Poglavje 3

Reševanje nelinearnih enačb

Na iskanje rešitve enačbe oblike

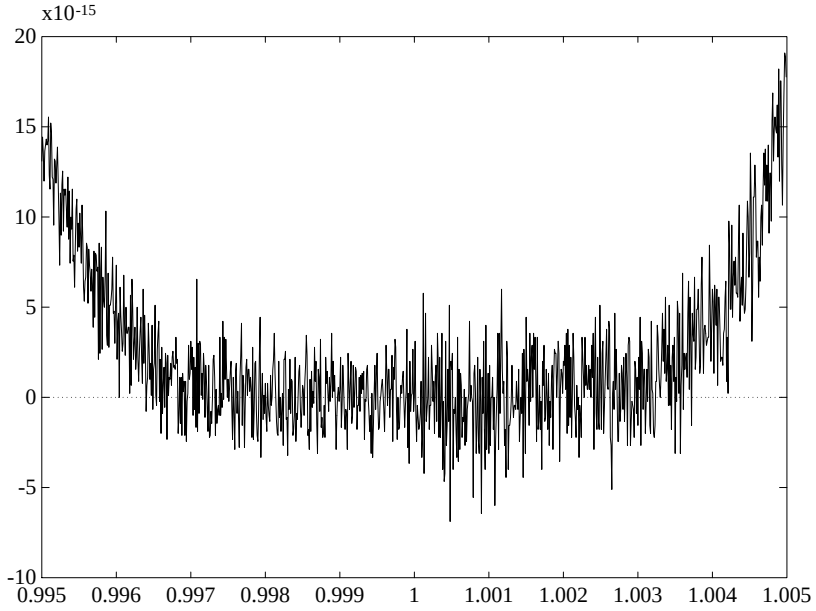
$$f(x) = 0 \tag{3.1}$$

zelo pogosto naletimo pri reševanju tehničnih problemov. Pri tem je lahko nelinearna funkcija f podana eksplicitno, pogosto pa je znan le postopek, kako pri določeni vrednosti neodvisne spremenljivke izračunamo njeno vrednost. Tako je $f(x)$ lahko vredost rešitve diferencialne enačbe v določeni točki ali vrednost nekega integrala. Nelinearno enačbo (3.1) lahko eksaktno rešimo le v zelo posebnih primerih, navadno se je moramo lotiti s kakšno od numeričnih metod. To pomeni, da ne bomo dobili točne vrednosti rešitve, ampak le njen približek. Kaj pomeni 'približek x^* k rešitvi enačbe' (3.1), pa je seveda odvisno od problema. Včasih je x^* taka točka, za katero je enačba (3.1) 'približno' izpolnjena, kar pomeni, da je $|f(x^*)|$ majhno število. Drugič pa je x^* 'blizu' točne rešitve enačbe (3.1). Poleg tega ima lahko enačba (3.1) več rešitev, večina numeričnih metod pa nam izračuna le eno do njih. Zgodi pa se tudi, da enačba (3.1) nima nobene rešitve

Pri numeričnem računanju ničle nelinearne funkcije se navadno ne moremo izogniti zaokrožitvenim napakam zaradi končne aritmetike, ki jo uporabljajo računalniki. Če izračunamo vrednosti polinoma

$$P_6(x) = 1 - 6x + 15x^2 - 20x^3 + 15x^4 - 6x^5 + x^6 \tag{3.2}$$

(slika 3.1) v bližini ničle $x = 1$ dobimo vtis, da ima ta polinom veliko število ničel na intevalu $(0.996, 1.004)$. Ta primer nam kaže, da ničle polinoma P_6 ne moremo določiti na več kot 2 mesti natančno, kljub temu, da smo vrednost



Slika 3.1: Izračunane vrednosti polinoma $(x-1)^6$ v bližini ničle

polinoma računali z natančnostjo 15 mest. Zaokrožitvena napaka je mnogo manjša, če ta polinom zapišemo v obliki

$$P_6(x) = (1-x)^6,$$

vendar moramo za tak zapis poznati vse ničle polinoma, torej vse ničle enačbe (3.1).

V tem poglavju si bomo ogledali metode za numerično reševanje nelinearnih enačb. Začeli bomo s preprostim problemom, ki nas privede do nelinearne enačbe, ob kateri si bomo ogledali nekaj splošnih lastnosti nelinearnih enačb. V naslednjih razdelkih bomo po vrsti opisali

1. Metodo bisekcije
2. Metodo regula falsi
3. Sekantno metodo
4. Newtonovo metodo

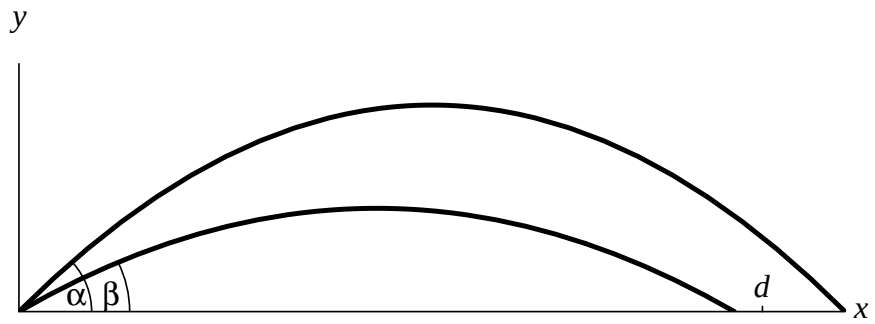
5. Kvazi-Newtonovo metodo

6. Metodo fiksne točke

Poglavje bomo končali s povzetkom najznačilnejših lastnosti opisanih metod.

