Metode za reševanje nelinearnih enačb

Jure Korbar

Nelinearne enačbe so enačbe, ki lahko vsebujejo linearne člene npr. (x, $2x$, $\frac{x}{3}$) in konstante, poleg tega pa vsebujejo še nelinearne člene npr. (x^2, x^3, $\sin(x)$). Primer linearne enačbe je

$$\frac{6x}{7} = 13, \tag{1}$$

primer nelinearne enačbe pa je

$$3x^3 + 5x^2 = 4. \tag{2}$$

Če táko enačbo pretvorimo v funkcijo, je njen graf različen od premice. Vsako enačbo lahko pretvorimo v problem iskanja ničel funkcije tako, da vse člene enačbe prestavimo na eno stran in te člene zajamemo v funkciji $f(x)$. Tako pretvorjena enačba (1) nam da funkcijo

$$f(x) = \frac{6x}{7} - 13. \tag{3}$$

Sedaj rešujemo enačbo $f(x) = 0$. Rešitve te enačbe dobimo tako, da poiščemo *ničle* oz. *korene* funkcije $f(x)$.

Najbolj klasičen primer nelinearnih enačb so polinomi oz. iskanje ničel le-teh. Za iskanje ničel polinoma druge stopnje v srednji šoli spoznamo formulo za ničle kvadratne enačbe, za polinome višje stopnje pa nam predstavijo Hornerjev algoritem, ki pa je precej omejen, saj z gotovostjo najde le racionalne ničle. Kako pa bi se lotili reševanja spodnjih enačb?

$$\cos(x) = x \tag{4}$$

$$e^x = -x \tag{5}$$

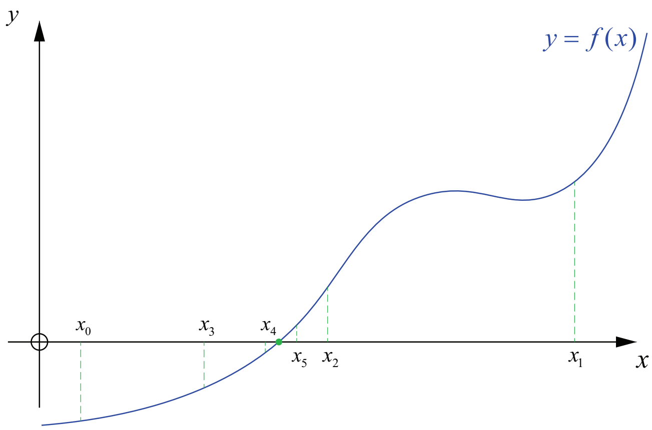
Omenjeni enačbi spadata v kategorijo transcendentnih enačb[1], zanje pa je značilno, da pogosto nimajo analitičnih rešitev, če v enačbi nastopa spremenljivka, tako v obliki transcendentne funkcije (leva stran enačb (4) in (5)) kot tudi v obliki algebrske funkcije (desna stran enačb (4) in (5)). Izjema tega pravila je npr. enačba $\sin(x) = x$, za katero vemo, da je rešitev enačbe $x = 0$, a si moramo priznati, da smo to rešitev uganili, saj vemo, da je vrednost obeh strani enačbe za primer $x = 0$ enaka 0.

Kako pa se potem v praksi lotimo reševanja enačb (4) in (5)? Odgovor se skriva v numeričnih metodah; vsako enačbo bomo pretvorili v problem iskanja ničel funkcije. Začeli bomo z oceno ničle in nato iterativno iskali boljši približek. Začetni približek bomo določili z izrisom grafa funkcije in ocenili, kje se ničla nahaja, nato pa bo izbrana metoda iskanja ničle določila boljši približek z vsako iteracijo te metode. V praksi ta približek velikokrat poznamo ali pa ga določimo s kakšno drugo vrsto grobe aproksimacije. Najprej bomo predstavili teoretično ozadje metode in ob enačbah in grafih pokazali njihovo delovanje, predstavili njihove prednosti in slabosti, za konec pa bomo enačbi (4) in (5) rešili s predstavljenimi metodami.

