UNIVERZA NA PRIMORSKEM
Fakulteta za matematiko, naravoslovje in informacijske tehnologije

Matematika – 1. stopnja

Maruša Pajk

Numerično reševanje polinomskih enačb

Zaključna projektna naloga

Mentor: doc. dr. Vito Vitrih

Koper, 2011

Zahvala

Rada bi se zahvalila mentorju doc. dr. Vitu Vitrihu za sprejeto mentorstvo, pomoč, čas in nasvete pri izdelavi zaključne projektne naloge.

Posebna zahvala gre moji družini in fantu za vso izkazano podporo, potrpežljivost in spodbudo v času študija.

Povzetek

Ko matematičnega problema ne znamo rešiti analitično, ga rešujemo z numeričnimi metodami. Te so predstavljene z algoritmi, kateri so za uporabo v projektni nalogi sprogramirani v programu Octave. Eden izmed problemov, ki ga običajno rešujemo numerično, je reševanje nelinearnih enačb. Poznamo metode za reševanje poljubnih nelinearnih enačb in pa prirejene metode za reševanje polinomskih enačb. Za polinome pete ali višje stopnje namreč v splošnem ne obstaja eksaktna formula, s katero bi iz koeficientov polinoma izračunali njegove ničle, zato ničle izračunamo numerično. Zavedati pa se moramo, da so rešitve pri numeričnem reševanju problemov le približki za točne rezultate. S primerjanjem hitrosti konvergence numeričnih metod smo prišli do ugotovitev, da je za reševanje nelinearnih enačb v večini primerov najhitrejša tangentna metoda. Za reševanje polinomskih enačb pa je v primeru, ko potrebujemo samo eno ničlo naenkrat, najprimernejša Laguerrova metoda, če želimo vse ničle izračunati hkrati pa Aberth-Ehrlichova metoda.

Abstract

When mathematical problem cannot be solved analytically, we have to use numerical methods. These are presented with algorithms, which are programmed in Octave for the use in this thesis. One of the problems, which usually has to be solved using numerical methods, is how to find zeros of non-linear functions. There exist several methods for solving general non-linear equations and modified methods for solving polynomial equations. For polynomials of fifth or higher degree, there does not exist an exact formula for finding roots from the coefficients of the polynomial, therefore we have to calculate roots numerically. We must be aware, that the solutions obtained with numerical methods are only approximations to precise results. By comparing the speed of convergence of numerical methods we came to the conclusion, that for solving non-linear equations, in most cases, the fastest is Newton's method. When solving polynomial equations, where only one solution is needed, we prefer to use Laguerre's method, but when all roots are needed at the same time then the best choice is perhaps the Aberth-Ehrlich method.

Math. Subj. Class. (2011): 65H04, 65H05, 41A25

Ključne besede: numerične metode, algoritem, Octave, nelinearne enačbe, polinomske enačbe, konvergenca

Keywords: numerical methods, algorithm, Octave, non-linear equations, polynomial equations, convergence

Kazalo

Poglavje 1

Uvod

Ko rešujemo probleme iz znanosti, imamo pogosto opravka z zapletenimi izrazi, katere težko natančno izračunamo. Če imamo možnost računanja z ulomki, potem nimamo težav z zaokrožitvenimi napakami, toda velikokrat moramo računati z decimalnimi približki, kjer pa so zaokrožitvene napake neizogibne. V teh primerih uporabljamo numerične metode, s katerimi lahko rešujemo mnogo težje probleme, kot bi jih z analitičnimi metodami. Moramo pa se zavedati, da v tem primeru zaradi napak, ki so vedno prisotne pri numeričnem računanju, dobimo le približno rešitev problema. Zato je pomembno, da znamo realistično oceniti napako približka. Numerične metode predstavljamo z algoritmi. To so nedvoumni zapisi postopkov, ki nas pripeljejo do rešitve problema. Ko imamo dan problem, se moramo najprej odločiti za primerno numerično metodo, pri kateri moramo oceniti velikost morebitnih napak, določiti število iteracij, velikost koraka, sproti preverjati natančnost vmesnih rezultatov in intervenirati v primeru, ko so napake prevelike ali iteracije ne konvergirajo. Nato moramo izbrani algoritem sprogramirati, tako da ga razume računalnik. V zaključni nalogi so algoritmi sprogramirani v programu Octave, s pomočjo katerega so rešeni primeri v naslednjih poglavjih.

V matematiki uporabljamo numerične metode, ko do rešitve ne moremo priti analitično. Eden izmed znanih primerov je iskanje ničel polinoma pete stopnje. Abel-Ruffinijev izrek pravi, da za polinomske enačbe pete ali višje stopnje ne obstaja končno število operacij seštevanja, odštevanja, množenja, deljenja in korenjenja, s katerimi bi iz koeficientov polinoma izračunali ničle. Dokaz tega izreka si lahko ogledate v [7]. Rešitvi poljubne kvadratne enačbe $ax^2 + bx + c = 0$ lahko izrazimo s pomočjo analitične formule:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Podobne formule poznamo tudi za enačbe tretje in četrte stopnje, kjer se pojavljajo kubični in četrti koreni (glej [7]). Abel-Ruffinijev izrek pa nam pove, da obstajajo polinomske enačbe pete stopnje $ax^5 + bx^4 + cx^3 + dx^2 + ex + f = 0$, pri katerih nimamo druge možnosti, kot da ničle izračunamo numerično. Tak primer je enačba $x^5 - x + 1 = 0$. Podobno brez numerike težko rešimo večino transcendentnih enačb, kot na primer $x + \ln x = 0$, ki smo jo uporabili v primerih drugega poglavja, in tudi večino nelinearnih enačb in sistemov, ki jih srečamo pri številnih praktičnih problemih.

Zaključna projektna naloga je razdeljena na dve glavni poglavji. Prvo opisuje numerično reševanje poljubnih nelinearnih enačb in obsega pregled standardnih metod, kot so bisekcija, regula falsi, navadna iteracija, tangentna metoda, sekantna metoda, Mullerjeva metoda, inverzna interpolacija, kombinirane metode in metoda (f, f', f''). Drugo pa govori o metodah, ki so posebej prirejene za polinomske enačbe. Na začetku je navedenih nekaj izrekov, nato pa so opisane Bernoullijeva metoda, Graeffejeva metoda, Laguerrova metoda z redukcijo, Durand-Kernerjeva metoda in Aberth-Ehrlichova metoda. Kot glavna literatura je bila uporabljena knjiga [2], iz katere so tudi vzeti in prirejeni nekateri primeri.

Poglavje 2

Numerično reševanje poljubnih nelinearnih enačb

Pri numeričnem reševanju nelinearnih enačb iščemo rešitve enačbe

$$f(x) = 0, \tag{2.1}$$

kjer je f realna oziroma kompleksna zvezna funkcija (glej [5]). Enačba (2.1) ima eksaktno rešitev le v zelo posebnih primerih, zato pri reševanju navadno uporabimo numerične metode, s katerimi dobimo zadovoljiv približek točne rešitve enačbe. Recimo, da je α enostavna ničla funkcije f, ki je v točki α zvezno odvedljiva, in je odvod inverzne funkcije enak $1/f'(\alpha)$. Če imamo približek $\hat{\alpha}$ za α in hkrati velja $|f(\hat{\alpha})| \leq \epsilon$, potem ocenimo $|\hat{\alpha} - \alpha| \leq \epsilon/|f'(\alpha)|$. Natančno izračunane približke za ničlo torej dobimo, ko tangenta funkcije f v točki α ni preveč položna. Če je tangenta zelo položna, potem bomo v tem primeru težko izračunali natančen približek za ničlo (glej [5]). Taki ničli rečemo zelo občutljiva ničla. Če imamo večkratno ničlo, potem je $f'(\alpha) = 0$ in rečemo, da je ničla neskončno občutljiva. Večkratnih ničel torej ne moremo izračunati z isto natančnostjo kot enostavne ničle, ne glede na to, katero numerično metodo uporabimo. Enačba (2.1) ima lahko več rešitev, vendar pa nam večina numeričnih metod izračuna le eno od njih. Lahko se tudi zgodi, da enačba (2.1) nima nobene rešitve.

2.1 Bisekcija

Bisekcija je najbolj enostavna in zanesljiva metoda za reševanje enačbe (2.1). Recimo, da poznamo tak interval $[a, b]$, da za zvezno funkcijo $f : \mathbb{R} \longrightarrow \mathbb{R}$ velja $f(a) \cdot f(b) < 0$. Zaradi zveznosti ima funkcija f na intervalu (a, b) vsaj eno ničlo (glej [6]).

Če vzamemo $c = (a + b)/2$, potem imamo za $f(c)$ tri možnosti:

1.) $f(c) = 0$, kar pomeni, da je c že rešitev,

2.) $f(a) \cdot f(c) < 0$, kar pomeni, da za novi interval vzamemo $[a, c]$,

3.) $f(a) \cdot f(c) > 0$, kar pomeni, da za novi interval vzamemo $[c, b]$.

Na ta način v vsakem koraku nadaljujemo z razpolovljenim intervalom, ki vsebuje vsaj eno ničlo. Postopek končamo, ko je interval dovolj majhen in točko s sredine intervala vrnemo kot približek za ničlo funkcije f. Bisekcija ima dokaj počasno konvergenco (glej [5]).

Algoritem 2.1.1 (Bisekcija)**.** Podani imamo točki a in b, v katerih je f različno predznačena in toleranco ϵ.

while $|b - a| > \epsilon$
    $c = a + (b - a)/2$
    if $sign(f(a)) = sign(f(c))$
        $a = c;\ f(a) = f(c)$
    else
        $b = c;\ f(b) = f(c)$
    end
end

Točko c računamo po formuli $c = a + (b - a)/2$, ker ima ta izraz pri numeričnem računanju nekaj prednosti pred izrazom $c = (a + b)/2$ (glej [5, str. 36]). Metoda odpove, če imamo na iskanem intervalu samo sode ničle, ker ima v tem primeru funkcija na obeh straneh ničle enak predznak. Ravno tako z bisekcijo ne moremo iskati ničel kompleksnih funkcij. Če ima funkcija f na intervalu (a, b) več kot eno ničlo, potem bo bisekcija našla samo eno izmed njih.