3.1 Predstavitev problema

Za začetek si oglejmo preprost problem , pri katerem bomo spoznali nekaj najpogostejših problemov, s katerimi se lahko srečamo pri iskanju ničel nelinearnih enačb.



Slika 3.2: Streljanje s topom v tarčo

S topovsko kroglo želimo zadeti cilj na razdalji d od topa. Začetna hitrost izstrelka naj bo v_0, naklon cevi naj bo ϑ (glej sliko 3.2).

Začetna vertikalna hitrost granate je $v_0 \sin \vartheta$, iz fizike pa vemo, da je višina granate po času t enaka

$$y(t) = v_0 t \sin \vartheta - \frac{gt^2}{2},$$

kjer smo z g označili težnostni pospešek. Granata pade na tla po času

$$T = \frac{2v_0 \sin \vartheta}{g}.$$

Ker je vodoravna hitrost granate enaka $v_0 \cos \vartheta$ (zračni upor bomo zanemarili), bo domet granate enak $Tv_0 \cos \vartheta$. Da lahko določimo pravilen naklon

cevi, moramo torej rešiti enačbo

$$\frac{2v_0^2 \cos\vartheta \sin\vartheta}{g} = d,$$

oziroma

$$f(\vartheta) \equiv \frac{2v_0^2 \cos\vartheta \sin\vartheta}{g} - d = 0. \tag{3.3}$$

Ob tej enačbi si lahko ogledamo nekaj značilnosti, povezanih z nelinearnimi enačbami.

1. Enačba je poenostavljen model resničnega stanja. Pri izpeljavi smo upoštevali, da sta ustje topovske cevi in cilj v isti višini, kar velja le redko, zanemarili smo upor zraka in veter. Ko dobimo abstrakten numeričen problem, kot je enačba (3.3), se je koristno vprašati po njenem pomenu, še preden začnemo z numeričnim reševanjem.

2. Enačba morebiti sploh nima rešitve. Ker je produkt $\cos\vartheta \sin\vartheta$ lahko največ $\frac{1}{2}$ (pri $\vartheta = \frac{\pi}{4}$), enačba (3.3) za

   $$d > \frac{v_0^2}{g}$$

   nima nobene rešitve.

3. Rešitev, kadar obstaja, ni enolična, saj sta funkciji sin in cos periodični. Vsaka rotacija topovske cevi za poln krog spet pomeni rešitev. Pri reševanju se moramo zavedati tudi teh 'čudnih' rešitev.

4. Če je $d < v_0^2/g$ in $\vartheta_1 < \frac{\pi}{2}$, potem je tudi $\frac{\pi}{2} - \vartheta_1$ rešitev. Obe rešitvi sta smiselni, vendar je lahko za topničarja ena ugodnejša od druge. Ugotoviti moramo, katera.

5. Funkcija f je dovolj enostavna, da lahko izračunamo njen odvod, torej lahko uporabimo metodo, ki zahteva poleg poznavanja funkcije tudi poznavanje njenega odvoda (Newtonova metoda).

6. Enačbo (3.3) pravzaprav lahko rešimo tudi eksaktno, saj velja enakost $2\cos\vartheta \sin\vartheta = \sin 2\vartheta$. Le redko se zgodi, da nelinearno enačbo lahko rešimo eksaktno, vendar se splača poskusiti vsaj poenostaviti enačbo, preden jo začnemo reševati numerično.

7. Če naš model izpopolnimo, da bo bolje ustrezal dejanskemu problemu, npr. upoštevamo zračni upor, lahko dobimo sistem enačb, ki ga lahko rešimo le numerično. Funkcije postanejo tako zapletene, da ne moremo več dobiti analitičnega odvoda, torej moramo uporabiti takšno metodo, ki ne potrebuje vrednosti odvoda.

V praksi lahko topničar reši svoj problem z metodo poskusov in napak, tako da topovsko cev dviga ali spušča, dokler cilja ne zadane. Podobno delujejo tudi nekatere numerične metode.

3.2 Metoda bisekcije

Od sedaj naprej bomo reševali nelinearno enačbo (3.1). Metoda bisekcije sloni na naslednji lastnosti zvezne funkcije, ki jo že poznamo iz matematike:

Izrek 3.2.1. *Funkcija f naj bo zvezna na $[a, b]$ in naj bo g število med $f(a)$ in $f(b)$. Potem obstaja število $x \in [a, b]$, da je $g = f(x)$.*

Če lahko najdemo tak interval $[a, b]$, da je $f(a) \cdot f(b) < 0$, potem iz izreka 3.2.1 lahko sklepamo, da ima enačba (3.1) vsaj eno rešitev na intervalu $[a, b]$. Izračunamo jo s pomočjo naslednjega algoritma:

Algoritem 3.2.1 (Bisekcija)**.** Naj bo f zvezna funkcija na intervalu $[a, b]$ in naj bo $f(a)f(b) < 0$, potem naslednji algoritem izračuna število c, ki se od ničle funkcije f razlikuje za manj kot ε_x ali pa je vrednost $|f(c)| < \varepsilon_y$.

$c = (a + b)/2$; $z = f(a)$; $y = f(c)$
while (**abs**$(y) > \varepsilon_y$)&$(b - a > \varepsilon_x)$
    if $y * z < 0$
        $b = c$
    else
        $a = c$; $z = y$
    end
    $c = (a + b)/2$
    $y = f(c)$
end

Pri vsakem prehodu skozi zanko algoritma se interval, na katerem iščemo ničlo funkcije f razpolovi, zato se po k prehodih dolžina intervala zmanjša

na $(b-a)/2^k$. Ker zahtevamo, da se približek, izračunan z algoritmom Bisekcija razlikuje od točne ničle enačbe (3.1) za manj kot je vnaprej predpisana toleranca ε_x, se bo algoritem uspešno končal najmanj po

$$\log_2 \frac{b-a}{\varepsilon_x} + 1$$

korakih, če že prej nismo našli točke, v kateri je absolutna vrednost funkcije f manjša kot ε_y.