Bisekcijska metoda

Od vseh metod je najbolj preprosta in poznana bisekcijska metoda. Njeno delovanje je prikazano na sliki 1. Metoda temelji na dejstvu, da mora biti na intervalu, kjer zvezna funkcija spremeni predznak, vsaj ena ničla. Torej moramo pri tej metodi najprej definirati interval, na katerem se ničla pojavi (kjer funkcija spremeni predznak). Na sliki 1 je to interval $[x_0, x_1]$. Bisekcijska metoda ta interval razpolovi ($x_2 = \frac{x_0+x_1}{2}$) in tako dobimo dva podintervala. Ničla se nahaja na podintervalu, na katerem pride do spremembe predznaka (interval $[x_0, x_2]$). Tako dobimo nov, krajši interval, na katerem se nahaja ničla, in s tem je končana prva iteracija. V naslednji iteraciji

[1] Primer transcendentnih funkcij so: $\sin(x)$, $\cos(x)$, e^x, x^x …; transcendentne enačbe vsebujejo transcendentne funkcije, lahko pa vsebujejo tudi člene algebrskih funkcij ($x, x^2, x^{\frac{3}{2}}, \frac{x^3+5x}{3x^2-1}$ …).



SLIKA 1.
Potek delovanja bisekcijske metode: f je funkcija, katere ničlo iščemo, x_0 in x_1 sta meji začetnega intervala, na katerem se ničla pojavi, x_2, x_3, x_4 in x_5 pa so točke, ki jih dobimo s to metodo.

uporabimo interval $[x_0, x_2]$. Nadaljevanje metode da intervale $[x_3, x_2], [x_4, x_2], [x_4, x_5]$.

Ocena napake

Napaka, ki jo naredimo pri bisekcijski metodi (kot tudi pri ostalih metodah), je odvisna od števila iteracij in metode same. Največja začetna napaka je enaka velikosti začetnega intervala $\Delta x = x_1 - x_0$. Napako po n-ti iteraciji označimo z ε_n. Največja možna napaka po prvi iteraciji je enaka

$$\varepsilon_1 = \frac{\Delta x}{2},$$

po drugi iteraciji je napaka

$$\varepsilon_2 = \frac{\frac{\Delta x}{2}}{2} = \frac{\Delta x}{2^2},$$

po n-ti iteraciji pa je napaka

$$\varepsilon_n = \frac{\Delta x}{2^n}. \tag{6}$$

Ničlo moramo natančno izračunati, zato metodo po primernem številu iteracij ustavimo. Da dosežemo želeno natančnost ε, enačbo (6) obrnemo in logaritmiramo:

$$\begin{aligned} 2^n &= \frac{\Delta x}{\varepsilon}, \\ n \ln 2 &= \ln \frac{\Delta x}{\varepsilon} \\ n &= \log_2 \frac{\Delta x}{\varepsilon}, \end{aligned} \tag{7}$$

pri čemer n zaokrožimo na naslednje celo število.

Prednost bisekcijske metode je ta, da spada med zaprte metode. To pomeni, da vedno najdemo ničlo, ki je znotraj začetnega intervala, in je nemogoče, da bi se iskanje ničle prestavilo v področje izven začetnega intervala. Ima pa tudi določene slabosti:

- konvergira relativno počasi (druge metode potrebujejo nižje število iteracij za primerljivo natančnost rezultata);
- ničel sode stopnje[2] s to metodo ne moremo najti, saj funkcija v tem primeru ne spremeni predznaka;
- metoda ne razlikuje med ničlo in polom, saj je vezana le na spremembo predznaka funkcije, ne preverja pa vrednosti funkcije v okolici potencialne ničle;
- na intervalu, ki vsebuje več ničel, le-teh ne razločimo in zaznamo samo eno izmed njih.

