčíst z tzv. fázového modelu („phase line“, obecněji „phase space“), ukážeme to v následujícím ohlédnutí za předchzím příkladem.

1a. $x^a = x$ pro $0 < a < 1$. Zde byly dva stacionární body, 0 a 1. Pro libovolné x_0 platilo $x_k \rightarrow 1$. Tomu odpovídá tento obrázek:



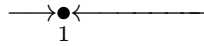
Vidíme, že 1 je stabilní stacionární bod, při výchylce nás tam systém vrátí, zatímco 0 je nestabilní.

1a. $x^a = x$ pro $a > 1$. Zde byly dva stacionární body, 0 a 1, a chování záleželo na volbě x_0 .



Vidíme, že 0 je stabilní stacionární bod, zatímco 1 je nestabilní.

2a. $\frac{1}{x^a} = x$ pro $0 < a < 1$. Zde byl jeden stacionární bod. Pro libovolné $x_0 > 0$ platilo $x_k \rightarrow 1$. Tomu odpovídá tento obrázek:



Vidíme, že 1 je stabilní stacionární bod.

2b. $\frac{1}{x^a} = x$ pro $a > 1$. Zde byl jeden stacionární bod 1, je nestabilní.

△

Viděli jsme, že situace může být zajímavá. Existuje jeden pojem, který dokáže pomoci.

Definice.

Nechť φ je funkce na intervalu I . Řekneme, že je tam **kontraktivní** nebo že je to **kontrakce**, jestliže existuje $q < 1$ takové, že pro všechna $x, y \in I$ platí

$$|\varphi(x) - \varphi(y)| \leq q \cdot |x - y|.$$

Například lineární funkce $y = ax + b$ je kontrakce přesně tehdy, když $a < 1$. Obecně snadno dokážeme, že všechny kontrakce jsou spojité. Koeficient kontrakce q je měřítkem kvality, čím blíže nule, tím víc funkce sblížuje hodnoty.

Věta. (Banachova o pevném bodě)

Nechť φ je kontraktivní funkce na $I = \langle a, b \rangle$ s koeficientem q taková, že $\varphi[I] \subseteq I$. Pak v I existuje právě jedno řešení x_f rovnice $\varphi(x) = x$, přičemž pro všechny volby $x_0 \in I$ posloupnost daná $x_{k+1} = \varphi(x_k)$ konverguje k x_f a platí

$$|x_f - x_{k+1}| \leq q|x_f - x_k| \quad \text{a} \quad |x_f - x_{k+1}| \leq \frac{q}{1-q}|x_{k+1} - x_k|.$$

Důkaz: Uvažujme $h(x) = x - \varphi(x)$. Je to spojitá funkce (neboť φ je kontrakce, tudíž spojitá). Protože $\varphi(a) \in I$, je $\varphi(a) \geq a$ neboli $h(a) \leq 0$, díky $\varphi(b) \leq b$ naopak $h(b) \geq 0$. Podle Věty o mezihodnotě existuje x_f splňující $h(x_f) = 0$.

Jednoznačnost: Jsou-li x_f a y_f řešení, pak

$$|x_f - y_f| = |\varphi(x_f) - \varphi(y_f)| \leq q|x_f - y_f|$$

neboli $(1 - q)|x_f - y_f| \leq 0$. Protože $1 - q > 0$, vychází $|x_f - y_f| \leq 0$, tedy $|x_f - y_f| = 0$.

Začneme-li s libovolným x_0 , pak odhadujeme $|x_f - x_{k+1}| = |\varphi(x_f) - \varphi(x_k)| \leq q|x_f - x_k|$, indukci pak $|x_f - x_k| \leq q^k|x_f - x_0| \rightarrow 0$. Každá taková posloupnost tedy nutně konverguje k x_f .

Lze také počítat

$$|x_f - x_{k+1}| = |\varphi(x_f) - \varphi(x_k)| \leq q|x_f - x_k| = q|(x_f - x_{k+1}) + (x_{k+1} - x_k)| \leq q|x_f - x_{k+1}| + q|x_{k+1} - x_k|$$

neboli

$$(1 - q)|x_f - x_{k+1}| \leq q|x_{k+1} - x_k|.$$

Odtud už vyplyne druhý odhad. □

Podle věty je to metoda lineární, takže jsme závislí na q , čím menší je q , tím rychleji daná posloupnost konverguje k x_f .

Druhá nerovnost z věty je užitečná pro ukončování algoritmu. Pro zadanou přesnost ε máme požadavek $|x_f - x_k| < \varepsilon$, díky výsledku věty vidíme, že tedy čekáme, až $|x_k - x_{k-1}| < \frac{1-q}{q}\varepsilon$.

Informaci o možném q lze dostat například takto:

Věta.

Nechť φ má na (vnitřku) intervalu I spojitou derivaci. Jestliže existuje $q < 1$ takové, že $|\varphi'(t)| \leq q$ na I^o , pak φ je kontrakce na I s koeficientem q .