Ker je navadno časovno najzahtevnejši del opisanega algoritma računanje vrednosti funkcije f, velja omeniti, da moramo na vsakem koraku algoritma enkrat izračunati vrednost funkcije f.

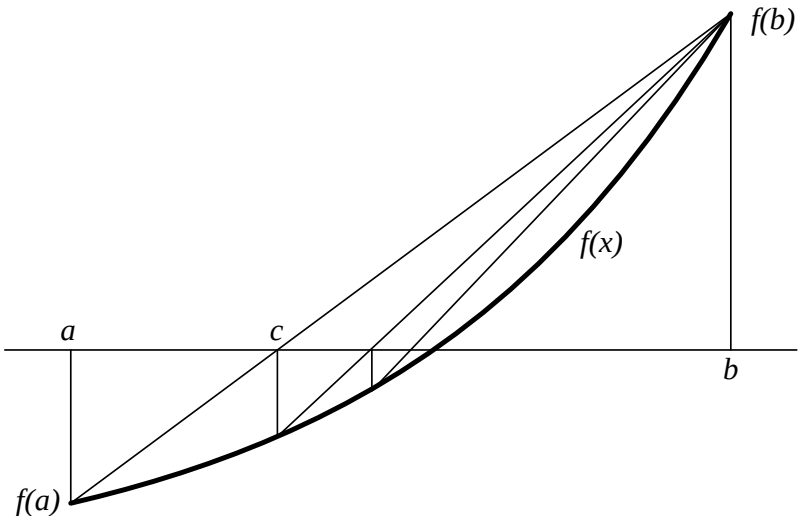
Primer 3.2.1. Izračunajmo rešitev enačbe

$$f(x) \equiv x^3 + 2x^2 + 10x - 20 = 0 \tag{3.4}$$

s pomočjo bisekcije. Izberimo si $\varepsilon_x = 10^{-6}, \varepsilon_y = 10^{-5}$. Ker sta vrednosti $f(0) = -20 < 0$ in $f(2) = 16 > 0$, lahko vzamemo za začetni interval $[0, 2]$. Rezultati računanja so zbrani v tabeli 3.1. ■

a	c	b	$f(c)$
0.000000	1.000000	2.000000	$-7.00 \cdot 10^{0}$
1.000000	1.500000	2.000000	$2.87 \cdot 10^{0}$
1.000000	1.250000	1.500000	$-2.42 \cdot 10^{0}$
1.250000	1.375000	1.500000	$1.30 \cdot 10^{-1}$
1.250000	1.312500	1.375000	$-1.16 \cdot 10{+}0$
1.312500	1.343750	1.375000	$-5.24 \cdot 10^{-1}$
1.343750	1.359375	1.375000	$-1.98 \cdot 10^{-1}$
1.359375	1.367187	1.375000	$-3.41 \cdot 10^{-2}$
1.367187	1.371093	1.375000	$4.82 \cdot 10^{-2}$
1.367187	1.369140	1.371093	$7.01 \cdot 10^{-3}$
1.367187	1.368164	1.369140	$-1.35 \cdot 10^{-2}$
1.368164	1.368652	1.369140	$-3.28 \cdot 10^{-3}$
1.368652	1.368896	1.369140	$1.86 \cdot 10^{-3}$
1.368652	1.368774	1.368896	$-7.10 \cdot 10^{-4}$
1.368774	1.368835	1.368896	$5.76 \cdot 10^{-4}$
1.368774	1.368804	1.368835	$-6.70 \cdot 10^{-5}$
1.368804	1.368820	1.368835	$2.54 \cdot 10^{-4}$
1.368804	1.368812	1.368820	$9.39 \cdot 10^{-5}$
1.368804	1.368808	1.368812	$1.34 \cdot 10^{-5}$
1.368804	1.368806	1.368808	$-2.67 \cdot 10^{-5}$
1.368806	1.368807	1.368808	$-6.64 \cdot 10^{-6}$

Tabela 3.1: Metoda bisekcije za enačbo $x^3 + 2x^2 + 10x - 20 = 0$



Slika 3.3: Metoda regula falsi

3.3 Metoda regula falsi

Pri metodi bisekcije interval na vsakem koraku razpolovimo, ne glede na to, kakšni sta vrednosti funkcije f na robovih intervala. Upamo lahko, da bomo ničlo funkcije hitreje našli, če bomo srednjo točko izbirali glede na vrednosti $f(a)$ in $f(b)$. Naj bo zopet $[a, b]$ tak interval, da je $f(a)f(b) < 0$, točko c pa si izberimo tam, kjer daljica skozi točki $T_0 = (a, f(a))$ in $T_1 = (b, f(b))$ seka abscisno os (glej sliko 3.3):

$$c = a - f(a)\frac{b-a}{f(b)-f(a)}. \tag{3.5}$$

Na ta način pridemo do algoritma, ki je znan pod imenom *regula falsi*.

Algoritem 3.3.1 (Regula falsi)**.** Naj bo f zvezna funkcija na intervalu $[a, b]$ in naj bo $f(a)f(b) < 0$. Naslednji algoritem izračuna število c, ki se od ničle funkcije f razlikuje manj od $\varepsilon_x > 0$ ali pa je $|f(c)| < \varepsilon_y$.