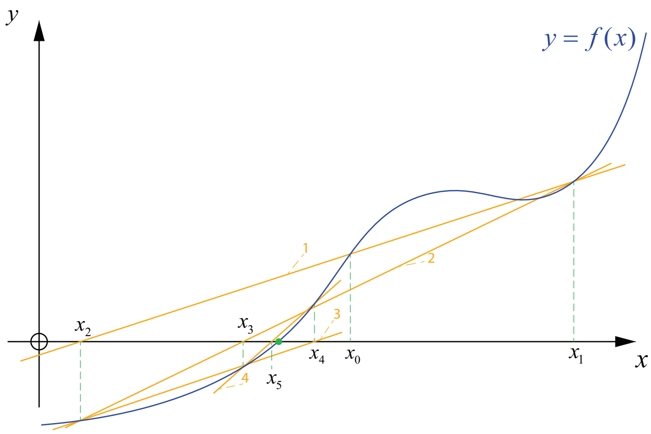
Sekantna metoda

Sekantna metoda spada med odprte metode iskanja ničel. Najprej pojasnimo njeno delovanje, nato pa bomo na primeru iz slike 2 videli, da s to metodo zares lahko najdemo ničlo, ki ne leži znotraj začetnega intervala.

Tako kot pri bisekcijski metodi si moramo tudi pri sekantni izbrati začetni interval $[x_0, x_1]$, za katerega predvidevamo, da v njem leži ničla (toda ni nujno, da to drži). Nato vrišemo premico skozi točki $f(x_0)$ in $f(x_1)$ ter pogledamo, kje je presečišče premice oz. sekante (po čemer se metoda imenuje) z absciso. To točko označimo z x_2. Interval za iskanje ničle v naslednji iteraciji je $[x_1, x_2]$.

Na sliki 2 vidimo, da se po štirih iteracijah bolje približamo dejanski ničli kot pri bisekcijski metodi,

[2]Enačba ima lahko več rešitev, ki so si medsebojno enake; če je število ponovitev neke rešitve sodo, imamo opravka s sodo ničlo (enačba $(x-6)^2 = 0$ ima eno dvojno rešitev: $x = 6$).



SLIKA 2.
Potek delovanja sekantne metode: f je funkcija, katere ničlo iščemo, x_0 in x_1 sta meji začetnega intervala, točke x_2, x_3, x_4 in x_5 pa so rezultat delovanja metode.

čeprav v začetnem intervalu $[x_0, x_1]$ ničla sploh ni bila vsebovana. Za boljšo predstavo so na sliki sekante oštevilčene glede na pripadajočo iteracijo.

Opisani postopek moramo sedaj prevesti v enačbe, da to metodo lahko dejansko uporabimo. Izbrali smo vrednosti x_0 in x_1 in poznamo vrednost funkcije v teh dveh točkah, zanima pa nas, kako določimo vrednost x_2. Pri tem si pomagamo s podobnimi trikotniki, kot je prikazano na sliki 3.

Vidimo, da sta trikotnika 1 in 2 podobna, torej je razmerje stranic za oba trikotnika enako. Iz te ugotovitve lahko zapišemo enačbo:

$$\frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_0)}{x_0 - x_2}.$$

Enačbo obrnemo in izpostavimo x_2:

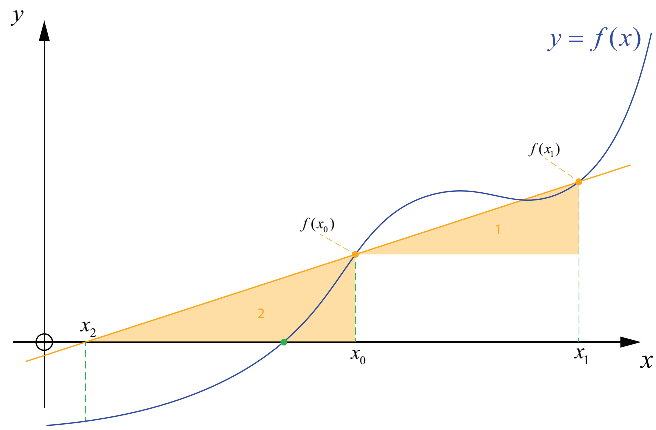
$$x_0 - x_2 = f(x_0) \cdot \frac{x_1 - x_0}{f(x_1) - f(x_0)},$$

$$x_2 = x_0 - f(x_0) \cdot \frac{x_1 - x_0}{f(x_1) - f(x_0)}. \tag{8}$$

V naslednji iteraciji v enačbi (8) za vrednost x_0 uporabimo vrednost x_1 iz prejšnje iteracije, za vrednost x_1 pa uporabimo x_2 iz prejšnje iteracije.