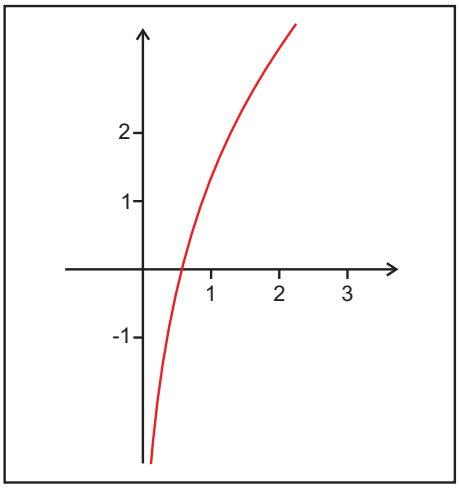
Primer 2.1.2. Z bisekcijo poiščimo ničle funkcije

$$f(x) = x + \ln x. \tag{2.2}$$

Ta funkcija je definirana za vsak pozitiven x in ima natanko eno pozitivno ničlo. To vemo, ker je odvod

$$f'(x) = 1 + \frac{1}{x},$$

kar implicira, da je funkcija (2.2) monotono naraščajoča za $x > 0$. Ker je $f(1) = 1$ ima edino ničlo na intervalu $(0, 1)$ (glej sliko 2.1). Ker imamo pri $x = 0$ vrednost $-\infty$, moramo za levo krajišče vzeti pozitivno število.



Slika 2.1: Graf funkcije (2.2).

Če vzamemo interval $[0.1, 1]$, potem s toleranco $\epsilon = \frac{1}{2}10^{-6}$ po 21-tih korakih bisekcije dobimo približek $\hat{\alpha}$ za ničlo na 6 decimalk natančno:

$$\hat{\alpha} = 0.567143.$$

S strojno natančnostjo izračunana ničla pa je

$$\hat{\alpha} = 0.567143290409784,$$

ki jo dobimo po 50-ih korakih, kar je prikazano v tabeli 2.1.

Tabela 2.1: Metoda bisekcije za iskanje ničle funkcije (2.2).

korak	$\hat{\alpha}$
1	0.550000000000000
2	0.775000000000000
3	0.662500000000000
4	0.606250000000000
5	0.578125000000000
6	0.564062500000000
⋮	⋮
48	0.567143290409787
49	0.567143290409785
50	0.567143290409784

2.2 Regula falsi

Metoda regula falsi je zelo stara metoda in je podobna metodi bisekcije, le da začetnega intervala ne razpolavljamo, ampak ga zožujemo. Vzemimo interval $[a, b]$, tako da je $f(a) \cdot f(b) < 0$. Sedaj potegnemo daljico skozi točki $(a, f(a))$ in $(b, f(b))$, ki seka x-os. Enačba te daljice je

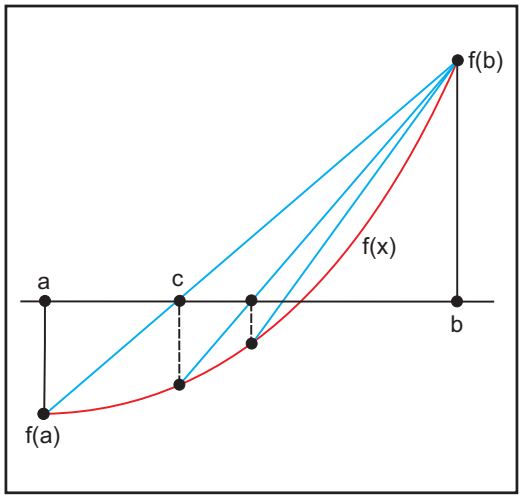
$$y - f(a) = \frac{f(b) - f(a)}{b - a}(x - a).$$

Točko c si izberimo na presečišču daljice z abscisno osjo:

$$c = a - f(a)\frac{b - a}{f(b) - f(a)}. \tag{2.3}$$

Če je $f(c) = 0$, je c ničla in postopek lahko zaključimo. Če je $f(a) \cdot f(c) > 0$, potem za krajišče a vzamemo c in postopek nadaljujemo na intervalu $[c, b]$, če pa je $f(a) \cdot f(c) < 0$, potem za novi interval vzamemo $[a, c]$. Postopek ponavljamo, dokler interval ni dovolj majhen. Končni rezultat je sredina zadnjega intervala.

Pričakovali bi, da se bosta pri izvajanju postopka obe krajišči intervala približevali ničli funkcije. Vendar se v praksi to navadno ne zgodi, saj se običajno spreminja le eno krajišče intervala, drugo pa ostaja nespremenjeno (glej [4]). Vseeno pa je metoda uporabna, saj približki, ki jih dobimo z metodo regula falsi, večinoma konvergirajo proti ničli funkcije f hitreje kot približki, ki jih dobimo z bisekcijo. Geometrijski pomen te metode je prikazan na sliki 2.2.



Slika 2.2: Metoda regula falsi.

Algoritem 2.2.1 (Regula falsi)**.** Podani imamo točki a in b, v katerih je f različno predznačena, in toleranco ϵ.

while $|b-a| > \epsilon$
    $c = a - f(a) * (b-a)/(f(b) - f(a))$
    if $sign(f(a)) = sign(f(c))$
        $a = c;\ f(a) = f(c)$
    else
        $b = c;\ f(b) = f(c)$
    end
end

Primer 2.2.2. Z metodo regula falsi poiščimo ničle funkcije (2.2) iz primera 2.1.2. Ko računamo s toleranco $\epsilon = \frac{1}{2}10^{-6}$, dobimo ničlo

$$\hat{\alpha} = 0.567143,$$

ki je na 6 decimalk natančna, po 17-ih korakih. S strojno natančnostjo izračunano ničlo

$$\hat{\alpha} = 0.567143290409784$$

pa smo dobili po 39-ih korakih.

V primerjavi z bisekcijo (primer 2.1.2) torej lahko vidimo, da približki, ki smo jih dobili z metodo regula falsi, v tem primeru hitreje konvergirajo k ničli funkcije (2.2).

2.3 Navadna iteracija

Pri metodi navadne iteracije nelinearni enačbi $f(x) = 0$ poiščemo ekvivalentno enačbo, tako da je

$$f(x) = 0 \Longleftrightarrow g(x) = x, \tag{2.4}$$

kjer je g iteracijska funkcija. To pomeni, da morata obe enačbi imeti iste rešitve, skupaj z večkratnostjo rešitev. Za funkcijo g torej velja, da ima negibne točke v ničlah funkcije f. Navedimo nekaj primerov, kako lahko zadostimo pogoju (2.4):

$$\begin{aligned} g(x) &= x - f(x), \\ g(x) &= x - Cf(x), \quad C \neq 0, \\ g(x) &= x - h(x)f(x), \quad h(x) \neq 0. \end{aligned}$$

Približke računamo tako, da si izberemo začetni približek x_0 in izvajamo iteracijo

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

Iteracijo izvajamo dokler še velja $|x_{r+1} - x_r| > \epsilon$. Da bodo približki res konvergirali proti rešitvi, moramo paziti pri izbiri iteracijske funkcije in začetnega približka. Tudi če izberemo začetni približek poljubno blizu negibne točke, to še ne zagotavlja, da bo zaporedje res konvergiralo proti tej točki. Naše zaporedje zagotovo konvergira k negibni točki, če na nekem intervalu $[a, b]$, ki vsebuje negibno točko, za vsak $x \in [a, b]$ velja $|g'(x)| < 1$ (glej [5]). O hitrosti konvergence v bližini negibne točke α odloča število $|g'(\alpha)|$. Če je $|g'(\alpha)| < 1$, potem je α privlačna točka in imamo zagotovljeno konvergenco v neki okolici okrog α, če pa je $|g'(\alpha)| > 1$, potem je α odbojna točka (glej [5]). Zelo pomembno je vedeti, kako hitro neka numerična metoda konvergira proti negibni točki, saj želimo s čim manj računanja priti do čim boljšega približka.

Definicija 2.3.1. Denimo, da zaporedje $(x_r)_{r\in\mathbb{N}}$ konvergira proti negibni točki α. Potem pravimo, da je red konvergence enak p, če obstajata taki konstanti $C_1, C_2 > 0$, da za dovolj pozne člene x_r velja

$$C_1|x_r - \alpha|^p \leq |x_{r+1} - \alpha| \leq C_2|x_r - \alpha|^p. \tag{2.5}$$

Red konvergence p pomeni, da spada napaka približka x_{r+1} v velikostni razred $|x_r - \alpha|^p$. V primeru navadne iteracije, lahko z odvajanjem ugotovimo red konvergence.

Lema 2.3.2. *Naj bo iteracijska funkcija g v okolici rešitve α enačbe $x = g(x)$ p-krat zvezno odvedljiva in naj velja $|g'(\alpha)| < 1$, $g^{(k)}(\alpha) = 0$ za $k = 1, \ldots, p-1$ ter $g^{(p)}(\alpha) \neq 0$. Potem ima metoda navadne iteracije $x_{r+1} = g(x_r), r = 0, 1, \ldots,$ v bližini rešitve α red konvergence p.*

Dokaz. Za dokaz leme uporabimo razvoj funkcije g v Taylorjevo vrsto okrog α in dobimo

$$x_{r+1} = g(x_r) = \alpha + \frac{1}{p!}(x_r - \alpha)^p g^{(p)}(\xi),$$

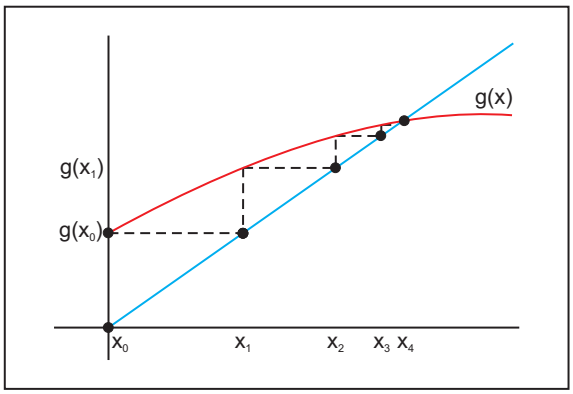
kjer je ξ točka med x_r in α. Če ocenimo absolutno vrednost p-tega odvoda funkcije g v okolici točke α navzdol in navzgor, dobimo konstanti C_1 in C_2 iz (2.5). □

Za nekatere konvergence imamo posebna imena. Ti primeri so:

- $p = 1$, ki ji pravimo linearna konvergenca, kjer na vsakem koraku dobimo konstantno število novih točnih decimalk,

- $p = 2$, ki ji pravimo kvadratična konvergenca, kjer na vsakem koraku dobimo podvojeno število novih točnih decimalk,
- $p = 3$, ki ji pravimo kubična konvergenca, kjer se na vsakem koraku število novih točnih decimalk potroji,
- $1 < p < 2$, ki ji pravimo superlinearna konvergenca in je hitrejša od linearne ter počasnejša od kvadratične.

Na sliki 2.3 lahko vidimo geometrijski pomen navadne iteracije.



Slika 2.3: Geometrijski pomen navadne iteracije.

Primer 2.3.3. Izračunajmo rešitev enačbe

$$x = -\ln x,$$

ki smo jo obravnavali že v primeru 2.1.2. Vemo, da ima eno samo ničlo, ki je blizu 0.5. Ker za iteracijsko funkcijo $g(x) = -\ln x$ velja $|g'(x)| = |-1/x| > 1$ za vsak $x \in (0, 1)$, zaporedje $x_{r+1} = -\ln x_r$, $r = 0, 1, \ldots$, za vsak $x_0 \neq \alpha$ divergira. Zato moramo enačbo zapisati v drugi obliki.