$f_a = f(a)$; $f_b = f(b)$
$c = a - f_a * (b - a)/(f_b - f_a)$
$f_c = f(c)$
while (**abs**$(f_c) > \varepsilon_y$)&$(b - a > \varepsilon_x)$
    if $f_a * f_c < 0$

$b = c;\ f_b = f_c$
else
$a = c;\ f_a = f_c$
end
$c = a - f_a * (b - a)/(f_b - f_a)$
$f_c = f(c)$
end

Primer 3.3.1. Izračunajmo rešitev enačbe (3.4) še z metodo regula falsi. Spet izberimo $\varepsilon_x = 10^{-6}, \varepsilon_y = 10^{-5}$ in začetni interval $[0, 2]$. Rezultati računanja so zbrani v tabeli 3.2.

a	c	b	$f(c)$
1.111111	1.324296	2.000000	$-9.27 \cdot 10^{-1}$
1.111111	1.324296	2.000000	$-9.27 \cdot 10^{-1}$
1.324296	1.361301	2.000000	$-1.58 \cdot 10^{-1}$
1.361301	1.367547	2.000000	$-2.65 \cdot 10^{-2}$
1.367547	1.368596	2.000000	$-4.46 \cdot 10^{-3}$
1.368596	1.368772	2.000000	$-7.48 \cdot 10^{-4}$
1.368772	1.368802	2.000000	$-1.25 \cdot 10^{-4}$
1.368802	1.368807	2.000000	$-2.10 \cdot 10^{-5}$
1.368807	1.368807	2.000000	$-3.53 \cdot 10^{-6}$
1.368807	1.368808	2.000000	$-5.93 \cdot 10^{-7}$

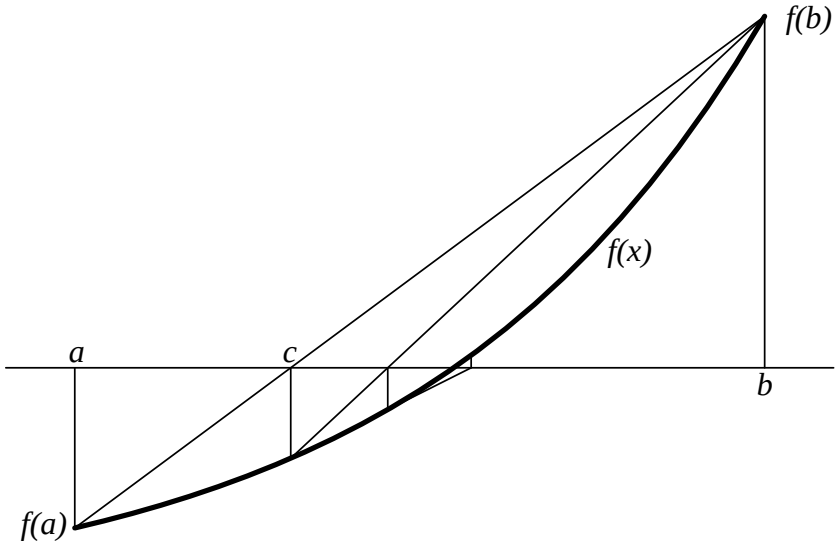
Tabela 3.2: Metoda regula falsi za enačbo $x^3 + 2x^2 + 10x - 20 = 0$

V primerjavi z bisekcijo (tabela 3.1) lahko vidimo, da približki, dobljeni z metodo regula falsi hitreje konvergirajo proti ničli funkcije f. Opazimo lahko tudi, da se spreminja le levo krajišče intervala, desno pa ostaja ves čas pri $x = 2$, kar je posledica konveksnosti funkcije v bližini iskane rešitve. ■

Metoda regula falsi deluje podobno kot bisekcija: interval, na katerem leži ničla funkcije f sistematično zmanjšujemo, dokler ga ne zmanjšamo pod vnaprej predpisano dolžino ε_x. Ker se dolžina intervala pri bisekciji vedno zmanjša na polovico, lahko vnaprej ocenimo potrebno število korakov iteracije, pri reguli falsi pa tega ne moremo. Vseeno pa približki, ki jih dobimo

s metodo regula falsi, večinoma konvergirajo proti ničli funkcije f hitreje kot približki, dobljeni z bisekcijo. Pri obeh metodah pa ves čas računanja poznamo interval, na katerem leži iskana ničla.

Hitrost konvergence regule falsi lahko povečamo, če se odpovemo kontroli predznaka funkcijske vrednosti v izračunanih točkah. S tem izgubimo interval, na katerem leži ničla, kar pomeni, da konvergenca zaporedja približkov ni več zagotovljena vnaprej. Tako dobimo *sekantno metodo* (slika 3.4):



Slika 3.4: Sekantna metoda

Algoritem 3.3.2 (Sekantna metoda)**.** Naj bo f zvezna funkcija. Če sta dani točki a in b, naslednji algoritem izračuna število c, za katerega je $|f(c)|$ manj od $\varepsilon_y > 0$, ali pa se po N korakih konča brez rezultata: $c = NaN$.