Red konvergence

V procesu primerjave sekantne metode z bisekcijsko se pojavi vprašanje, katera se hitreje približuje ničli.



SLIKA 3.
Izpeljava enačbe za sekantno metodo: $f(x_0)$ in $f(x_1)$ sta vrednosti funkcije f v krajiščih začetnega intervala $[x_0, x_1]$, x_2 pa je točka, ki jo dobimo po prvi iteraciji.

To hitrost narekuje red konvergence, ki nam pove, katera metoda bo potrebovala manj iteracij za izračun ničle z zadovoljivo natančnostjo. Red konvergence določimo tako, da primerjamo oceno napake iz prejšnje iteracije z oceno napake trenutne iteracije. To v splošnem zapišemo kot

$$\varepsilon_n = C \cdot \varepsilon_{n-1}^{k}, \tag{9}$$

kjer je C konstanta, k pa imenujemo red konvergence, ki bistveno vpliva na hitrost delovanja metode. Enačba (9) za bisekcijsko metodo je videti tako:

$$\varepsilon_n = \frac{1}{2}\varepsilon_{n-1}^{1}, \tag{10}$$

za sekantno metodo pa

$$\varepsilon_n = C \cdot \varepsilon_{n-1}^{1.618\ldots} \tag{11}$$

Red konvergence sekantnte metode je ravno zlati rez, a je takšen le pri dovolj lepih funkcijah. Ker je višji kot pri bisekcijski metodi, ta metoda v splošnem zahteva manjše število iteracij za želeno natančnost.

Druge prednosti sekantne metode so:

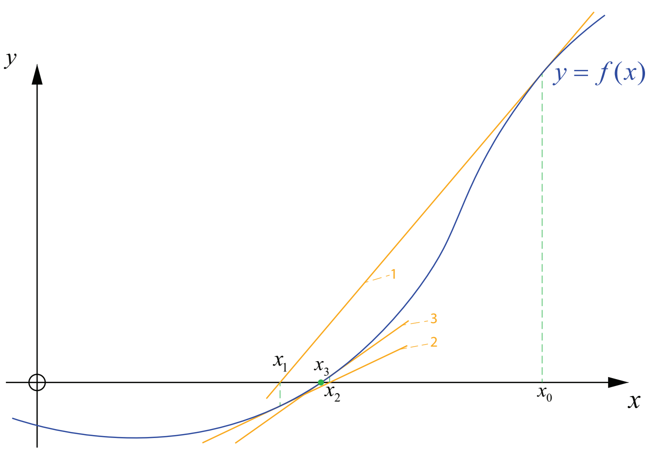
- metoda ne temelji na spremembi predznaka in uspe najti ničlo, tudi če je ta sode stopnje;
- preprosta implementacija; ne potrebujemo dodatnih informacij o funkciji, ki bi omejile uporabnost metode.

Tudi sekantna metoda ima določene slabosti:

- spada med odprte metode, zaradi česar lahko najde drugo ničlo od predvidene, odvisno od občutljivosti funkcije (zaradi lokalnih ekstremov lahko sekanta postane zelo položna, kar lahko popolnoma spremeni interval iskanja), lahko se tudi zacikla ali divergira v neskončnost;
- obstajajo metode, ki imajo še višji red konvergence in so zato hitrejše od sekantne metode.