(i) Ekvivalenta oblika enačbe je

$$x = e^{-x}.$$

Njena iteracijska funkcija je $g(x) = e^{-x}$ in njen odvod je $g'(x) = -e^{-x}$, za katerega velja, da je $|g'(x)| < 1$ za vsak pozitiven x. Če vzamemo $x_0 = 0.5$ in natančnost $\epsilon = \frac{1}{2}10^{-6}$, dobimo po 24-ih korakih ničlo na 6 decimalk natančno.

Ker je to dokaj počasna konvergenca, nas zanima, če lahko enačbo še spremenimo, da bo zaporedje približkov hitreje konvergiralo.

(ii) Enačbi iz primera (i) na obeh straneh prištejemo ax in delimo z $1+a$, da dobimo enačbo

$$x = \frac{ax + e^{-x}}{1+a}.$$

Odvod njene iteracijske funkcije v točki α je

$$g'(\alpha) = \frac{a-\alpha}{1+a}.$$

Če vstavimo $a = x$ (glej [2, str. 68]), dobimo enačbo

$$x = \frac{x^2 + e^{-x}}{1+x},$$

katere iteracijska funkcija je

$$g(x) = \frac{x^2 + e^{-x}}{1+x}.$$

Njen odvod je

$$g'(x) = \frac{(x+2)(x-e^{-x})}{(1+x)^2}$$

in $|g'(\alpha)| = 0$, kar je najmanj, kar lahko dobimo. Torej je v tem primeru konvergenca hitrejša, kot v primeru (i). Za $x_0 = 0.5$ dobimo že v štirih korakih ničlo strojne natančnosti, kar je vidno v tabeli 2.2.

Tabela 2.2: Navadna iteracija za enačbo $f(x) = x + \ln x$.

korak k	x_k
1	0.571020439808422
2	0.567155568744114
3	0.567143290533261
4	0.567143290409784

Iz tega primera lahko sklepamo, da ima absolutna vrednost odvoda iteracijske funkcije velik vpliv na hitrost konvergence zaporedja približkov, in sicer manjša kot je, hitrejša je konvergenca.

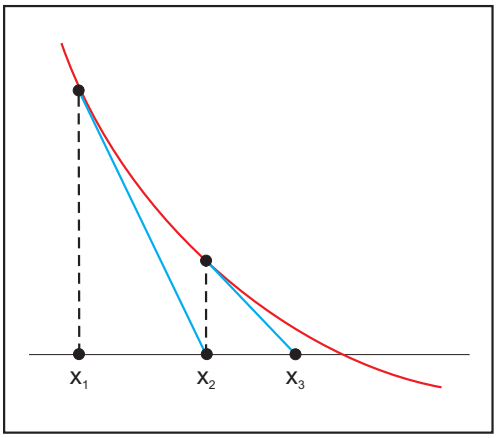
2.4 Tangentna metoda

Tangentna metoda, ki ji pravimo tudi Newtonova metoda, je poseben primer navadne iteracije in zato potrebuje samo en začetni približek. Je mnogo hitrejša od bisekcije, vendar ne vodi vedno do rešitve. Pri tej metodi funkcijo f aproksimiramo s tangento v točki $(x_r, f(x_r))$. Približek za ničlo funkcije f je presečišče te tangente z x-osjo.

Enačba tangente skozi točko $(x_r, f(x_r))$ je $y = f(x_r) + f'(x_r)(x - x_r)$. Iz tega sledi, da je presečišče tangente z x-osjo v točki $x_r - f(x_r)/f'(x_r)$, ki jo vzamemo kot naslednji približek x_{r+1}. Postopek ponavljamo za $r = 0, 1, ...$ in dobimo iteracijo

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

Pri tangentni metodi moramo na vsakem koraku iteracije izračunati eno vrednost funkcije in eno vrednost odvoda funkcije, torej je metoda tipa (f, f'). Geometrijski pomen tangentne metode je predstavljen na sliki 2.4.



Slika 2.4: Geometrijski pomen tangentne metode.

Kot že rečeno, je tangentna metoda pravzaprav posebna oblika navadne iteracije, kjer za iteracijsko funkcijo vzamemo

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

Ko to funkcijo odvajamo, dobimo

$$g'(x) = \frac{f(x)f''(x)}{f'^2(x)}.$$

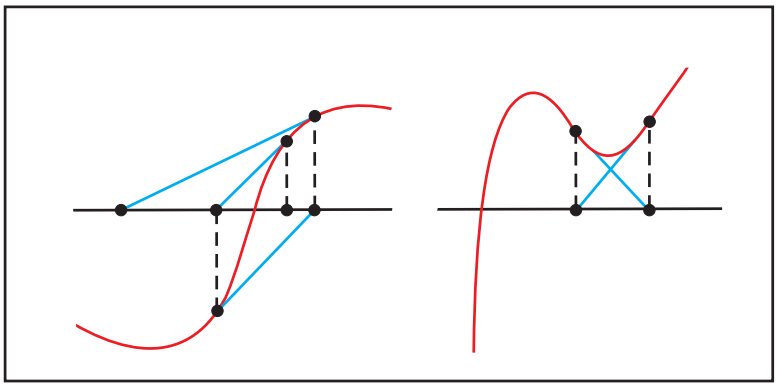
Če je α enostavna ničla, potem je

$$g'(\alpha) = 0,\ g''(\alpha) = \frac{f''(\alpha)}{f'(\alpha)}.$$

Če je $f''(\alpha) \neq 0$, potem je konvergenca tangentne metode kvadratična, sicer je višjega reda. Če je α m-kratna ničla, potem se lahko prepričamo, da je

$$g'(\alpha) = \lim_{x \to \alpha} g'(x) = 1 - \frac{1}{m},$$

iz česar sledi, da je $g'(\alpha) \neq 0$ za $m > 1$ in je konvergenca linearna (glej [2]). Lahko se nam tudi zgodi, da iteracija ne konvergira, v primeru, ko začetni približek ni dovolj blizu ničle ali, ko se zgodi zaciklanje približkov, kar lahko vidimo na sliki 2.5.



Slika 2.5: Divergenca tangentne metode, ko začetni približek ni dovolj blizu ničle (levo) in zaciklanje tangentne metode (desno).

Primer 2.4.1. Izračunajmo ničlo funkcije (2.2) s tangentno metodo. Če si za začetni približek izberemo $x_0 = 0.5$, dobimo rešitev že v 4-ih korakih, kar je predstavljeno v tabeli 2.3.

Tabela 2.3: Zaporedje približkov za enačbo $x + \ln x = 0$, dobljeno s tangentno metodo.

korak k	x_k
1	0.564382393519982
2	0.567138987715060
3	0.567143290399369
4	0.567143290409784

2.5 Tangentna metoda za kompleksne ničle

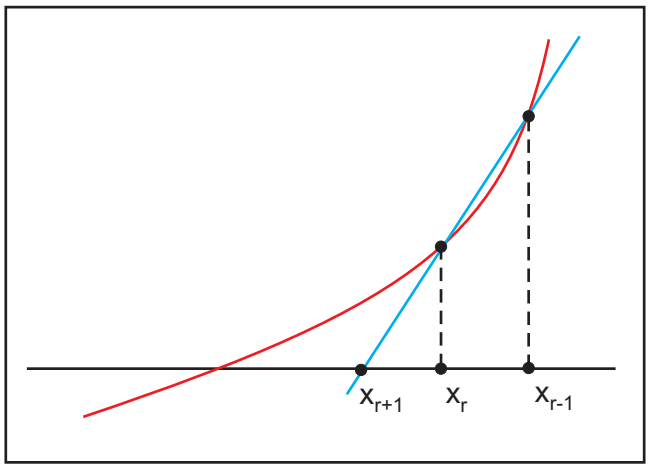
Tangentno metodo lahko uporabljamo tudi za računanje kompleksnih ničel funkcij. Če je funkcija f realna, potem moramo začeti s kompleksnim približkom, saj drugače ostanemo na realni osi. Naj bo f analitična funkcija z začetnim približkom z_0 in naj velja $f'(z_0) \neq 0$. Označimo z $w(z) = |f(z)|$ ploskev nad kompleksno ravnino. Potem je kompleksno število

$$z_1 = z_0 - \frac{f(z_0)}{f'(z_0)},$$

ki ga dobimo z enim korakom tangentne metode, presečišče dveh premic v kompleksni ravnini. Prva je presečišče tangentne ravnine na ploskev $w = |w(z)|$ v točki $(z_0, w(z_0))$ z ravnino $w = 0$, druga pa je projekcija gradienta na ploskev $w = |w(z)|$ v točki $(z_0, w(z_0))$ na ravnino $w = 0$. To pomeni, da potujemo iz točke $(z_0, w(z_0))$, ki je na ploskvi w, po tangentni ravnini v smeri najhitrejšega padanja funkcije do ravnine $w = 0$, kjer je naslednji približek za ničlo funkcije f po tangentni metodi. Podroben dokaz te zveze najdemo v [2].

2.6 Sekantna metoda

Če smo pri tangentni metodi funkcijo v bližini ničle aproksimirali s tangento, jo pri sekantni metodi aproksimiramo s sekanto skozi točki $(x_{r-1}, f(x_{r-1}))$ in $(x_r, f(x_r))$, kot lahko vidimo na sliki 2.6.



Slika 2.6: Geometrijski pomen sekantne metode.

V prvem koraku si izberemo dva začetna približka x_0 in x_1, v katerih je funkcija lahko enako predznačena. Za izračun naslednjih približkov uporabimo iteracijsko formulo:

$$x_{r+1} = x_r - \frac{f(x_r)(x_r - x_{r-1})}{f(x_r) - f(x_{r-1})}, \quad r = 1, 2, ...$$

Sekantna metoda ni navadna iteracija, saj je naslednji približek odvisen od zadnjih dveh približkov in ne le od zadnjega. Sekantna metoda je po obnašanju zelo podobna tangentni metodi, saj tudi tu velja, da bomo imeli konvergenco, če bomo imeli dva dovolj dobro izbrana približka. Konvergenca je pri tej metodi superlinearna (natančneje: red metode je 1,62 (glej [4, str. 49])), kar pomeni, da je malce počasnejša kot pri tangentni metodi. Vendar, ker pri sekantni metodi ne potrebujemo računati odvoda, je v primerih, ko je odvod zelo zapleten oziroma ga sploh ne moremo izračunati, lahko dober nadomestek za tangentno metodo.

