UNIVERSITETI NËNË TEREZA

FAKULTETI I SHKENCAVE TË INFORMATIKËS

PROGRAMI STUDIMOR: MATEMATIKË-INFORMATIKË

Punim magjistrature

Tema

EFIKASITETI DHE KRAHASUESHMËRIA E DISA METODAVE TE IZOLIMI I RRËNJËVE TE POLINOMET

Mentor: Doc. dr. Egzona Iseni *Student: Vaid Halidi*

Korrik, 2021

Përmbajtja

Lista e tabelave 4
Lista e figurave 5
Abstrakti.. 6
Abstract 7
Hyrje 8
1. Polinomet dhe rrënjët e polinomeve 10
1.1 Polinomet njëvaribëlshe 10
1.2 Rrënjët e polinomeve njëvaribëlshe 10
1.3 Përafrimi i rrënjëve duke shfrytëzuar metoda iterative 11
1.3.1 Metoda e përgjysmimit 11
1.3.2 Metoda e Newton-it për përafrim të rrënjëve 13
1.3.3 Metoda e Halley-it 17
2. Disa metoda për izolim të rrënjëve 19
2.1 Metoda grafike për izolimin e rrënjëve. 21
2.2 Metoda analitike për lokalizimin e rrënjëve 23
2.3 Skema e Horner-it 24
2.4 Metoda e Müller-it 26
2.5 Metoda e Newton-it për lokalizimin e rrënjëve 29
2.6 Rregulla e Descartes-it për shenjat 33
3. Metoda e Sturm-it dhe Vincent-it 34
3.1 Metoda e Sturm-it 34
3.2 Metoda e Vincent-it 38
3.2.1 Metoda e Vincent-Akritas-Strzebonski (VAS) 40
3.2.2 Metoda e Vincent-Collins-Akritas (VCA) 41
4. Implementimi i Matlab për izolimin dhe krahasimin e rrënjëve të polinomeve 44
4.1 Metoda e përgjysmimit 44
4.2 Metoda e Newton-it për përafrimin e rrënjëve 45
4.3 Metoda e Halley-it 47
4.4 Metoda e Horner-it 49
4.5 Metoda e Newton-it për lokalizimin e rrënjëve 51
4.6 Metoda e Sturm-it 53

4.7 Metoda e Vincent-it 57
Përfundimi 63
Bibliografia 65

Lista e tabelave

Tabela 2.1: Vlerat për funksionet $P(x) = x^3$ dhe $Q(x) = 8x - 5$ 22
Tabela 2.2: Vlerat për funksionet $P(x) = x^3$ dhe $Q(x) = -2x + 4$ 22
Tabela 2.3: Tabela e shenjave për disa vlera të $x^3 - 8x + 5$ dhe $3x^2 - 8$ 24
Tabela 2.4: Tabela e shenjave për $x^3 + 2x - 4$ dhe $3x^2 + 2$ 24
Tabela 3.1: Tabela e Sturm-it për $P(x) = x^3 + x^2 - 2x - 1$ 38
Tabela 3.2: Tabela e zgjeruar e Sturm-it për $P(x) = x^3 + x^2 - 2x - 1$ 38