$x_s = a;\ f_s = f(x_s)$
$x_n = b;\ f_n = f(x_n)$
$n = 0$
while $(\mathbf{abs}(f_n) > \varepsilon_y)\&(n < N)$
    $c = x_s - f_s * (x_n - x_s)/(f_n - f_s)$
    $x_s = x_n;\ f_s = f_n$
    $x_n = c;\ f_n = f(c)$
    $n = n + 1$
end

if abs$(f_n) > \varepsilon_y$
$c = NaN$
end

Primer 3.3.2. Izračunajmo rešitev enačbe (3.4) še s sekantno metodo. Spet izberimo $\varepsilon_y = 10^{-5}$ in začetni interval $[0, 2]$. Rezultati računanja so zbrani v tabeli 3.3.

n	x_n	c	$f(c)$
1	2.000000	1.111111	$-5.04 \cdot 10^{1}$
2	1.111111	1.324296	$-9.27 \cdot 10^{-1}$
3	1.324296	1.372252	$7.27 \cdot 10^{-2}$
4	1.372252	1.368763	$-9.40 \cdot 10^{-4}$
5	1.368763	1.368808	$-9.37 \cdot 10^{-7}$

Tabela 3.3: Približki rešitve enačbe $x^3 + 2x^2 + 10x - 20 = 0$, dobljeni s sekantno metodo

V primerjavi z bisekcijo (tabela 3.1) in regulo falsi (tabela 3.2) lahko vidimo, da približki, dobljeni s sekantno metodo še hitreje konvergirajo proti ničli funkcije f. ■

Pri uporabi sekantne metode moramo biti previdni, saj se, posebej kadar smo še daleč od rešitve enačbe, prav lahko zgodi, da zaporedje približkov sploh ne konvergira.

3.4 Newtonova (tangentna) metoda

Enačbo (3.5), s katero izračunamo nov člen v zaporedju približkov z metodo regula falsi ali sekantno metodo, lahko zapišemo tudi kot

$$x_{n+1} = x_n + a_n,$$

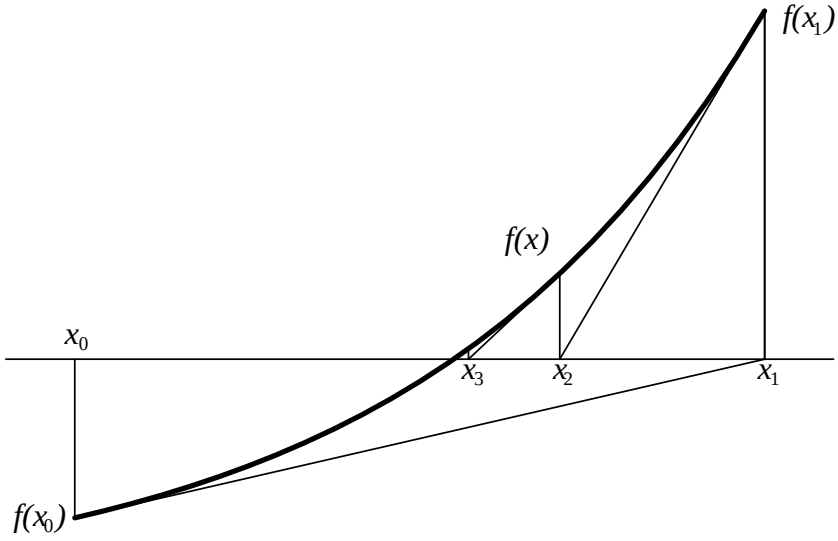
kjer je *popravek* a_n približka x_n določen kot

$$a_n = \frac{-f(x_n)}{c_n}.$$

Diferenčni kvocient $c_n = (f(x_n) - f(x_{n-1}))/(x_n - x_{n-1})$ v imenovalcu pomeni naklonski kot sekante skozi točki $T_{n-1} = (x_{n-1}, f(x_{n-1}))$ in $T_n = (x_n, f(x_n))$. Vemo tudi, da je v primeru, ko je funkcija f odvedljiva, diferenčni kvocient c_n enak odvodu $f'(x')$ v neki točki x' med x_n in x_{n-1}. Če diferenčni kvocient $(f(b) - f(a))/(b - a)$ v formuli (3.5) zamenjamo z vrednostjo odvoda v točki x_n, dobimo formulo

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \tag{3.6}$$

ki predstavlja osnovo *Newtonove*[1] ali *tangentne* metode. (slika 3.5)



Slika 3.5: Newtonova metoda

Algoritem 3.4.1 (Newtonova metoda)**.** Naj bo f zvezno odvedljiva funkcija. Če je dana točka a, naslednji algoritem izračuna število c, za katerega je $|f(c)|$ manj od $\varepsilon_y > 0$, ali pa se po N korakih konča brez rezultata: $c = NaN$.

$c = a$; $z = f(c)$
$n = 0$

[1]Sir Isaac Newton (1643 Woolsthorpe – 1727 London), angleški matematik, fizik in astronom, eden najpomembnejših matematikov. Skupaj z Nemcem Gottfriedom Wilhelmom von Leibnizem velja za začetnika infinitezimalnega računa. Opis metode, ki je danes znana kot Newtonova je objavil leta 1987 v znameniti knjigi *Principia Mathematica*.

while (**abs**$(z) > \varepsilon_y$)&$(n < N)$
    $c = c - z/f'(c)$
    $z = f(c)$
    $n = n + 1$
end
if abs$(z) > \varepsilon_y$
    $c = NaN$
end

Primer 3.4.1. Izračunajmo rešitev enačbe (3.4) z Newtonovo metodo. Ponovno naj bo $\varepsilon_y = 10^{-5}$, začetna točka pa naj bo $x = 0$. Rezultati računanja so zbrani v Tabeli 3.4.

n	x_n	z
1	2.000000	$1.60 \cdot 10^{1}$
2	1.466666	2.12
3	1.371512	$5.70 \cdot 10^{-2}$
4	1.368810	$4.46 \cdot 10^{-5}$
5	1.368808	$2.73 \cdot 10^{-11}$

Tabela 3.4: Približki rešitve enačbe $x^3 + 2x^2 + 10x - 20 = 0$, dobljeni z Newtonovo metodo

Če to primerjamo s sekantno metodo (Tabela 3.3) lahko vidimo, da približki, dobljeni z Newtonovo metodo še hitreje konvergirajo proti ničli funkcije f. ■

V primerjavi s prej omenjenimi metodami, moramo na vsakem koraku Newtonove metode izračunati dve funkcijski vrednosti: $f(x_k)$ in $f'(x_k)$, kar lahko predstavlja hud problem, predvsem kadar

- je odvod f' težko izračunljiva funkcija ali
- je funkcija f podana z zapleteno formulo, tako da je velika verjetnost napake pri računanju odvoda in pri zapisu formule za izračun odvoda ali

- je funkcijska vrednost $f(x_k)$ rezultat daljšega numeričnega računanja. V tem primeru navadno odvoda sploh ne moremo izraziti v obliki formule.