Primer 2.6.1. Izračunajmo ničlo funkcije (2.2) še s sekantno metodo. Če za začetna približka vzamemo $x_0 = 0.5$ in $x_1 = 0.6$, potem za ničlo izračunano s strojno natančnostjo potrebujemo 5 korakov, kar vidimo v tabeli 2.4.

Tabela 2.4: Zaporedje približkov za enačbo $x + \ln x = 0$, dobljeno s sekantno metodo.

korak k	x_{k+1}
1	0.568413897526397
2	0.567120282313471
3	0.567143306843229
4	0.567143290409997
5	0.567143290409784

Konvergenca je hitra, ker imamo dobra začetna približka, vendar lahko vidimo, da približki dobljeni s tangentno metodo v primeru 2.4.1 hitreje konvergirajo proti ničli funkcije (2.2).

2.7 Mullerjeva metoda

Mullerjeva metoda je posplošitev sekantne metode. Pri tej metodi funkcijo aproksimiramo s parabolo, ki gre skozi tri točke $(x_r, f(x_r))$, $(x_{r-1}, f(x_{r-1}))$, $(x_{r-2}, f(x_{r-2}))$. Za naslednji približek x_{r+1} vzamemo ničlo te parabole, ki je

najbližje zadnjemu približku x_r. Mullerjeva metoda je primerna za računanje kompleksnih ničel, saj v primeru parabole s kompleksnimi ničlami, metoda sama iz začetnih realnih približkov preide v kompleksne približke in skonvergira do kompleksne ničle.

Recimo, da interpolacijski polinom skozi zadnje tri točke zapišemo kot

$$p(x) = a(x - x_r)^2 + b(x - x_r) + c,$$

potem lahko za njegove koeficiente izpeljemo naslednje formule (glej [5])

$$\begin{aligned} a &= \frac{(x_{r-1} - x_r)(f(x_{r-2}) - f(x_r)) - (x_{r-2} - x_r)(f(x_{r-1}) - f(x_r))}{(x_{r-2} - x_r)(x_{r-1} - x_r)(x_{r-2} - x_{r-1})}, \\ b &= \frac{(x_{r-2} - x_r)^2(f(x_{r-1}) - f(x_r)) - (x_{r-1} - x_r)^2(f(x_{r-2}) - f(x_r))}{(x_{r-2} - x_r)(x_{r-1} - x_r)(x_{r-2} - x_{r-1})}, \\ c &= f(x_r) \end{aligned}$$

in dobimo

$$x_{r+1} = x_r - \frac{2c}{b \pm \sqrt{b^2 - 4ac}}. \tag{2.6}$$

Če v enačbi (2.6) pred korenom izberemo tako, da bo imenovalec po absolutni vrednosti čim večji, potem dobimo rešitev bližjo x_r. Red konvergence Mullerjeve metode je 1.84, kar pomeni, da gre znova za superlinearno konvergenco (glej [5]).

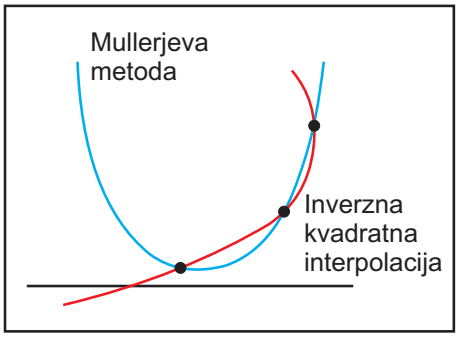
Primer 2.7.1. Rešimo enačbo $x + \ln x = 0$. Izbrati moramo tri začetne približke, ki naj bodo $x_0 = 0.4$, $x_1 = 0.5$, $x_2 = 0.6$. Rezultati so navedeni v tabeli 2.5:

Tabela 2.5: Zaporedje približkov za enačbo $x + \ln x = 0$, dobljeno z Mullerjevo metodo.

korak k	x_{k+2}
1	0.566810786015138
2	0.567142780548834
3	0.567143290406252
4	0.567143290409784

2.8 Inverzna interpolacija

Inverzna interpolacija deluje tako, da funkcijo f aproksimiramo s parabolo p, za katero velja $p(f(x_r)) = x_r$, $p(f(x_{r-1})) = x_{r-1}$ in $p(f(x_{r-2})) = x_{r-2}$. Za naslednji približek vzamemo $x_{r+1} = p(0)$. Red konvergence za to metodo je enak kot pri Mullerjevi metodi, in sicer $1,84$ (glej [5]). Od Mullerjeve metode se razlikuje po tem, da imamo pri inverzni interpolaciji vedno samo eno presečišče z x-osjo, ki ga vzamemo za naslednjo točko. Razliko med metodama vidimo na sliki 2.7.



Slika 2.7: Razlika med inverzno interpolacijo in Mullerjevo metodo.

V primeru realne funkcije nam inverzna kvadratna interpolacija iz realnih začetnih približkov vedno da realen približek in zato ni primerna za računanje kompleksnih ničel, tako kot je za to primerna Mullerjeva metoda. Večinoma se uporablja samo kvadratna interpolacija, saj pri uporabi več kot treh zadnjih točk ne pridobimo skoraj nič pri redu konvergence. Če vzamemo samo zadnji dve točki, potem dobimo sekantno metodo.

Primer 2.8.1. Ponovno si oglejmo reševanje enačbe $x + \ln x = 0$ in si, kot v primeru 2.7.1, izberimo tri začetne približke $x_0 = 0.4, x_1 = 0.5$, $x_2 = 0.6$. Rezultati so navedeni v tabeli 2.6:

Tabela 2.6: Reševanje enačbe $x + \ln x = 0$ z inverzno interpolacijo.

korak k	x_{k+2}	x_{k+1}	x_k
1	0.567140998310507	0.600000000000000	0.500000000000000
2	0.567143290282535	0.567140998310507	0.600000000000000
3	0.567143290409784	0.567143290282535	0.567140998310507
4	0.567143290409784	0.567143290409784	0.567143290282535

Lahko vidimo, da je konvergenca približkov dobljenih z inverzno interpolacijo res enaka kot z Mullerjevo metodo (primer 2.7.1).

2.9 Kombinirane metode - Brentova metoda

Glavni namen kombiniranih metod je, da pospešimo konvergenco in povečamo robustnost. Če imamo na primer začetni interval $[a, b]$, tak da je $f(a) \cdot f(b) < 0$, potem lahko bisekcijo kombiniramo s kakšno izmed hitrejših metod. Postopek je tak, da najprej izračunamo nov približek s hitrejšo metodo od bisekcije. Če je sedaj naš novi približek na intervalu $[a, b]$, ga obdržimo in zmanjšamo interval, sicer pa uporabimo bisekcijo. To nam zagotavlja, da bomo na koncu res dobili ničlo.

Ena izmed znanih kombiniranih metod je Brentova metoda, ki kombinira bisekcijo, sekantno metodo in inverzno kvadratno interpolacijo. Ta metoda ima superlinearno konvergenco, bisekcija pa zagotavlja, da konvergenca ni prepočasna in da ne pademo iz intervala $[a, b]$. Poglejmo si delovanje te metode. Imamo začetni interval $[a, b]$, da je $f(a) \cdot f(b) < 0$, in toleranci $\epsilon > 0$ in $\delta > 0$. Želimo, da velja $|f(b)| \leq |f(a)|$, zato a in b zamenjamo, če je potrebno, ter vzamemo $c := a$. Dokler je $|a - b| \geq \epsilon$, ponavljamo naslednji postopek:

1) Najprej izračunamo točko s, tako da, če je $c \neq a$, naredimo en korak inverzne kvadratne interpolacije, sicer pa en korak sekantne metode.

2) Če velja kateri izmed naslednjih pogojev

   - s ne leži med $(3a + b)/4$ in b,
   - v prejšnjem koraku smo uporabili bisekcijo in $|s - b| \geq |b - c|/2$ ali $|b - c| < \delta$,
   - v prejšnjem koraku nismo uporabili bisekcije in $|s - b| \geq |c - d|/2$ ali $|c - d| < \delta$,

   potem naredimo korak bisekcije in vzamemo $s := (a + b)/2$.

3) $d := c, c := b$, če je $f(a) \cdot f(s) < 0$, potem $b := s$, sicer $a := s$.

4) Če je $|f(a)| < |f(b)|$, zamenjamo a in b.

2.10 Metoda (f, f', f'')

Metodo (f, f', f'') izpeljemo z razvijanjem inverzne funkcije v Taylorjevo vrsto okoli funkcijske vrednosti $y = f(x)$. Če upoštevamo le prve tri člene, dobimo iteracijsko formulo

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)} - \frac{f''(x_r)f^2(x_r)}{2f'^3(x_r)}, \quad r = 0, 1, \ldots$$

Ta iteracija je pravzaprav izboljšana tangentna metoda, pri kateri poleg f in f' dodatno izračunamo še f'' za tekoči približek. Iteracijska funkcija je

$$g(x) = x - \frac{f(x)}{f'(x)} - \frac{f''(x)f^2(x)}{2f'^3(x)}$$

in njen odvod je

$$g'(x) = h(x)f^2(x),$$

kjer je

$$h(x) = \frac{3f''^2(x) - f'(x)f'''(x)}{2f'^4(x)}.$$

Vidimo, da je $g'(\alpha) = g''(\alpha) = 0$, $g'''(\alpha) = 2h(\alpha)f'^2(\alpha)$, kar nam da kubično konvergenco za enostavne ničle (glej [2]). Metoda je uporabna, ko nam računanje drugega odvoda funkcije ne povzroča preveč dodatnega dela.