Lista e figurave

Figura 1.1: Metoda e përgjysmimit ... 11
Figura 1.2: Paraqitja grafike e metodës së Newton-it për përafrimin e rrënjëve ... 17
Figura 1.3: Algoritmi i metodës së Newton-it ... 17
Figura 2.1: Paraqitja grafike e $P(x)$ dhe $Q(x)$ ... 22
Figura 2.2: Paraqitja grafike e $P(x)$ dhe $Q(x)$ ... 23
Figura 2.3: Metoda e Müller-it ... 29
Figura 3.1: Algoritmi i Sturm-it ... 36
Figura 3.2: Algoritmi i Sturm-it (gjetja e rrënjëve) ... 36
Figura 3.3: Algoritmi VAS ... 41
Figura 3.4: Algoritmi VCA ... 43
Figura 4.1: Metoda e përgjysmores në Matlab ... 45
Figura 4.2: Rezultatet e metodës se përgjysmores ... 45
Figura 4.3: Paraqitja grafike e $P(x) = 2x^4 - 3x - 2$ ... 45
Figura 4.4: Metoda e Newton-it në Matlab ... 46
Figura 4.5: Rezultati i metodës së Newton-it ... 47
Figura 4.6: Metoda e Halley-it në Matlab ... 49
Figura 4.7: Rezultati i metodës së Halley-it ... 49
Figura 4.8: Kodi i metodës së Horner-it ... 51
Figura 4.9: Rezultati i metodës së Horner-it ... 51
Figura 4.10: Metoda e Newton-it për lokalizimin e rrënjëve ... 52
Figura 4.11: Rezultati i metodës së Newton-it për lokalizimin e rrënjëve ... 53
Figura 4.12: Paraqitja grafike e funksionit $P(x) = 3x^2 + 4x - 5$ ... 53
Figura 4.13: Kodi i metodës në Sturm-it ... 54
Figura 4.14: Rezultati i metodës së Sturm-it ... 54
Figura 4.15: Gjetja e rrënjëve me metodën e Sturm-it ... 57
Figura 4.16: Rezultati i rrënjëve sipas metodës së Sturm-it ... 57
Figura 4.17: Algoritmi VCA ... 59
Figura 4.18: Rezultatet e kodit VCA ... 59
Figura 4.19: Paraqitja grafike e $P(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$ ... 60
Figura 4.20: Izolimi i rrënjëve me metodën VCA, dhe përafrimi me metodën e përgjysmimit ... 62

Abstrakti

Në këtë punim kemi paraqitur një numër të konsiderueshëm metodash për izolimin e rrënjëve të polinomeve, edhe atë metoda grafike për izolimin e rrënjëve, metoda analitike për lokalizimin e rrënjëve, Skema e Horner-it, metoda e Müller-it, metoda e Newton-it për lokalizimin e rrënjëve, metoda e Sturm-it dhe metoda e Vincent-it (metoda e Vincent-Akritas-Strzebonski dhe metoda e Vincent-Collins-Akritas).

Gjithashtu, përveç kësaj, për secilën metodë janë zgjidhur shembuj, të cilët na kanë shërbyer për të parë konkretisht përparësitë dhe mangësitë e secilës metodë. Për më tepër, të gjitha këto metoda i janë nënshtruar edhe studimit në aspektin softuerik, përkatësisht përmes krijimit të kodeve në Matlab, i cili na ka dhënë një pasqyrë të pastër për secilën prej tyre, madje edhe e ka karakterizuar shpejtësia dhe saktësia e zgjidhjes, gjë kjo mjaft e rëndësishme për izolimin e rrënjëve të polinomeve.

Fjalët kyçe: polinomet, izolimi i rrënjëve reale, efikasiteti, krahasimi i metodave për izolim të rrënjëve, Matlab, metoda e Sturm-it, metoda e Vincent-it, Geogebra.

Abstract

On this thesis have presented a considerable number of root isolation methods such as graphic method, analytic method, Horner's scheme, Müller's scheme, Newton's localization method, Sturm's method and Vincent's (Vincent-Akritas-Strzebonski method and Vincent-Collins-Akritas method).

Also, for all the above-mentioned methods we have solved some examples, while solving those examples we presented some advantages and disadvantages of these methods. Furthermore all these methods are solved using software methods, more specifically Matlab, where we get a clear image and we can compare their speeds and their precisions, which is very important while isolating roots.

Keywords: polynomials, real root isolation, efficiency, isolation method comparison, Matlab, Sturm's method, Vincent's method, Geogebra.

Hyrje

Teoria e polinomeve përbën një teoritë më të përhapura matematike, e që trajtohet në degë të ndryshme matematike, si në algjebër, analizë, si dhe në analizën numerike. Me fjalë tjera, ata zënë një vend të pazëvendësueshëm në teorinë e përafrimeve, dhe jo vetëm.

Aplikueshmëria e tyre e gjerë ka qenë një ndër motivet kyçe që na shtyri në trajtimin e kësaj problematike. Duke marr në konsideratë se si teori është shumë e gjerë, ne u fokusuam vetëm në një pjesë të saj të veçantë, atë të gjetjes së rrënjëve, mirëpo me metoda të analizës numerike. Nuk po themi se kemi thënë gjithçka për to, po megjithatë, pretendojmë se kemi hedhur në pah pjesë të rëndësishme të kësaj lëmie.