Newton-Raphsonova metoda

To je ena izmed najbolj razširjenih metod (imenovana tudi Newtonova oz. tangentna metoda) za iskanje ničel. Tako kot sekantna metoda tudi ta spada med odprte metode. Za razliko od prej omenjenih metod si tu ne izberemo začetnega intervala ampak zgolj eno začeto točko. Metoda deluje tako, da pri izbrani vrednosti x na funkcijo postavi tangento in določi novi približek za ničlo tam, kjer ta tangenta seka absciso. To vrednost uporabimo tudi za naslednjo iteracijo. Potek delovanja je prikazan na sliki 4.



SLIKA 4.
Potek delovanja Newtonove metode: f je funkcija, katere ničlo iščemo, x_0 je začetni približek iskane ničle, x_1, x_2 in x_3 pa so izboljšani približki ničle dobljeni s to metodo.

Da pa lahko v dani točki na funkcijo postavimo tangento, moramo poznati točko, v kateri se tangenta dotika funkcije, prav tako pa moramo poznati njen naklon. Naklon tangente matematično popišemo z odvodom funkcije. Odvod funkcije f označimo kot f'. Vrednost $f'(x)$ odvoda funkcije f v točki x pove, kakšen je smerni koeficient tangente na funkcijo f v točki x. Tu se ne bomo ukvarjali z izračunom odvoda; vedeti moramo le, da odvod potrebujemo, če želimo uporabiti Newtonovo metodo, saj zanjo potrebujemo naklonske koeficiente tangent.

Na sliki 4 vidimo, da v tem primeru že po dveh iteracijah pridemo zelo blizu ničle, s tretjo iteracijo pa se približamo toliko, da na tej sliki ne razločimo razlike med ničlo in rezultatom tretje iteracije. Newtonova metoda ima namreč kvadratični red konvergence, zaradi česar je zelo priljubljena.

Ob sliki 5 izpeljimo še enačbo za Newtonovo metodo. Začetno oceno ničle kot ponavadi označimo z x_0, v točki $f(x_0)$ pa postavimo tangento z naklonom $f'(x_0)$. Naklon tangente v točki x_0 je definiran kot

$$f'(x_0) = \frac{\Delta y}{\Delta x} = \frac{f(x_0)}{x_0 - x_1}. \tag{12}$$

Iz enačbe (12) izpostavimo x_1 in dobimo enačbo za določitev ničle po Newton-Raphsonovi metodi:

$$x_0 - x_1 = \frac{f(x_0)}{f'(x_0)},$$
$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}. \tag{13}$$

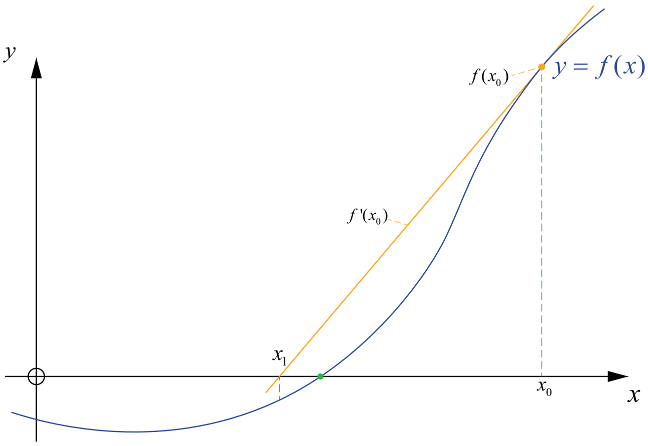
Prednost Newtonove metode je predvsem red konvergence. Metoda zato hitreje najde boljši približek ničle. Kot vse metode pa ima tudi ta svoje slabosti:

- metoda ima kvadratični red konvergence dejansko samo v bližnji okolici enostavne ničle (če je ničla višjega reda, je konvergenca nižja);
- spada med odprte metode in se, tako kot sekantna metoda, lahko oddalji od ničle, če naletimo na okolico lokalnega ekstrema;
- lahko divergira, če so začetni približki slabi.

Ostale metode

Poleg navedenih metod poznamo še druge, npr. Mullerjeva metoda, Ridderjeva metoda in metoda *regula falsi* (v dobesednem prevodu to pomeni *metoda napačnega pravila*).