Poglavje 3

Prirejene metode za reševanje polinomskih enačb

Metode, omenjene v prejšnjem poglavju, se lahko uporabijo tudi za iskanje ničel polinomov. Vendar se pri reševanju polinomskih enačb ponavadi zaplete, ker želimo izračunati tudi kompleksne ničle, čeprav so koeficienti polinoma realni, ali pa nas zanimajo vse ničle danega polinoma. Če polinom zapišemo v obliki

$$p(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n, \quad a_i \in \mathbb{R},\ x \in \mathbb{C},$$

kjer so a_k koeficienti polinoma in x spremenljivka, potem vemo, da je polinom p stopnje n, če velja $a_0 \neq 0$. Polinom stopnje n ima natanko n ničel v $\mathbb{C}$, če štejemo ničle z njihovimi večkratnostmi. Če ima polinom same realne koeficiente, potem vemo, da so vse razen k ničel (ki nastopajo v konjugiranih parih) realne ničle, pri čemer je $k \in \{0, 2, 4, \ldots, 2 \cdot \lfloor \frac{n}{2} \rfloor\}$. Pomembno pri polinomih je tudi to, da nam izračun vrednosti polinoma v neki točki omogoča Hornerjev algoritem, ki je linearne časovne zahtevnosti. Vrednost polinoma v točki x_0 izračunamo tako, da definiramo naslednje zaporedje konstant:

$$\begin{aligned} b_0 &= a_0 \\ b_1 &= a_1 + b_0x_0 \\ &\vdots \\ b_{n-1} &= a_{n-1} + b_{n-2}x_0 \\ b_n &= a_n + b_{n-1}x_0, \end{aligned}$$

kjer je $p(x_0) = b_n$. Da se prepričamo, da je to res, lahko polinom zapišemo kot

$$p(x) = a_n + x(a_{n-1} + x(a_{n-2} + \cdots + x(a_1 + a_0x)\cdots)). \tag{3.1}$$

Če sedaj iterativno vstavljamo b_i v enačbo (3.1), dobimo

$$\begin{aligned} p(x_0) &= a_n + x_0(a_{n-1} + x_0(a_{n-2} + \cdots + x_0(a_1 + b_0 x_0)\cdots)) \\ &= a_n + x_0(a_{n-1} + x_0(a_{n-2} + \cdots + x_0 b_1 \cdots)) \\ &\vdots \\ &= a_n + x_0 b_{n-1} \\ &= b_n. \end{aligned}$$

Problem iskanja ničel polinomov se pojavi pri polinomih pete ali višje stopnje. Kot je bilo povedano že v uvodu, nam Abel-Ruffinijev izrek pove, da obstajajo polinomske enačbe pete stopnje

$$ax^5 + bx^4 + cx^3 + dx^2 + ex + f = 0,$$

pri katerih nimamo druge možnosti, kot da ničle izračunamo numerično. V nadaljevanju bomo opisali posebne numerične metode, ki so prirejene iskanju ničel polinomov in konvergirajo hitro ter skoraj za vsak začetni približek. Te metode so Bernoullijeva, Graeffejeva, Laguerrova, Durand-Kernerjeva in Aberth-Ehrlichova metoda. Najprej pa si poglejmo dva izreka, ki nam pomagata pri določanju števila ničel polinoma na izbranem intervalu.

3.1 Descartesovo pravilo in Sturmovo zaporedje

Včasih je pomembno vedeti, koliko ničel ima polinom na nekem izbranem intervalu. Če je ta interval $(-\infty, 0)$ ali $(0, \infty)$, potem nam pri tem problemu pomaga Descartesovo pravilo, sicer pa Sturmovo zaporedje.

Descartesovo pravilo pravi, da je število pozitivnih ničel enačbe $p(x) = 0$ enako številu menjav predznakov v zaporedju neničelnih koeficientov polinoma $p(x)$ ali pa je od njega manjše za nek večkratnik števila 2. Število negativnih ničel pa dobimo tako, da prvotnemu polinomu zamenjamo predznak pri členih z liho potenco in nato preštejemo število menjav predznakov. Tudi tu velja, da je seštevek menjav lahko manjši za neko sodo število (glej [3]).

To, da je število lahko manjše za večkratnik števila 2, je posledica tega, da ima lahko polinom kompleksne ničle, ki vedno nastopajo v parih. Torej, če vemo, da ima polinom samo realne ničle, nam Descartesovo pravilo najde eksaktno število pozitivnih in negativnih ničel (glej [8]).

Primer 3.1.1. Vzemimo polinom

$$x^3 + x^2 - x - 1 = 0, \tag{3.2}$$

pri katerem se predznak v zaporedju neničelnih koeficientov $1, 1, -1, -1$ menja enkrat. Torej ima po Descartesovem pravilu ta polinom le eno pozitivno ničlo. Ta polinom lahko faktoriziramo in zapišemo kot

$$(x+1)^2(x-1), \tag{3.3}$$

kjer vidimo, da sta ničli 1, ki je enostavna, in -1, ki je dvakratna. Torej ima polinom res eno pozitivno ničlo. Sedaj zgornjemu polinomu zamenjamo predznake pri členih z liho potenco, saj na ta način dobimo število negativnih ničel polinoma (3.2). Dobimo polinom

$$-x^3 + x^2 + x - 1 = 0,$$

pri katerem se predznak v zaporedju koeficientov $-1, 1, 1, -1$ menja dvakrat, torej ima polinom (3.2) dve ali nobeno negativno ničlo. Vidimo, da se to sklada s faktorizacijo (3.3).

Poglejmo si sedaj še, kako si lahko pomagamo s Sturmovim zaporedjem polinomov. Naj bo p polinom s samimi enostavnimi ničlami. Sturmovo zaporedje je končno zaporedje polinomov $p_0, p_1, \ldots, p_m$, s padajočimi stopnjami, ki zadoščajo naslednjim lastnostim:

- $p_0 = p$ ima same enostavne ničle,
- če je $p(\xi) = 0$, potem velja $sign(p_1(\xi)) = sign(p'(\xi))$,
- če je $p_i(\xi) = 0$ za $0 < i < m$, potem velja $sign(p_{i-1}(\xi)) = -sign(p_{i+1}(\xi))$,
- p_m ne spremeni predznaka.

Poseben primer polinomov, ki tvorijo Sturmovo zaporedje je:

$$\begin{aligned}
p_0(x) &:= p(x), \\
p_1(x) &:= p'(x), \\
p_2(x) &:= -rem(p_0, p_1), \\
p_3(x) &:= -rem(p_1, p_2), \\
&\vdots \\
p_m(x) &:= -rem(p_{m-2}, p_{m-1}), \\
0 &:= -rem(p_{m-1}, p_m).
\end{aligned}$$

Ker je stopnja polinoma p_{i+1} manjša od stopnje p_i za $0 < i < m$, je zgornji algoritem končen. Zadnji polinom p_m predstavlja največji skupni delitelj polinoma p in njegovega odvoda. Ker ima p le enostavne ničle, nima nobene skupne ničle s svojim odvodom in zato je p_m neničelna konstanta.

Izrek 3.1.2 (Sturmov izrek). *Naj bo $p_0 = p, p_1, p_2, \ldots, p_m$ Sturmovo zaporedje in naj bo $\sigma(\xi)$ število zamenjav predznakov (ničel ne štejemo) v zaporedju*

$$p_0(\xi), p_1(\xi), p_2(\xi), \ldots, p_m(\xi).$$

Za realni števili $a < b$ velja, da je število različnih realnih ničel polinoma p na intervalu $(a, b]$ enako

$$\sigma(a) - \sigma(b).$$

Dokaz tega izreka najdemo v [9]. Podoben izrek obstaja tudi za polinome, ki nimajo samih enostavnih ničel (glej [9]).

Poglejmo si sedaj nekaj numeričnih metod, ki so prirejene za iskanje ničel polinomov.

3.2 Bernoullijeva metoda

Bernoullijevo metodo lahko uporabljamo za iskanje dominantnih ničel polinoma

$$p(x) = x^n - a_1 x^{n-1} - a_2 x^{n-2} - \ldots - a_n. \tag{3.4}$$

Zgornji polinom ima ničle $x_1, x_2, \ldots, x_n$, za katere velja

$$|x_1| > |x_2| \geq |x_3| \geq \ldots \geq |x_n|.$$

Dominantno ničlo x_1 izračunamo kot

$$x_1 = \lim_{k \to \infty} \frac{s_{k+1}}{s_k}, \tag{3.5}$$

pri čemer je s_k izračunan rekurzivno kot

$$s_k = a_1 s_{k-1} + a_2 s_{k-2} + \ldots + a_n s_{k-n}, \tag{3.6}$$

kjer za $k \leq n$ vstavimo $s_0 = k$ in $s_i = 0$ za $i < 0$. Števila s_k so v tesni povezavi z ničlami enačbe (3.4) in med njimi velja zveza

$$s_k = x_1^k + x_2^k + \ldots + x_n^k.$$

Formulam iz (3.6) pravimo *Newtonove formule* in povezujejo koeficiente polinoma z vsotami potenc ničel polinoma. Te formule izpeljemo na dva načina:

1.) Če je $k \geq n$, enačbo (3.4) pomnožimo z x^{k-n} in po vrsti vstavimo za x ničle $x_1, \ldots x_n$. Vse dobljene enačbe nato seštejemo in dobimo formulo (3.6).

2.) Če je $k < n$, potem vzamemo polinom (3.4), ki ima ničle $x_1, \ldots, x_n$. Njegov odvod je

$$p'(x) = nx^{n-1} - (n-1)a_1x^{n-2} - \cdots - a_{n-1}.$$

Če polinom zapišemo v ničelni obliki, dobimo

$$p(x) = (x - x_1) \cdots (x - x_n)$$

in njegov odvod je

$$p'(x) = \sum_{i=1}^{n} \frac{p(x)}{x - x_i}.$$

Če primerjamo koeficiente iz obeh odvodov pri potenci x^{n-k-1}, dobimo Newtonovo formulo (3.6):

$$-(n-k)a_k = s_k - a_1s_{k-1} - \cdots - a_{k-1}s_1 - na_k.$$

Vidimo, da nam Newtonove formule dajo pri danih koeficientih polinoma vsote potenc njegovih ničel. Iz (3.6) dobimo kvocient

$$\frac{s_{k+1}}{s_k} = \frac{x_1^{k+1} + x_2^{k+1} + \cdots + x_n^{k+1}}{x_1^k + x_2^k + \cdots + x_n^k},$$

ki ga lahko zapišemo tudi kot

$$\frac{s_{k+1}}{s_k} = x_1 \frac{1 + (\frac{x_2}{x_1})^{k+1} + \cdots + (\frac{x_n}{x_1})^{k+1}}{1 + (\frac{x_2}{x_1})^k + \cdots + (\frac{x_n}{x_1})^k}.$$

Kvocienti $|x_i/x_1|$ so manjši od 1 za vsak $i > 1$, iz česar sledi enačba (3.5). Konvergenca je odvisna od velikosti kvocienta $|x_2/x_1|$, saj manjši kot je, hitreje kvocienti sosednjih členov zaporedja s_k konvergirajo k x_1.

Primer 3.2.1. Vzemimo enačbo $x^3 - 6x^2 - 8x + 7 = 0$. Najprej jo preoblikujmo v pravo obliko

$$x^3 = 6x^2 + 8x - 7,$$

kjer dobimo $a_1 = 6, a_2 = 8, a_3 = -7$. V tabeli 3.1 so izračunane vrednosti s_k, ki smo jih dobili po formuli (3.6) in kvocienti, ki predstavljajo približke za dominantno ničlo.

Tabela 3.1: Tabela izračunov za primer 3.2.1.

k	s_k	s_k/s_{k-1}
1	6	
2	52	8.666667
3	339	6.519231
4	2408	7.103245
5	16796	6.975083
6	117667	7.005656
7	823514	6.998683
8	5764848	7.000304
9	40353531	6.999930
10	282475372	7.000016
11	1977326544	6.999996

Primer izračuna za $k = 3$: Vemo, da je $a_1 = 6, a_2 = 8, a_3 = -7, s_0 = 3$:

$$s_3 = a_1 s_2 + a_2 s_1 + a_3 s_0 = 6 \cdot 52 + 8 \cdot 6 - 7 \cdot 3 = 339.$$

Vidimo, da zaporedje konvergira k ničli $x = 7$, ki nam predstavlja dominantno ničlo. Ostali ničli sta še $\frac{1}{2}(-1-\sqrt{5}) \approx -1.61803$ in $\frac{1}{2}(\sqrt{5}-1) \approx 0.618034$. Hitrost konvergence je povezana z razmerjem $|x_2/x_1| = |-1.61803/7| = 0.231$.