Trajtimi i temës është bërë në katër kapituj edhe atë:

Në kapitullin e parë, do të japim një vështrim të përgjithshëm mbi polinomet, si përkufizimet, teoremat (lexo: vetitë) më themelore që vlejnë në përgjithësi për polinomet, duke i ilusfruar ato edhe me ndonjë shembull, ku e kemi parë të nevojshme.

Në kapitullin e dytë, jemi ndalur te disa metoda për izolim të rrënjëve, gjithsejtë gjashtë sosh, ku secila është trajtuar si temë më vete dhe duke paraqitur boshtin e idesë së secilës prej tyre , kuptohet edhe duke i ilustruar edhe me shembuj konkret.

Në kapitullin e katërtë, i kemi shtjelluar Sturm-it dhe Vincent-it, duke ja bashkangjitur atyre edhe metodën Vincent-Akritas-Strzebonski dhe Vincent-Collins-Akritas. Për të cilat, përveç shembujve që do të japim duke i zgjidhur me metodat përkatëse, do të japim edhe kode në Matlab për gjetjen e rrënjëve të polinomëve për softuerit të lartëcekur, në të cilin janë dhënë kodet në formën që të zgjidhin problemin tonë me metodat e sipërpërmendura.

Në kapitullin e katërtë kemi vënë në pah efikasitetin e softuerit Matlab në zgjidhjen e kësaj problematike, ku përmes këtij programi janë dhënë kode për secilën nga metodat që është trajtuar teoritikisht janë dhënë shembuj të zgjidhur përmes tij, respektivisht metodës që kemi trajtuar më herët.Njëherit, janë bërë edhe krahasime në mes të metodave në shpejtësinë dhe saktësinë e zgjidhjeve.

Le të përmendim edhe këtë se, disa nga grafiqet e polinomeve , të cilat kanë qenë të nevojshme për konkretizim të temës janë realizuar me Geogebra.

Nuk marrim përsipër të themi se kemi bërë të përkryerën, po se kemi dhënë më të mirën.

1. Polinomet dhe rrënjët e polinomeve

1.1 Polinomet njëvaribëlshe

Përkufizim 1.1 *[1]* Shprehja matematike e formës

$$P_n(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_0 \tag{1.1}$$

polinom njëvariabëlsh i fuqisë n, ku $a_0, a_1, \dots, a_n$ janë koeficientët e polinomit, kurse n është një numër i plotë jonegativ dhe x quhet variabël.

Përkufizim 1.2 Shkalla e polinomit (1.1) është më e madhja prej n-ve për të cilën $a_0 \neq 0$ (një n e tillë ekziston pasi bashkësia $\{i|a_i \neq 0\}$ është bashkësi e fundme), shkallën e një polinomi P do ta shënojmë deg P.

Përkufizim 1.3 Koeficientin pranë shkallës më të madhe e quajmë *koeficienti kryesor*.

Përkufizim 1.4 Polinomi (1.1) quhet *polinom linear* nëse $deg(P_n) = 1$.

Përkufizim 1.5 Një polinom quhet *monik* nëse koeficienti kryesor është 1 pra $a_0 = 1$

Përkufizim 1.6 Nëse një polinom i ka të gjithë koeficientët $a_i = 0, i = 1,2, \dots, n$ atëherë ky polinom quhet *polinom zero*.

Përkufizim 1.7 Nëse për çdo $a_k, 0 \leq k \leq n$ kemi që $a_k \in \mathbb{R}$ atëherë themi se polinomi (1.1) është polinom real.

1.2 Rrënjët e polinomeve njëvaribëlshe

Përkufizim 1.8 Le të jetë (1.1) një polinom real dhe $c \in \mathbb{R}$, atëherë elementin $a_0c^n + a_1c^{n-1} + \cdots + a_n$ e quajmë vlera e polinomit për $x = c$ dhe e shënojmë me $P(c)$.