Tem problemom se lahko izognemo, če namesto formule (3.6) uporabimo formulo

$$x_{n+1} = x_n - \frac{f(x_n)}{d_n}, \tag{3.7}$$

kjer je d_k primeren, lahko izračunljiv približek za $f'(x_n)$. Enačba (3.7) predstavlja osnovo kvazi-Newtonove metode. Glede na to, kako izberemo približek d_n, je poznanih veliko kvazi-Newtonovih metod. Često uporabljana je metoda, pri kateri za d_n vzamemo vrednost odvoda $d = f'(x_0)$ v začetni točki iteracije in računamo vse iteracije z isto vrednostjo d. Če po določenem številu iteracij ničle nismo našli, izračunamo novo vrednost odvoda in postopek ponovimo. Tako pridemo do naslednjega algoritma:

Algoritem 3.4.2 (Kvazi-Newtonova metoda)**.** Funkcija f naj bo zvezno odvedljiva. Če je dano število a, naslednji algoritem izračuna število c, za katerega je $|f(c)|$ manj od $\varepsilon_y > 0$, ali pa se po N korakih konča brez rezultata: $c = NaN$.

$c = a$; $z = f(a)$; $d = f'(a)$
$n = 0$; $k = 0$
while (**abs**$(z) > \varepsilon_y$)&$(n < N)$
    $c = c - z/d$
    $z = f(c)$
    if $k == K$
        $d = f'(c)$; $k = 0$
    end
    $n = n + 1$; $k = k + 1$
end
if abs$(z) > \varepsilon_y$
    $c = NaN$
end

3.5 Metoda fiksne točke

Newtonovo metodo imamo lahko za poseben primer *metode fiksne točke*. Če je $f'(x) \neq 0$ na nekem intervalu in definiramo zvezno funkcijo

$$g(x) = x - \frac{f(x)}{f'(x)}, \tag{3.8}$$

potem lahko enačbo (3.6) zapišemo enostavneje kot

$$x_{n+1} = g(x_n). \tag{3.9}$$

Če zaporedje, definirano z rekurzijsko formulo (3.9), konvergira proti vrednosti ξ, potem je

$$\xi = \lim_{n\to\infty} x_{n+1} = \lim_{n\to\infty} g(x_n) = g(\xi),$$

kar pomeni, da je ξ *negibna (fiksna) točka* funkcije g. Očitno velja, da je vsaka negibna točka iteracijske funkcije Newtonove metode (3.8) tudi rešitev enačbe $f(x) = 0$.

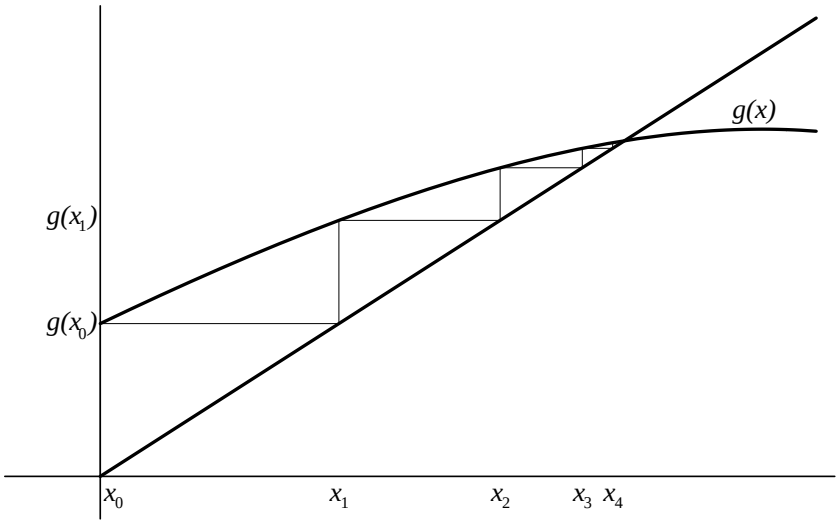
Metodo fiksne točke dobimo tako, da dano enačbo $f(x) = 0$ najprej zapišemo zapišemo v ekvivalentno enačbo oblike

$$x = g(x).$$

To lahko storimo na mnogo načinov. Enačbo (3.4) lahko, poleg drugih ekvivalentnih oblik, zapišemo tudi kot

$$\begin{aligned}
&\text{a)} \quad g(x) = \frac{20 - 2x^2 - x^3}{10} \\
&\text{b)} \quad g(x) = \frac{20 + 10x - 2x^2 - x^3}{20} \\
&\text{c)} \quad g(x) = \frac{1}{2}\sqrt{20 - 10x - x^3} \\
&\text{d)} \quad g(x) = \sqrt[3]{20 - 10x - 2x^2} \\
&\text{e)} \quad g(x) = x - \frac{x^3 + 2x^2 + 10x - 20}{c} \qquad \text{za poljuben } c \neq 0.
\end{aligned} \tag{3.10}$$

Ko imamo iteracijsko funkcijo g in začetni približek, lahko tvorimo zaporedje približkov po pravilu $x_{n+1} = g(x_n)$ (slika 3.6). Vprašati se moramo, kdaj tako dobljeno zaporedje (x_n) konvergira proti rešitvi enačbe (3.1). Brez dokaza (dokaz lahko radovedni bralec poišče na primer v [2]) navedimo *konvergenčni izrek*:



Slika 3.6: Metoda fiksne točke

Izrek 3.5.1. *Naj bo funkcija g na intervalu $[a, b]$ zvezno odvedljiva, njene vrednosti naj zadoščajo pogoju*

$$a \leq x \leq b \quad \Rightarrow \quad a \leq g(x) \leq b,$$

za odvode pa naj velja

$$\sup_{x \in [a,b]} |g'(x)| = \lambda < 1.$$

Potem velja:

1. *Enačba $x = g(x)$ ima natanko eno rešitev ξ na intervalu $[a, b]$;*

2. *Za vsak začetni približek $x_0 \in [a, b]$ bo zaporedje x_n konvergiralo proti ξ;*

3. $$|\xi - x_n| \leq \frac{\lambda}{1 - \lambda} |x_n - x_{n-1}|;$$

4. $$|\xi - x_n| \leq \frac{\lambda^n}{1 - \lambda} |x_0 - x_1|;$$

5.

$$\lim_{n\to\infty} \frac{\xi - x_{n+1}}{\xi - x_n} = g'(\xi),$$

kar pomeni, da je v bližini rešitve

$$\xi - x_{n+1} \approx g'(\xi)(\xi - x_n).$$

Ta izrek nam zagotavlja, da za vsak interval, na katerem je $|g'|$ manj od 1, zaporedje približkov, dobljeno s pravilom (3.9) vedno konvergira proti rešitvi enačbe $x = g(x)$.

Algoritem 3.5.1 (Metoda fiksne točke)**.** Dana naj bo zvezna funkcija g. Če je dana točka x, naslednji algoritem izračuna število c, za katerega je $|c-g(c)|$ manj od $\varepsilon_y > 0$, ali pa se po N korakih konča brez rezultata: $c = NaN$.

$n = 0$; $c = x$; $z = g(x)$
while (**abs**$(c - z) > \varepsilon_y$)&$(n < N)$
    $c = z$
    $z = g(c)$
    $n = n + 1$
end
if abs$(c - z) > \varepsilon_y$
    $c = NaN$
end

Primer 3.5.1. Izračunajmo rešitev enačbe (3.4) z metodo fiksne točke. Ponovno naj bo $\varepsilon_y = 10^{-5}$, začetna točka pa naj bo $x = 0$.

Najprej moramo izbrati primerno obliko iteracije. Od iteracijskih funkcij (3.10) so neprimerne a), c) in d), ker imajo v bližini ničle odvod, ki je po absolutni vrednosti večji kot 1 (glej problem 2). Iteracijska funkcija e) ima odvod, ki je po absolutni vrednosti manjši od 1 za $c < -0.6$, odvod iteracijske funkcije b) pa je blizu 0, zato izberemo iteracijo

$$x_{n+1} = \frac{20 + 10x_n - 2x_n^2 - x_n^3}{20}; \quad x_0 = 0. \tag{3.11}$$

Rezultati računanja so zbrani v tabeli 3.5.

Približki, dobljeni z iteracijo (3.11) konvergirajo proti ničli funkcije f nekoliko počasneje kot pri Newtonovi metodi. Pripomniti moramo, da je

n	x_n	$f(x_n)$
0	0.000000	$-2.00 \cdot 10^{1}$
1	1.000000	$-7.00 \cdot 10^{0}$
2	1.350000	$-3.94 \cdot 10^{-1}$
3	1.369731	$+1.94 \cdot 10^{-2}$
4	1.368757	$-1.07 \cdot 10^{-3}$
5	1.368811	$+5.87 \cdot 10^{-5}$
6	1.368808	$-3.22 \cdot 10^{-6}$

Tabela 3.5: Približki rešitve enačbe $x^3 + 2x^2 + 10x - 20 = 0$, dobljeni z iteracijo (3.11)

hitrost konvergence pri metodi fiksne točke tem večja, čim manjša je absolutna vrednost odvoda iteracijske funkcije v bližini ničle, kot napoveduje tudi izrek 3.5.1. ■

3.6 Povzetek

Povzemimo glavne značilnosti opisanih metod za reševanje nelinearnih enačb:

1. Za razliko od bisekcije, regule falsi in sekantne metode, pri katerih vedno potrebujemo dve točki, da izračunamo novi približek (zato jim pravimo dvočlenske metode), pri Newtonovi metodi in metodi fiksne točke potrebujemo le eno točko za izračun naslednjega približka. Zato pravimo, da sta Newtonova metoda in metoda fiksne točke enočlenski.

2. Pri bisekciji in pri reguli falsi imamo rešitev ves čas na omejenem intervalu, katerega dolžina se postopoma zmanjšuje, pri ostalih metodah med iteracijo nimamo zanesljive informacije o lokaciji rešitve, navadno je konvergenca približkov zagotovljena le v dovolj majhni okolici rešitve: če je začetni približek daleč od tiste rešitve enačbe, ki jo iščemo, zaporedje približkov lahko divergira ali pa konvergira proti kakšni drugi rešitvi.

3. Od vseh omenjenih metod najpočasneje konvergira zaporedje približkov, dobljenih z bisekcijo, nekoliko hitreje zaporedje, dobljeno z regulo falsi, še hitreje zaporedje, dobljeno s sekantno metodo, najhitreje pa zaporedje, dobljeno z Newtonovo metodo. Pri metodi fiksne točke je konvergenca približkov tem hitrejša, čim manjša je absolutna vrednost odvoda iteracijske funkcije.