3.3 Graeffejeva metoda

Graeffejevi metodi pravimo tudi *metoda kvadriranja korenov*, ki je bila zelo učinkovita metoda pred uporabo računalnikov. Kasneje se je izkazalo, da je zapletena za programiranje in se metoda ni obdržala. Posebnost te metode je, da nam da vse korene enačbe hkrati.

Vzemimo polinom stopnje n z realnimi koeficienti

$$p(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n, \tag{3.7}$$

ki ima ničle $x_1, x_2, \ldots, x_n$. Sedaj želimo sestaviti polinom z ničlami, za katere velja, da so po absolutnih vrednostih močno separirane:

$$|x_1| \gg |x_2| \gg \cdots \gg |x_n|.$$

Nato uporabimo *Vietove formule*, ki nam dajo naslednje zveze

$$\begin{aligned}
-\frac{a_1}{a_0} &= x_1 + x_2 + \cdots + x_n, \\
\frac{a_2}{a_0} &= x_1x_2 + x_1x_3 + \cdots + x_{n-1}x_n, \\
-\frac{a_3}{a_0} &= x_1x_2x_3 + x_1x_2x_4 + \cdots + x_{n-2}x_{n-1}x_n, \\
&\vdots \\
(-1)^n\frac{a_n}{a_0} &= x_1x_2\cdots x_n.
\end{aligned}$$

V formulah imamo na desni strani na prvem mestu dominantne člene. Če te člene obdržimo, ostale pa zanemarimo, potem dobimo

$$(-1)^k\frac{a_k}{a_0} \doteq x_1x_2\cdots x_k, \quad k = 1, \ldots, n.$$

Če sosednje enačbe iz zgornje zveze delimo, potem dobimo približke za vse korene

$$x_k \doteq -\frac{a_k}{a_{k-1}}, \quad k = 1, \ldots, n.$$

Glavni korak Graeffejeve metode je konstruiranje polinoma, ki ima za korene negativne kvadrate ničel polinoma (3.7). Polinom (3.7) lahko zapišemo tudi kot

$$p(x) = a_0(x - x_1)(x - x_2)\cdots(x - x_n). \tag{3.8}$$

Želimo dobiti polinom stopnje n, ki ima ničle $-x_1^2, -x_2^2, \ldots, -x_n^2$. Če polinom (3.8) pomnožimo s polinomom

$$p(-x) = (-1)^n a_0(x + x_1)(x + x_2)\cdots(x + x_n),$$

dobimo

$$p(x)p(-x) = (-1)^n a_0^2(x^2 - x_1^2)(x^2 - x_2^2)\cdots(x^2 - x_n^2).$$

Ker je zgornji polinom stopnje $2n$, uvedemo novo spremenjlivko $z := -x^2$ in dobimo polinom

$$P(z) = a_0^2(z + x_1^2)(z + x_2^2)\cdots(z + x_n^2),$$

ki je stopnje n in ima za ničle ravno negativne kvadrate ničel polinoma (3.7). Sedaj nas zanimajo le še koeficienti tega polinoma, ki jih dobimo s pomočjo zveze

$$P(z) = p(x)p(-x)$$

in nove spremenljivke $z = -x^2$. Zapišimo sedaj $P(z)$ v željeni obliki kot

$$P(z) = b_0 z^n + b_1 z^{n-1} + \cdots + b_{n-1}z + b_n.$$

Sledi, da je

$$p(x)p(-x) = (-1)^n(a_0^2 x^{2n} - (a_1^2 - 2a_0a_2)x^{2n-2} + \cdots + (-1)^n a_n^2). \qquad (3.9)$$

Če v enačbi (3.9) namesto x^2 pišemo $-z$, dobimo:

$$\begin{aligned}
b_0 &= a_0^2, \\
b_1 &= a_1^2 - 2a_0a_2, \\
b_2 &= a_2^2 - 2a_1a_3 + 2a_0a_4, \\
b_3 &= a_3^2 - 2a_2a_4 + 2a_1a_5 - 2a_0a_6, \qquad (3.10) \\
&\vdots \\
b_{n-2} &= a_{n-2}^2 - 2a_{n-3}a_{n-1} + 2a_{n-4}a_n, \\
b_{n-1} &= a_{n-1}^2 - 2a_{n-2}a_n, \\
b_n &= a_n^2.
\end{aligned}$$

Iz zvez (3.10) lahko razberemo splošno formulo

$$b_k = a_k^2 + 2\sum_{i=1}^{m}(-1)^i a_{k-i}a_{k+i}, \quad k = 0, 1, \ldots, n,$$

kjer je $m := min(k, n-k)$. Postopek kvadriranja korenov lahko večkrat ponovimo. Če naredimo r ponovitev, dobimo polinom

$$P(z) = A_0 z^n + A_1 z^{n-1} + \cdots + A_{n-1}z + A_n$$

z ničlami $-x_1^p, -x_2^p, \ldots, -x_n^p$, kjer je $p = 2^r$. Pri dovolj velikem r-ju postanejo ničle polinoma (3.7), za katere velja $|x_1| > |x_2| > \cdots > |x_n| \geq 0$, močno separirane. Te ničle lahko izračunamo po formuli od prej:

$$x_k^p \doteq \frac{A_k}{A_{k-1}}, \quad k = 1, \ldots, n.$$

Iz teh ničel pa potem dobimo približke za absolutne vrednosti ničel polinoma (3.7), tako da korenimo

$$|x_k| \doteq \sqrt[p]{\frac{A_k}{A_{k-1}}}, \quad k = 1, \ldots, n.$$

Na koncu moramo le še pravilno določiti predznak, kar ni preveč težka naloga (glej [2, str. 131]) in dobimo ničle. Proces kvadriranja prekinemo, ko so koeficienti naslednjega polinoma enaki kvadratom koeficientov prejšnjega.

Primer 3.3.1. Vzemimo enačbo iz primera 3.2.1:

$$x^3 - 6x^2 - 8x + 7 = 0.$$

V tabeli 3.2 vidimo 5 ponovitev procesa kvadriranja korenov. Koeficienti A_i pomenijo koeficiente naslednje enačbe, ki ima za ničle negativne kvadrate ničel prejšnje enačbe s koeficienti a_i:

Tabela 3.2: Tabela izračunov za primer 3.3.1.

r	$A_0 = a_0^2$	$A_1 = a_1^2 - 2a_0a_2$	$A_2 = a_2^2 - 2a_1a_3$	$A_3 = a_3^2$
0	1	-6	-8	7
1	1	52	148	49
2	1	2408	16808	2401
3	1	5764848	270945648	5764801
4	1	$3.323293057 \cdot 10^{13}$	$7.334507777 \cdot 10^{16}$	$3.323293057 \cdot 10^{13}$
5	1	$1.104427674 \cdot 10^{27}$	$5.379498224 \cdot 10^{33}$	$1.104427674 \cdot 10^{27}$

Primer izračuna za $r = 3$: Prejšnja enačba ima koeficiente $a_0 = 1, a_1 = 2408, a_2 = 16808, a_3 = 2401$. Torej izračunamo koeficiente naslednje enačbe na ta način:

$$\begin{aligned} A_0 &= a_0^2 = 1, \\ A_1 &= a_1^2 - 2a_0a_2 = 2408^2 - 2 \cdot 1 \cdot 16808 = 5764848, \\ A_2 &= a_2^2 - 2a_1a_3 = 16808^2 - 2 \cdot 2408 \cdot 2401 = 270945648, \\ A_3 &= a_3^2 = 2401^2 = 5764801. \end{aligned}$$

Sedaj izračunamo absolutne vrednosti treh realnih ničel s pomočjo dvaintridesetega korena kvocienta sosednjih koeficientov iz zadnje vrstice tabele 3.2.

Dobimo

$$\begin{aligned} |x_1| &\doteq 7 \\ |x_2| &\doteq 1.618033989, \\ |x_3| &\doteq 0.618034046. \end{aligned}$$

Vemo, da je vsota ničel enaka 6 in produkt -7. Določiti moramo še predznake ničlam in edina možnost je, da je druga ničla negativna, ostali dve pa pozitivni.

3.4 Laguerrova metoda

Recimo, da imamo polinom $p(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n$, z ničlami $x_1, \ldots, x_n$. Ta polinom lahko zapišemo kot

$$p(x) = a_0(x - x_1)\cdots(x - x_n).$$

Definirajmo funkcijo

$$S_1(x) := \frac{p'(x)}{p(x)} = \sum_{i=1}^{n} \frac{1}{x - x_i}$$

in njen odvod z nasprotnim predznakom

$$S_2(x) := -S_1'(x) = \frac{p'^2(x) - p(x)p''(x)}{p^2(x)} = \sum_{i=1}^{n} \frac{1}{(x - x_i)^2}.$$

Če želimo izračunati ničlo x_n, potem označimo izbrano obratno vrednost faktorja v $p(x)$ kot

$$a(x) = \frac{1}{x - x_n}.$$

Nato označimo še aritmetično sredino preostalih obratnih vrednosti faktorjev v $p(x)$ kot

$$b(x) = \frac{1}{n-1} \sum_{i=1}^{n-1} \frac{1}{x - x_i},$$

odmik i-te obratne vredosti faktorja od aritmetične sredine z

$$d_i(x) = \frac{1}{x - x_i} - b(x)$$

in vsoto kvadratov odmikov kot

$$d(x) = \sum_{i=1}^{n-1} d_i^2(x).$$

Za lažje razumevanje bomo argument x izpustili in izrazili S_1 in S_2 le z a, b in d. Dobimo

$$\begin{aligned} S_1 &= a + (n-1)b, \\ S_2 &= a^2 + \sum_{i=1}^{n-1}(b+d_i)^2 = a^2 + (n-1)b^2 + 2b\sum_{i=1}^{n-1} d_i + d = \\ &= a^2 + (n-1)b^2 + d. \end{aligned}$$

Iz teh dveh enačb eliminiramo b in izrazimo a:

$$a = \frac{1}{n}\left[S_1 \pm \sqrt{(n-1)(nS_2 - S_1^2 - nd)}\right].$$

Sedaj lahko iz te enačbe dobimo x_n:

$$x_n = x - \frac{n}{S_1 \pm \sqrt{(n-1)(nS_2 - S_1^2 - nd)}}.$$

S_1 in S_2 lahko izračunamo pri poljubnem x, količine d pa ne moremo izračunati. Če člen nd zanemarimo, potem dobimo približek za x_n. Bližje kot je x ničli x_n, boljši je približek, zato uporabljamo formulo iterativno in dobimo Laguerrovo metodo:

$$\begin{aligned} S_1 &= \frac{p'(x_n^{(r)})}{p(x_n^{(r)})}, \\ S_2 &= \frac{p'^2(x_n^{(r)}) - p(x_n^{(r)})p''(x_n^{(r)})}{p^2(x_n^{(r)})}, \\ x_n^{(r+1)} &= x_n^{(r)} - \frac{n}{S_1 \pm \sqrt{(n-1)(nS_2 - S_1^2)}}, \end{aligned}$$

oziroma

$$x_n^{(r+1)} = x_n^{(r)} - \frac{np(x_n^{(r)})}{p'(x_n^{(r)}) \pm \sqrt{(n-1)((n-1)p'^2(x_n^{(r)}) - np(x_n^{(r)})p''(x_n^{(r)}))}}.$$

Za numerično stabilnost si izberemo tisti predznak, ki da imenovalcu večjo absolutno vrednost. Laguerrova metoda za poljuben realni začetni približek skonvergira k eni ničli. V primeru enostavne ničle ima ta metoda kubično konvergenco. Dokaz tega dejstva najdemo v [2, str. 134].