Bonus: Jestliže je φ monotónní na I , pak jsou i všechny posloupnosti $\{x_k\}$ vzniklé iterací monotónní.

Důkaz: Pro libovolné $x, y \in I$ existuje podle Věty o střední hodnotě ξ mezi x a y , tedy $\xi \in I$ takové, že $|\varphi(x) - \varphi(y)| = |\varphi'(\xi)| \cdot |x - y|$. Předpoklad o φ' ukazuje, že φ splňuje podmínku kontrakce.

Teď předpokládejme, že φ je rostoucí, tedy $\varphi' > 0$ (klesající případ se dělá obdobně).

Pro pevný bod x_f máme $x_f - x_{k+1} = \varphi'(\xi)(x_f - x_k)$. Znaménko rozdílu $x_f - x_k$ je tedy vždy stejné a díky kontraktivitě je x_{k+1} blíže k x_f než x_k . Toto již dává monotonii posloupnosti. Podle toho, zda zvolíme x_0 větší či menší než x_f , vychází $\{x_k\}$ klesající či rostoucí. □

Tato věta není jen prostředek k rozeznávání kontrakcí, ale také návod, jak mít co nejrychlejší konvergenzi. Chceme co nejmenší q , tedy co nejmenší maximum derivace.

Přechod od $f(x) = 0$ k $\varphi(x) = x$ lze opravdu dělat mnoha tvůrčími způsoby, nejen tím nudným $x = x + f(x)$.

Příklad:

Uvažujme funkce f , která má na intervalu I nenulovou derivaci. Pak lze uvažovat

$$f(x) = 0 \iff -\frac{f(x)}{f'(x)} = 0 \iff x = x - \frac{f(x)}{f'(x)}.$$

Pokud na základě této rovnosti uděláme iterační schéma, dostaneme přesně Newtonovu metodu! O té víme, že konverguje často, například pro příhodné kombinace vlastností monotonie a konvexity, aniž by výsledná funkce φ byla kontrakcí.

Někdy se ke kontrakci dostaneme. Výše jsme Newtonovou metodou odvodili schéma $x_{k+1} = \frac{x_k}{2} + \frac{A}{2x_k}$ na výpočet čísla \sqrt{A} . Odpovídající funkce je $\varphi(x) = \frac{x}{2} + \frac{A}{2x}$, kde vychází $\varphi'(x) = \frac{1}{2} - \frac{A}{2x^2}$. Pokud pracujeme s čísly $x \sim \sqrt{A}$, tedy jsme blízko hledaného bodu, pak $\varphi'(x) \sim 0$, čímž se vysvětluje rychlá konvergence.

Úprava $\varphi'(x) = \frac{1}{2}(1 - \frac{A}{x^2})$ ukazuje, že $|\varphi'(x)| < 1$ pro x splňující $-1 < \frac{A}{x^2} < 3$, ovšem $A > 0$, takže dostáváme podmínku $|x|^2 > \frac{A}{3}$. Pokud tedy (pro $A > 1$) volíme například $x_0 = \frac{1}{2}A$, pracujeme s kontrakcí.

△

Je vidět, že není třeba se upínat na kontrakce, metoda přímé iterace překvapivě často konverguje i pro jiné funkce. Prostě to člověk zkusí a pokud dostane konvergující posloupnost, má vyhráno.

2b. Relaxace

Uvažujme rovnici $\varphi_0(x) = x$. Zvolíme-li číslo $\lambda \neq 0$, můžeme udělat následující fintu:

$$\varphi_0(x) = x \iff \lambda\varphi_0(x) = \lambda x \iff \lambda\varphi_0(x) + (1 - \lambda)x = x.$$

Dostáváme tak novou rovnici pro pevný bod s funkcí $\varphi(x) = \lambda\varphi_0(x) + (1 - \lambda)x$. Parametr λ vlastně říká, nakolik věříme klasické přímé iteraci, pokud zvolíme $\lambda = 1$, pak dostáváme $\varphi = \varphi_0$. Pokud zvolíme λ mezi 0 a 1, pak částečně bereme novou hodnotu z přímé iterace, ale pro jistotu konzervativně také bereme v úvahu hodnotu původní.

Pomocí vhodné volby tohoto parametru se můžeme pokusit dosáhnout, že $|\varphi'(x)| < 1$ na množině, kde nás tato iterační rovnice zajímá. Zajímavé je zvolit λ tak, aby $\varphi'(x_0) = 0$, ze vzorce $\varphi'(x) = \lambda\varphi_0'(x) + 1 - \lambda$ snadno odvodíme, že hledanou hodnotou je $\lambda = \frac{1}{1 - \varphi_0'(x_0)}$.

Tento nápad má teoretickou podporu v následujícím tvrzení.

Věta.