Mullerjeva metoda je podobna sekantni, a namesto dveh začetnih točk izberemo tri, skoznje pa namesto premice teče parabola (s tremi točkami namreč lahko enolično določimo parabolo).



SLIKA 5.
Izpeljava enačbe za Newtonovo metodo: x_0 je začetni približek ničle, $f(x_0)$ je vrednost funkcije f v točki x_0, $f'(x_0)$ je vrednost smernega koeficienta tangente na funkcijo f v točki x_0, x_1 pa je rezultat prve iteracije metode.

Ridderjeva metoda deluje podobno kot bisekcijska metoda, le da po izračunu srednje vrednosti to vrednost uporabi v posebni formuli. S tem dobi boljši približek, s katerim prvotni interval razdelimo na dva dela, potem pa kot pri bisekcijski metodi pogledamo, v katerem intervalu pride do spremembe predznaka.

Metoda *regula falsi* pa je skupek sekantne in bisekcijske metode. Je zaprta metoda, torej mora biti ničla vedno znotraj začetnega intervala. Nato po postopku sekantne metode določimo nov približek ničle, interval za naslednjo iteracijo pa je tam, kjer funkcija spremeni predznak.

Rešitev uvodnih problemov

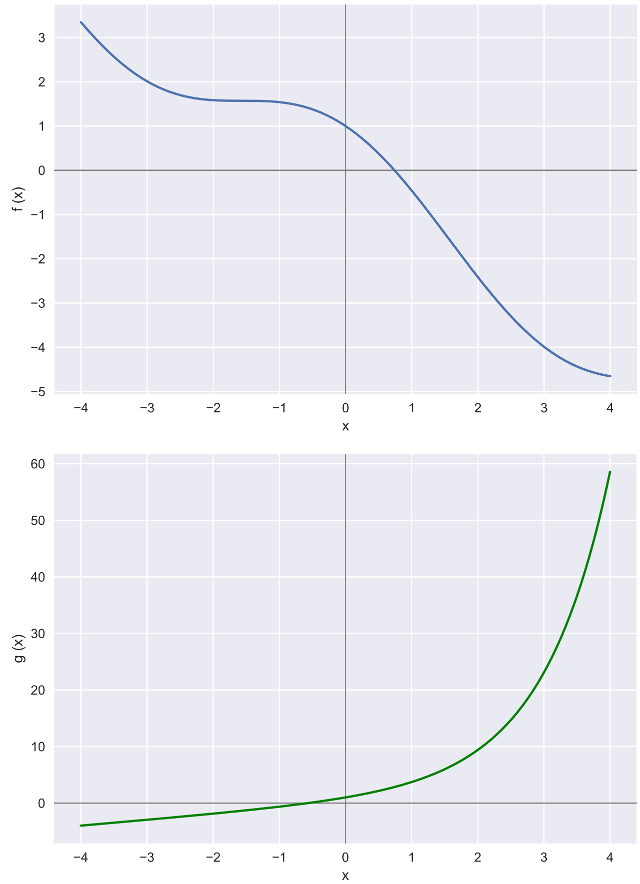
Lotimo se še problemov iz uvoda. Te bomo rešili s predstavljenimi metodami. Najprej pretvorimo enačbo (4) v funkcijo $f(x)$ in enačbo (5) v funkcijo $g(x)$:

$$f(x) = \cos(x) - x, \quad (14)$$

$$g(x) = e^x + x. \quad (15)$$

Sedaj iščemo ničle teh dveh funkcij. Pomagajmo si z grafi na sliki 6.

Iz grafa na sliki 6 zgoraj vidimo, da je ničla nekje na intervalu $0 < x < 1$. Ta podatek bomo uporabili



SLIKA 6.
Grafa funkcij $f(x)$ in $g(x)$

za začetni vrednosti pri bisekcijski metodi. Za začetno vrednost sekantne in Newtonove metode uporabimo vrednost $x = 0$. Sekantna metoda potrebuje dva začetna približka, tako kot metoda bisekcije. V tem primeru potrebujemo samo en začetni približek, saj ničlo iščemo s paketom *scipy* v okolju *python*, ta pa sam določi drugo vrednost intervala. Newtonova metoda potrebuje odvod funkcije $f(x)$, ki je

$$f'(x) = -\sin(x) - 1. \quad (16)$$

V tabeli 1 so prikazani rezultati iskanja ničel s temi tremi metodami na 14 decimalnih mest natančno.