4. Kot splošna metoda za reševanje nelinearnih enačb se najbolje obnese Newtonova metoda. Kadar je težko ali časovno zahtevno izračunati odvod funkcije f, je namesto Newtonove metode bolje uporabiti sekantno metodo.

5. Newtonova in sekantna metoda zanesljivo in hitro konvergirata le v neposredni bližini rešitve, zato je velikokrat ugodno začeti z zanesljivejšo, čeprav počasnejšo metodo (bisekcijo ali regulo falsi), ko pa se ničli dovolj približamo, računamo dalje s hitrejšo metodo (Newtonova ali sekantna).

6. Z Newtonovo in s sekantno metodo lahko izračunamo tudi kompleksne rešitve enačb. Potrebujemo le kompleksni začetni približek in (seveda) računati moramo s kompleksnimi števili.

3.7 Problemi

1. Število $\sqrt[n]{N}$ lahko izračunamo tako, da rešimo enačbo $x^n - N = 0$.
   (a) Izračunaj $\sqrt[3]{161}$ z metodo bisekcije. Kako izbrati začetni interval?
   (b) Izračunaj $\sqrt[4]{21.75}$ z metodo regula falsi. Kako izbrati začetni interval?
   (c) Izračunaj $\sqrt[5]{238.56}$ z Newtonovo metodo. Kaj bi bil primeren začetni približek?
   (d) *Dodatek* Primerjaj učinkovitost vseh treh metod.

2. Izračunaj vrednost odvodov iteracijskih funkcij (3.10) v bližini ničle $x \approx 1.369$. Katera od iteracijskih funkcij ima v bližini ničle odvod z najmanjšo absolutno vrednostjo?

3. Rešujemo enačbo
$$x^3 - 5 = 0.$$

(a) Kolikokrat moramo razpoloviti interval pri uporabi bisekcije, da dobimo $\sqrt[3]{5}$ z absolutno natančnostjo 10^{-9}, če je začetni interval $[0, 5]$?

(b) Izračunaj naslednje tri približke s sekantno metodo, če sta prva dva 0 in 5!

(c) Kako moramo izbrati parameter c, da bo iteracija $x_{n+1} = x_n + c(x_n^3 - 5)$ konvergirala proti $\sqrt[3]{5}$?

(d) Na kakšnem intervalu moramo izbrati začetno vrednost, da bo Newtonova iteracija zanesljivo konvergirala? (Namig: Newtonovo iteracijo zapiši v obliki metode fiksne točke.)

4. Dan je polinom

$$p(x) = x^8 - 170x^6 + 7392x^4 - 39712x^2 + 51200.$$

(a) Izračunaj ničle polinoma p.

(b) Opiši, kako bi največjo ničlo izračunal z metodo fiksne točke.

(c) Izračunaj po velikosti tretjo ničlo polinoma p_1, ki ga dobiš iz polinoma p tako, da koeficient pri x^2 spremeniš na -39710.

(d) Kolikšna je pri spremembi polinoma iz p v p_1 relativna sprememba druge ničle?

5. Poišči tisti koren enačbe

$$y = \operatorname{tg} x,$$

ki je najbližji 100.

(a) Kako določiš začetni interval za bisekcijo? Izračunaj koren na dve decimalni mesti.

(b) Izberi iteracijsko funkcijo za metodo fiksne točke. Izračunaj koren na tri decimalna mesta.

(c) Kako izbrati začetno točko za Newtonovo metodo? Izračunaj koren še z Newtonovo metodo (na štiri decimalna mesta).

6. Rešujemo enačbo

$$x - \sqrt{1 + x} = 0.$$

(a) Izračunaj vse njene ničle!

(b) Kakšen mora biti začetni interval za bisekcijo ali regulo falsi?

(c) Ali iteracija $x_{n+1} = \sqrt{1 + x_n}$ konvergira proti rešitvi, če izberemo x_0 dovolj blizu rešitve?

(d) Za katere vrednosti c metoda fiksne točke

$$x_{n+1} = x_n - c(x_n^2 - x_n - 1)$$

konvergira proti rešitvi, če je x_0 dovolj blizu rešitve?

7. Naj bo funkcija f definirana na intervalu $(0, \frac{\pi}{2})$ s predpisom

$$f(x) = \begin{cases} \frac{12x}{\pi} - 2 & \text{za} \quad \frac{\pi}{6} \leq x < \frac{\pi}{4} \\ \frac{-12x}{\pi} + 4 & \text{za} \quad \frac{\pi}{4} \leq x < \frac{\pi}{3} \\ 0 & \text{sicer.} \end{cases}$$

Rešujemo enačbo $f(x) = \sin x$.

(a) Skiciraj graf. Koliko ničel ima enačba?

(b) Zapiši algoritem za izračun najmanjše ničle z Newtonovo metodo. Izračunaj ničlo.

(c) Kako bi največjo ničlo izračunal z metodo fiksne točke? Zapiši algoritem in izračunaj ničlo.

8. Nelinearno enačbo $f(x) = 0$ rešujemo z metodo trisekcije. To je varianta metode bisekcije, pri kateri interval $[a, b]$, na katerem leži ničla, vsakič razdelimo na tri enake podintervale.

(a) Zapiši algoritem za metodo trisekcije za primer, ko je na intervalu $[a, b]$ natanko ena ničla funkcije f.

(b) Koliko je največje potrebno število korakov, če želimo izračunati približek, ki se od prave vrednosti razlikuje manj kot ε_x?

(c) Primerjaj bisekcijo in trisekcijo glede na potrebno število izračunov vrednosti funkcije f.

(d) Metodo trisekcije uporabi pri reševanju enačbe $2x - 3 = 0$ z začetnim intervalom $[-1.2, 10/3]$ in natančnostjo $\varepsilon_x = 0.1$.