Uvažujme schéma $x_{k+1} = \varphi(x_k)$ pro $x_k \in U$, kde U je nějaké okolí pevného bodu x_f funkce φ takové, že φ je na U n -krát spojitě diferencovatelná. Jestliže platí $\varphi'(x_f) = \dots = \varphi^{(n-1)}(x_f) = 0$ a $\varphi^{(n)}(x_f) \neq 0$, pak daná iterační metoda konverguje s rychlostí řádu n .

Vrátíme se teď k rovnici $f(x) = 0$ neboli hledání kořene. Trik s relaxací pro tento speciální případ vypadá následovně.

$$f(x) = 0 \iff \lambda f(x) = 0 \iff x = x + \lambda f(x).$$

Máme tedy rovnici pro pevný bod s funkcí $\varphi(x) = x + \lambda f(x)$. I zde je motivací zvolit λ tak, aby $\varphi'(x) = 1 + \lambda f'(x)$ bylo na oblasti, kde iterujeme, co nejmenší.

Zajímavý nápad je měnit λ dynamicky v každém kroku iterace, tedy při známém x_k chtít $\varphi'(x_k) = 0$. Dostáváme $\lambda = -\frac{1}{f'(x_k)}$ a tudíž $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$, tedy Newtonovu metodu. Tím se vysvětlí její překvapivá účinnost, je to vlastně iterace, u které každý krok optimalizujeme rychlost konvergence.

2c. Iterační metody pro funkce více proměnných

Uvažujme vektorovou funkci $F: \mathbb{R}^n \mapsto \mathbb{R}^n$. Řekneme, že vektor \vec{x}_f je jejím pevným bodem, jestliže $F(\vec{x}_f) = \vec{x}_f$.

K hledání pevného bodu použijeme podobnou metodu jako v případě funkce jedné proměnné, tedy iterace $\vec{x}_{k+1} = F(\vec{x}_k)$. Opět doufáme, že tyto vektory konvergují k pevnému bodu, čímž se ovšem dostáváme k otázce, co to vlastně ta konvergence vektorů znamená. Uvažujme posloupnost vektorů $\vec{x}_k \in \mathbb{R}^n$ a $\vec{x} \in \mathbb{R}^n$, pro vektor \vec{y} teď budeme souřadnice značit $(\vec{y})_i$. Řekneme, že $\lim(\vec{x}_k) = \vec{x}$, také značeno $\vec{x}_k \rightarrow \vec{x}$, jestliže $\lim((\vec{x}_k)_i) = (\vec{x})_i$ pro všechna $i = 1, \dots, n$.

Existuje obecnější teorie o kontrakcích a Věta o pevném bodě, kde se pracuje s normou vektoru (viz kapitola o maticích).

Pokud danou rovnici přepíšeme na tvar $F(\vec{x}) = \vec{0}$, je možné použít Newtonovu metodu. V takovém případě se používá vzorec

$$\vec{x}_{k+1} = \vec{x}_k - J_F(\vec{x}_k)^{-1}F(\vec{x}_k),$$

kde $J_F(\vec{x}_k)^{-1}$ je inverzní matice k jakobiánu $J_F(\vec{x}_k)$. Většinou je rychlejší tuto matici nepočítat, ale rovnou řešit systém

$$J_F(\vec{x}_k)(\vec{x}_{k+1} - \vec{x}_k) = -F(\vec{x}_k).$$

3. Shrnutí

Viděli jsme metody spolehlivé a pomalé, a metody rychlé (často) a nespolehlivé. V praxi se často kombinují, nejprve spolehlivými (typicky metodou bisekce) přibližně určí poloha kořene, načež se rychle zvýší přesnost odhadu rychlejší metodou. Obecně bývá dobré během algoritmu sledovat a případně zasáhnout.

V mnoha případech je možné úlohu tvůrčím způsobem upravit a tím zlepšit její vlastnosti ohledně konvergence, to se týká zejména přechodu k pevnému bodu.

Rychlost konvergence nepříjemně klesá pro vícenásobné kořeny. Zajímavá finta: Funkce $\frac{f(x)}{f'(x)}$ má stejné kořeny jako f , ale teď jsou již všechny jednoduché. Metody tedy konvergují nejlépe, jak to jde, ale je nutné řešit odstranitelné nespojitosti a vůbec problémy s nulami ve jmenovateli.

Někdy je vhodné udělat rozbor tvaru funkce, tedy určit monotonii a lokální extrém, čímž se situace vyjasní.

U numerických metod obvykle diskutujeme chování chyb. Zde byly všechny metody iterativní, které nahrazují horší odhad řešení lepším. To znamená, že pokud vneseme jen malou chybu, tak se nic neděje, ona se v dalším kroku iterace nepraví. Může dojít k menšímu zpomalení konvergence, nic horšího. Tyto metody jsou tedy z hlediska numerického příznivé, jen si musíme ohlídat, že ve výpočtu nemáme zdroj větších chyb (pozor na dělení nulou u sečen/Newtona, rozdíl téměř stejných čísel u sečen).