Vidimo, da je Newtonova metoda konvergirala že pri četrti iteraciji, sekantna metoda je konvergirala pri sedmi, bisekcijska metoda pa po desetih iteracijah še ni uspela doseči primerljive natančnosti. Le-to

iteracija	bisekcija	sekantna	Newtonova
1	0,50000000000000	0,99995000250029	0,75036386784024
2	0,50000000000000	0,68508231592328	0,73911289091136
3	0,62500000000000	0,73629993134121	0,73908513338528
4	0,68750000000000	0,73911934458885	0,73908513321516
5	0,71875000000000	0,73908511214521	0,73908513321516
6	0,73437500000000	0,73908513321500	0,73908513321516
7	0,73437500000000	0,73908513321516	0,73908513321516
8	0,73828125000000	0,73908513321516	0,73908513321516
9	0,73828125000000	0,73908513321516	0,73908513321516
10	0,73828125000000	0,73908513321516	0,73908513321516

TABELA 1.
Prvih 10 iteracij iskanja ničle funkcije $f(x)$

je dosegla šele pri 44. iteraciji. S tem smo našli želeni približek za rešitev enačbe (4).

Rešimo še enačbo (5). Iz grafa na sliki 6 spodaj razberemo, da ničla leži nekje na intervalu $-1 < x < 0$. Spet bosta ti dve vrednosti predstavljali začetni interval za bisekcijsko metodo, za sekantno in Newtonovo metodo pa uporabimo za začetni približek vrednost $x = -1$. Odvod funkcije $g(x)$, ki ga potrebujemo za uporabo Newtonove metode je:

$$g'(x) = e^x + 1. \qquad (17)$$

Pokažimo še, da zgornji predpis g' res podaja smerni koeficient tangente na funkcijo $e^x + x$. Na sliki 7 je na funkcijo g postavljena tangenta v točki $x = 0$. Vrednost funkcije v tej točki je

$$g(0) = e^0 + 0 = 1,$$

vrednost odvoda funkcije pa je po enačbi (17) enaka

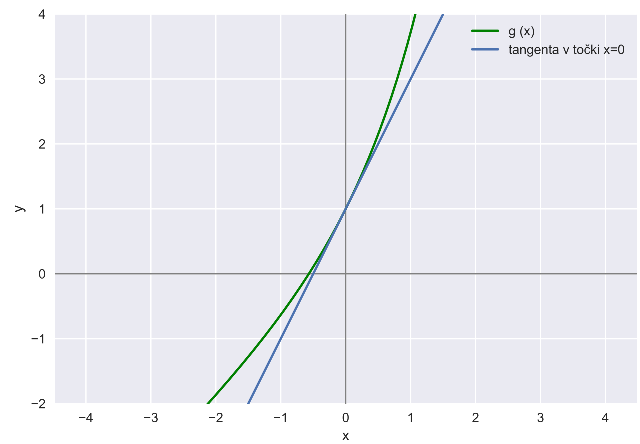
$$g'(0) = e^0 + 1 = 2.$$

Če v enačbo za premico $y = kx + n$ vstavimo vrednosti x, y in k lahko izračunamo vrednost n in tako dobimo enačbo za tangento.

$$\begin{aligned} y &= kx + n, \\ 1 &= 0 \cdot x + n, \\ n &= 1. \end{aligned}$$

Enačba tangente se tako glasi:

$$y = 2x + 1. \qquad (18)$$



SLIKA 7.
Graf funkcije $g(x)$ in tangenta na funkcijo g v točki $x = 0$

Iz grafa na sliki 7 vidimo, da enačba (18) zares drži.