Primer 3.4.1. Vzemimo polinom pete stopnje

$$x^5 - 4x^4 + 6x^3 - 3x^2 + 2x + 2 = 0, \tag{3.11}$$

ki ima eno realno ničlo in dva para konjugirano kompleksnih ničel. Če za reševanje uporabimo tangentno metodo, dobimo z začetnim približkom $x_0 = 2 + i$ približke zapisane v tabeli 3.3.

Tabela 3.3: Tangentna metoda za enačbo (3.11) z $x_0 = 2 + i$.

korak k	x_k
1	1.947535771065183 +1.020667726550079i
2	1.947119286434461 +1.025717556555235i
3	1.947153442999702 +1.025698136346046i
4	1.947153443329095 +1.025698138695321i

Če si za začetni približek izberemo $x_0 = i$, potem dobimo približek za kompleksno ničlo $0.265518544073020 + 0.948845986366118i$ po šestih korakih.

Tabela 3.4: Tangentna metoda za enačbo (3.11) z $x_0 = i$.

korak k	x_k
1	0.185520361990950 +0.895927601809955i
2	0.276550432675542 +0.938377891370783i
3	0.265165792243013 +0.948901772136241i
4	0.265518444176035 +0.948845820071227i
5	0.265518544073075 +0.948845986366133i
6	0.265518544073020 +0.948845986366118i

Sedaj bomo enačbo (3.11) reševali z Laguerrovo metodo. Če vzamemo realni začetni približek $x_0 = -1$, dobimo ničlo izračunano s strojno natančnostjo v štirih korakih, kar je prikazano v tabeli 3.5.

Tabela 3.5: Laguerrova metoda za enačbo (3.11) z $x_0 = -1$.

korak k	x_k
1	-0.388161082236077
2	-0.425370663388826
3	-0.425343974804221
4	-0.425343974804230

Če si izberemo kompleksni začetni približek $x_0 = 2 + i$, dobimo najbližjo kompleksno ničlo v treh korakih, kot vidimo v tabeli 3.6.

Tabela 3.6: Laguerrova metoda za enačbo (3.11) z $x_0 = 2 + i$.

korak k	x_k
1	1.947182605248842 +1.025699801588844i
2	1.947153443329096 +1.025698138695325i
3	1.947153443329095 +1.025698138695322i

Za začetni približek $x_0 = i$, pa dobimo v treh korakih približek za kompleksno ničlo $0.265518544073020 + 0.948845986366118i$.

Tabela 3.7: Laguerrova metoda za enačbo (3.11) z $x_0 = i$.

korak k	x_k
1	0.268811977294902 +0.956500572950411i
2	0.265518367611264 +0.948846175228808i
3	0.265518544073020 +0.948845986366118i

V tem primeru vidimo, da je konvergenca Laguerrove metode hitrejša od konvergence tangentne metode.

3.4.1 Redukcija

Ko smo že izračunali približek za eno ničlo α polinoma

$$p(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n, \quad a_i \in \mathbb{R}, \quad a_0 \neq 0,$$

in želimo izračunati še ostale ničle, ne da bi z drugim približkom zopet dobili isto ničlo, uporabimo redukcijo. To je postopek izločevanja že izračunanih korenov. Opisali bomo *eksplicitno redukcijo*, kjer polinom $p(x)$ delimo z $x-\alpha$ (α je že izračunani prvi približek za realno ničlo) in ostanek zanemarimo:

$$p(x) = (x - \alpha)q(x) + A.$$

Sedaj nadaljujeno z iskanjem ničel polinoma q, ki je stopnje $n - 1$. Če je α približek za kompleksno ničlo, potem $p(x)$ delimo s polinomom $d(x) = (x - \alpha)(x - \bar{\alpha}) = x^2 + fx + g$ in ostanek zanemarimo

$$p(x) = d(x)q(x) + Ax + B.$$

Iskanje ostalih ničel nadaljujemo na polinomu q stopnje $n - 2$.

Direktna redukcija

Pri tej metodi delimo polinom po padajočih potencah spremenljivke x. Polinom zapišemo kot

$$p(x) = (x - \alpha)(b_0x^{n-1} + \cdots + b_{n-1}) + b_n, \tag{3.12}$$

kjer nam koeficiente $b_0, \ldots, b_n$ in ostanek da Hornerjev algoritem:

$$\begin{aligned}
&b_0 = a_0\\
&r = 1, \ldots, n-1\\
&\quad b_r = \alpha b_{r-1} + a_r\\
&b_n = p(\alpha)\\
&q(x) = b_0x^{n-1} + \cdots + b_{n-1}.
\end{aligned}$$

Naj bo β poljubna ničla polinoma $q(x) = b_0x^{n-1}+\cdots+b_{n-1}$, potem iz enačbe (3.12) sledi, da je β eksaktna ničla enačbe

$$p(x) - p(\alpha) = 0,$$

ki se od prvotne enačbe razlikuje le v prostem členu b_n. Če je $|\alpha|$ majhno in natančno izračunano število, potem je $|f(\alpha)|$ majhno število in vse nadaljne ničle so točne ničle malo spremenjene prvotne enačbe. V primeru, ko je α po absolutni vrednosti veliko število, pričakujemo veliko vrednost $|f(\alpha)|$, torej nadaljne ničle niso nujno tako točne kot prej. To pomeni, da bo direktna redukcija stabilna, če bomo ničle izločali po naraščajoči absolutni vrednosti.

Obratna redukcija

Tu delimo polinom po naraščajočih potencah spremenljivke x. Polinom zapišemo kot

$$p(x) = (x - \alpha)(c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}) + c_n x^n, \tag{3.13}$$

kjer koeficiente c_i dobimo po naslednjem algoritmu:

$$\begin{aligned} &c_0 = -a_n/\alpha \\ &r = 1, \ldots, n-1 \\ &\quad c_r = (c_{r-1} - a_{n-r})/\alpha \\ &c_n = a_0 - c_{n-1}. \end{aligned}$$

Iz enačbe (3.13) dobimo

$$c_n = \frac{p(\alpha)}{\alpha^n}.$$

Če je β ničla polinoma $q(x) = c_{n-1}x^{n-1} + \cdots + c_0$, potem iz (3.13) sledi, da je β eksaktna ničla polinoma

$$p(x) - \frac{p(\alpha)}{\alpha^n}x^n = 0.$$

Če je $|\alpha|$ velik, bomo imeli majhne motnje polinoma, če pa je $|\alpha|$ majhen, bodo motnje velike. Torej lahko sklepamo, da bo obratna redukcija numerično stabilna, če bomo ničle izločali po padajoči absolutni vrednosti.

Kombinirana redukcija

Pogosto ne vemo, za katero ničlo po velikosti je α približek, zato si pomagamo s kombinirano redukcijo. V tem primeru delimo polinom $p(x)$ z $(x - \alpha)$ z obeh strani (3.12) in (3.13). Izberemo si polinom

$$q(x) = b_0 x^{n-1} + \cdots + b_{n-r-1}x^r + c_{r-1}x^{r-1} + \cdots + c_0, \quad 0 \le r \le n,$$

in na podlagi algoritmov prejšnjih dveh metod vidimo, da je

$$(x - \alpha)q(x) = p(x) - Ax^r.$$

Če v zgornjo enačbo vstavimo $x = \alpha$, dobimo konstanto A:

$$A = \frac{p(\alpha)}{\alpha^r}.$$

Če je β poljubna ničla polinoma $q(x)$, potem je tudi ničla polinoma

$$p(x) - \frac{p(\alpha)}{\alpha^r} x^r = 0.$$

Če je motnja $(p(\alpha)/\alpha^r)x^r$ majhna, potem je kombinirana redukcija stabilna. Pri danem α indeks r izberemo tako, da je relativna motnja posameznega koeficienta prvotne enačbe najmanjša. Veljati mora

$$|a_{n-r}\alpha^r| = \max_{0 \le k \le n} |a_{n-k}\alpha^k|.$$

Če je $r = 0$, izvedemo direktno redukcijo, če je $r = n$ pa obratno redukcijo. V primeru, ko je $0 < r < n$ izberemo kombinirano redukcijo. S tem si zagotovimo, da bodo ničle reduciranega polinoma najboljši približki za ničle prvotnega polinoma.

3.5 Durand-Kernerjeva metoda

Durand-Kernerjeva metoda je zelo uporabna metoda, saj z njo izračunamo vse ničle polinoma hkrati. Naj bodo x_i, $i = 1, \ldots, n$, približki za enostavne ničle polinoma $p(x)$. Iščemo popravke $\triangle x_i, i = 1, \ldots, n$, da bodo $x_i + \triangle x_i$ čim bližje pravim ničlam. Želimo imeti

$$\prod_{i=1}^{n} (x - (x_i + \triangle x_i)) = p(x), \tag{3.14}$$

kjer bi členi $x_i + \triangle x_i$ bili enaki ničlam polinoma $p(x)$. Če levo stran enačbe (3.14) uredimo po členih $\triangle x_i$, dobimo

$$\prod_{j=1}^{n} (x - x_j) - \sum_{j=1}^{n} \triangle x_j \prod_{\substack{k=1 \\ k \neq j}}^{n} (x - x_k) + \sum_{\substack{j,k=1 \\ j<k}}^{n} \triangle x_j \triangle x_k \prod_{\substack{l=1 \\ l \neq j,k}}^{n} (x - x_l) - \cdots = p(x).$$

Da dobimo iteracijsko funkcijo, izpustimo vse člene, ki imajo potenco $\triangle x$ višjo kot prvi člen, kjer se pojavi:

$$\prod_{j=1}^{n} (x - x_j) - \sum_{j=1}^{n} \triangle x_j \prod_{\substack{k=1 \\ k \neq j}}^{n} (x - x_k) = p(x).$$

Če sedaj vstavimo $x = x_i$, dobimo

$$- \triangle x_i \prod_{\substack{k=1 \\ k \neq i}}^{n} (x_i - x_k) = p(x_i), \qquad i = 1, \ldots, n.$$

Definirajmo polinom

$$q(x) := \prod_{k=1}^{n} (x - x_k) \tag{3.15}$$

in dobimo enačbo

$$\triangle x_i = \frac{-p(x_i)}{q'(x_i)}, \qquad i = 1, \ldots, n.$$

Končno lahko zapišemo Durand-Kernerjevo metodo kot

$$x_i^{(r+1)} = x_i^{(r)} - \frac{p(x_i^{(r)})}{\prod_{\substack{k=1 \\ k \neq i}}^{n} \left(x_i^{(r)} - x_k^{(r)} \right)}, \qquad i = 1, \ldots, n. \tag{3.16}$$

Za iteracijsko funkcijo iz enačbe (3.16) velja, da so vsi prvi odvodi enaki 0, medtem ko drugi odvodi niso vsi ničelni, zato je konvergenca v bližini ničle kvadratična (glej [1, str. 343]). Metoda konvergira skoraj za vsak začetni vektor $x^{(0)} = [x_1^{(0)} \cdots x_n^{(0)}]^T \in \mathbb{C}^n$, pri katerem morajo biti vse komponente paroma različne, saj bi sicer pri naslednjem približku dobili v imenovalcu ničlo. Tako lahko za začetni vektor vzamemo kar n naključnih kompleksnih števil (glej [5, str. 61]).

Konvergenco lahko izboljšamo do superkvadratične, če pri računanju uporabimo nove izračunane vrednosti (glej [5, str. 61]). Dobimo algoritem, pri katerem iteriramo le približke z_i, ki še niso skonvergirali:

$$x_i^{(r+1)} = x_i^{(r)} - \frac{p(x_i^{(r)})}{\prod_{k=1}^{i-1} \left(x_i^{(r)} - x_k^{(r+1)} \right) \prod_{k=i+1}^{n} \left(x_i^{(r)} - x_k^{(r)} \right)}, \quad i = 1, \ldots, n.$$

Primer 3.5.1. Vzemimo polinom

$$x^5 - 4x^4 + 6x^3 - 3x^2 + 2x + 2 = 0,$$

iz primera 3.4.1. Za začetni vektor si izberimo $x^{(0)} = [-0.5, 0, 1, i, 2 + i]$ in z uporabo metode (3.16) dobimo v 7-ih korakih izračunane vse ničle polinoma.

Dobimo naslednje konvergentno zaporedje vektorjev

$$
\begin{aligned}
x^{(1)} &= \begin{bmatrix} -0.47126 + 0.34483i \\ -1.36180 + 0.70112i \\ 1.46652 + 0.26637i \\ 0.23366 - 1.12027i \\ 1.94738 - 0.98457i \end{bmatrix} \\
x^{(2)} &= \begin{bmatrix} -0.88411 + 0.34481i \\ 1.89181 + 2.26815i \\ 1.12889 + 0.58862i \\ 0.24124 - 1.02471i \\ 1.95371 - 1.01362i \end{bmatrix} \\
&\vdots \\
x^{(5)} &= \begin{bmatrix} -0.41651 - 0.00260i \\ 1.94874 + 1.01730i \\ 0.26556 + 0.94911i \\ 0.26548 - 0.94890i \\ 1.94716 - 1.02570i \end{bmatrix} \\
x^{(6)} &= \begin{bmatrix} -0.42536 - 0.00003i \\ 1.94715 + 1.02570i \\ 0.26552 + 0.94885i \\ 0.26552 - 0.94885i \\ 1.94715 - 1.02570i \end{bmatrix} \\
x^{(7)} &= \begin{bmatrix} -0.42534 + 0.00000i \\ 1.94715 + 1.02570i \\ 0.26552 + 0.94885i \\ 0.26552 - 0.94885i \\ 1.94715 - 1.02570i \end{bmatrix}.
\end{aligned}
$$

Ko metoda eno ničlo izračuna dovolj natančno, je ne spreminja več. Za prvo ničlo je metoda potrebovala 7 iteracij, za ostale pa 6.

3.6 Aberth-Ehrlichova metoda

Aberth-Ehrlichova metoda je sorodna Durand-Kernerjevi metodi, saj tudi tu računamo vse ničle hkrati, le da pri Aberth-Ehrlichovi metodi, poleg vrednosti polinoma, potrebujemo tudi vrednosti odvodov. Glede na enačbo (3.16) definirajmo

$$R_i(x) := \frac{-p(x)}{\prod_{\substack{k=1 \\ k \neq j}}^{n} (x - x_k)}, \qquad i = 1, \ldots, n.$$

Sedaj uporabimo tangentno metodo na racionalni funkciji $R_i(x)$ v točki x_i in dobimo

$$\triangle x_i = \frac{-R_i(x_i)}{R_i'(x_i)}, \qquad i = 1, \ldots, n.$$

Na ta način dobimo Aberth-Ehrlichovo metodo:

$$\triangle x_i = \frac{p(x_i)}{p(x_i) \sum_{\substack{k=1 \\ k \neq i}}^{n} \frac{1}{x_i - x_k} - p'(x_i)} \quad i = 1, \ldots, n, \tag{3.17}$$

kjer je $\triangle x_i = x_i^{(r+1)} - x_i^{(r)}$. Če uporabimo polinom $q(x)$ iz (3.15), potem lahko vsoto v enačbi (3.17) zapišemo kot $q''(x_i)/2q'(x_i)$ in dobimo

$$\triangle x_i = \frac{2p(x_i)q'(x_i)}{p(x_i)q''(x_i) - 2p'(x_i)q'(x_i)}.$$

Prednost te metode v primerjavi z Durand-Kernerjevo je, da ima v bližini enostavnih ničel kubično konvergenco (glej [1, str. 343]), saj so vsi prvi in drugi odvodi enaki 0, tretji pa ne.

Primer 3.6.1. Zopet si poglejmo polinom (3.11). Če si za začetni vektor izberemo $x^{(0)} = [-0.5, 0, 1, i, 2+i]$, potem z uporabo metode (3.17) dobimo v treh korakih izračunane vse ničle polinoma. Dobimo naslednje konvergentno zaporedje vektorjev

$$x^{(1)} = \begin{bmatrix} -0.41514 + 0.00680i \\ 0.00609 + 0.86161i \\ 2.00075 + 0.93963i \\ 0.26246 - 0.95277i \\ 1.94712 - 1.02567i \end{bmatrix}$$

$$x^{(2)} = \begin{bmatrix} -0.42531 - 0.00002i \\ 0.26377 + 0.94965i \\ 1.94714 + 1.02569i \\ 0.26552 - 0.94885i \\ 1.94715 - 1.02570i \end{bmatrix}$$

$$x^{(3)} = \begin{bmatrix} -0.42534 + 0.00000i \\ 0.26552 + 0.94885i \\ 1.94715 + 1.02570i \\ 0.26552 - 0.94885i \\ 1.94715 - 1.02570i \end{bmatrix}.$$

Za prvi dve ničli je metoda potrebovala 3 iteracije, za ostale tri ničle pa 2. Vidimo, da Aberth-Ehrlichova metoda na tem primeru res hitreje konvergira od Durand-Kernerjeve v primeru 3.5.1.

Poglavje 4

Zaključek

Skozi zaključno projektno nalogo smo spoznali numerične metode, namenjene reševanju poljubnih nelinearnih enačb in polinomov. Vsem metodam je dodan primer, ki bralcu omogoča lažje razumevanje snovi. Večino primerov sem reševala s programom Octave in rešitve primerjala z navedenimi trditvami v določenem poglavju.

Od metod omenjenih v drugem poglavju najpočasneje konvergira bisekcija, nekoliko hitreje metoda regula falsi, še hitreje sekantna metoda, sledita ji Mullerjeva metoda in inverzna interpolacija, najhitrejša pa je tangentna metoda. Pri metodi navadne iteracije je konvergenca približkov odvisna od velikosti absolutne vrednosti odvoda iteracijske funkcije. Tangentna, sekantna in Mullerjeva metoda so primerne tudi za računanje kompleksnih ničel.

V tretjem poglavju smo spoznali metode prirejene za reševanje polinomskih enačb. Med njimi so najpomembnejše Laguerrova, Durand-Kernerjeva in Aberth-Ehrlichova metoda. Zadnji dve hkrati računata vse ničle polinoma. Na primeru smo videli, da Aberth-Ehrlichova metoda, ki poleg vrednosti polinoma uporablja tudi vrednosti odvodov, konvergira hitreje kot Durand-Kernerjeva.

Numerične metode se zaradi tehnologije tudi same stalno razvijajo. Kljub temu se še vedno uporabljajo metode, ki jih poznamo že več stoletij, le da so nekatere med njimi nekoliko prirejene. Z razvojem računalnikov pa so preporod doživeli celo nekateri algoritmi, ki so veljali za potratne in neuporabne. S povečevanjem računske moči je možno reševati vedno večje probleme. Ker se hiter razvoj računalništva še vedno nadaljuje, lahko pričakujemo, da se bodo v prihodnje vedno bolj razvijale tudi numerične metode.

Literatura

[1] O. Aberth, Iteration Methods for Finding all Zeros of a Polynomial Simultaneously, Mathematics of Computation, **27** (122) (1973), 339–344.

[2] Z. Bohte, Numerično reševanje nelinearnih enačb, Društvo matematikov, fizikov in astronomov Slovenije, Ljubljana, 1993.

[3] S. A. Levin, Descartes' Rule of Signs - How hard can it be?, 2002, URL: `http://sepwww.stanford.edu/oldsep/stew/descartes.pdf/` (16. 8. 2011).

[4] B. Orel, Osnove numerične matematike, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, 2004.

[5] B. Plestenjak, Numerične metode, delovna verzija, 2010, URL: `www-lp.fmf.uni-lj.si/plestenjak/vaje/nafgg/Predavanja/Knjiga_NM.pdf/` (16. 8. 2011).

[6] E. Zakrajšek, Uvod v numerične metode, Društvo matematikov, fizikov in astronomov Slovenije, Ljubljana, 2000.

[7] H. Žolandek, The topological proof of Abel-Ruffini theorem, Journal of the Juliusz Schauder Center **16** (2000), 253–265.

[8] URL: `http://en.wikipedia.org/wiki/Descartes'_rule_of_signs/` (16. 8. 2011).

[9] URL: `http://en.wikipedia.org/wiki/Sturm's_theorem/` (16. 8. 2011).