Ponovimo postopek še za vrednost $x = 1$. Vrednost funkcije g je tu enaka

$$g(1) = e^1 + 1 = e + 1,$$

vrednost odvoda v tej točki pa je enaka

$$g'(1) = e^1 + 1 = e + 1.$$

Izračunajmo še začetno vrednost n tangente v tej točki:

$$\begin{aligned} y &= kx + n, \\ e + 1 &= (e + 1) \cdot 1 + n, \\ n &= 0. \end{aligned}$$

iteracija	bisekcija	sekantna	Newtonova
1	−1,00000000000000	−0,53787041498982	−0,53788284273999
2	−0,75000000000000	−0,56929601628683	−0,56698699140541
3	−0,62500000000000	−0,56715474022392	−0,56714328598912
4	−0,62500000000000	−0,56714328595119	−0,56714329040978
5	−0,59375000000000	−0,56714329040979	−0,56714329040978
6	−0,57812500000000	−0,56714329040978	−0,56714329040978
7	−0,57031250000000	−0,56714329040978	−0,56714329040978
8	−0,57031250000000	−0,56714329040978	−0,56714329040978
9	−0,56835937500000	−0,56714329040978	−0,56714329040978
10	−0,56738281250000	−0,56714329040978	−0,56714329040978

TABELA 2.
Prvih 10 iteracij iskanja ničle funkcije $g(x)$

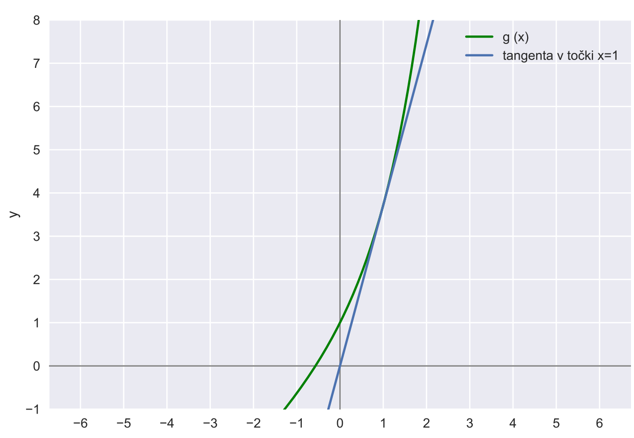
Enačba tangente v točki $x = 1$ je tako

$$y = (e + 1) \cdot x, \tag{19}$$

graf na sliki 8 pa potrjuje, da smo pravilno izračunali enačbo tangente.

Rezultati 10 iteracij teh metod za funkcijo $g(x)$ so predstavljeni v tabeli 2, prav tako na 14 decimalnih mest natančno.

Ponovno je najhitreje konvergirala Newtonova metoda, sekantna metoda je bila tudi zelo hitra, metoda bisekcije pa je za tako natančen rezultat potrebovala 48 iteracij.



SLIKA 8.
Graf funkcije $g(x)$ in tangenta na funkcijo g v točki $x = 1$

Poglejmo si še, kako v nekaj vrsticah v *python*-u pridemo do rezultata z modulom *scipy* na primeru funkcije $f(x)$:

```
import numpy as np
import scipy
from scipy.optimize import bisect, newton

def f(x):
    return np.cos(x) - x

def df(x):
    return -np.sin(x) - 1

print(bisect(f, 0, 1))
print(newton(f, 0))
print(newton(f, 0, df))
```

Na ta način torej rešujemo nelinearne enačbe, za katere nimamo eksplicitnih formul ali pa so te prezapletene, da bi jih splačalo uporabljali. Če modulu *newton* ne podamo odvoda funkcije, ta izvede sekantno metodo. S predstavljenimi metodami pa lahko iščemo tudi ničle polinomov, kar je zelo ugodno, saj za delovanje teh metod ni pomembno, kakšni so koeficienti polinoma (za Hornerjev algoritem so koeficienti ključnega pomena). Pri uporabi teh metod pa moramo biti pazljivi, saj ima vsaka določene slabosti, ki lahko vplivajo na rezultat. Proces iskanja ničel moramo zato prilagoditi glede na funkcijo in ničlo, ki jo iščemo.

× × ×