Duke shfrytëzuar mund të krijojmë një funksion polinomial duke e paraqit çdo $x \in \mathbb{R}$ te rezultati përkatës $P(x) = a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_0$ [2]

Përkufizim 1.9 Le të jetë (1.1) një polinom real dhe $c \in \mathbb{R}$. Themi se c është rrënjë e polinomit P nëse $P(x) = 0$ *[3]*

1.3 Përafrimi i rrënjëve duke shfrytëzuar metoda iterative

Metodat iterative zakonisht kërkojnë një apo më shumë përafrime fillestare për rrënjën(t) e kërkuara të polinomit. Kjo shpesh paraqet një problem më vete dhe ka teknika dhe metoda për gjetjen e tyre. Metoda më e thjeshtë për të gjetur një supozim është duke shikuar grafikun e polinomit, gjë që shpesh nuk është e mundur (p.sh. kur kemi të bëjmë me polinome shumë komplekse dhe të gjata). Disa nga këto metoda do të tregohen më vonë dhe kështu për këtë pjesë.

1.3.1 Metoda e përgjysmimit

Metoda më e thjeshtë për gjetjen e një përafrimi më të mirë me një rrënjë është *metoda e përgjysmimit*.

Teoremë 1.1 (Teorema e vlerës së ndërmjetme *[4]*) Le të jetë f një funksion në $\mathbb{R}$. E marrim në konsideratë intervalin $I = [a, b]$ ashtu që f është i vazhdueshëm në atë interval I dhe $\lambda \in \mathbb{R}$ ashtu që $f(a) < \lambda < f(b)$, atëherë ekziston një $c \in [a, b]$, ashtu që $f(c) = \lambda$.

Supozojmë se funksioni $f: \mathbb{R} \to \mathbb{R}$ është i vazhdueshëm në $[a, b]$ dhe $f(a) \cdot f(b) < 0$. Atëherë sipas teoremës së vlerës së ndërmjetme duhet të ketë të paktën një rrënjë në intervalin $[a, b]$. Intervali mund të zgjidhet mjaftueshëm i madh që të ketë më shumë se një rrënjë, por kjo nuk paraqet problem, pasi që algoritmi i përgjysmores gjithmonë do të konvergjojë në ndonjë rrënjë $\alpha \in [a, b]$ dhe një interval më të vogël që përmban vetëm një rrënjë. Meqenëse të gjitha funksionet polinomiale janë të vazhdueshëm [5], mund ta shfrytëzojmë këtë teoremë për të krijuar një algoritëm (shih *Figura 1.1*) i cili do te testohet në pjesën e katërt.

funksion përgjysmim (f, a, b, ε)
    $x \leftarrow \frac{a+b}{2}$
    nëse $x - a \leq c$ **atëherë**
        kthe c
    nëse $f(a)f(x) < 0$ **atëherë**
        kthe përgjysmim (f, a, x, ε)
    ndryshe
        kthe përgjysmim (f, x, b, ε)

Figura 1.1: Metoda e përgjysmimit

Në pjesën e katërt të këtij punimi kemi një shembull të metodës së përgjysmimit (shih *Shembull 4.1*).

Shpejtësia e konvergjencës, përderisa çdo iteracion na jep një përafrim më të mirë të zgjidhjes së vërtetë, algoritmi po konvergjon mjaft ngadalë. Në vazhdim do ta shqyrtojmë shpejtësinë e konvergjencës.

Përkufizim 1.10 Supozojmë se kemi një varg numrash real $x_0, x_1, \ldots$ dhe një numër real α ashtu që $\forall \varepsilon \in \mathbb{R}, \varepsilon > 0$ ekziston $k \in \mathbb{N}$ ashtu që $\forall l > k, |x_l - \alpha| < \varepsilon$. Atëherë themi se vargu konvergjon në pikën α dhe e shënojmë $\lim\limits_{k\to\infty} x_k = \alpha$. Për një varg themi se *konvergjon linearisht* nëse

$$\lim_{n\to\infty} \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|} = c$$

për ndonjë $0 < c < 1$. c quhet *koeficienti konvergjencës lineare* të x_n në α.

Le të shënojmë x_n vlerën e n – të të x – it në *Figura 1.1* atëherë

$$\alpha = \lim_{n\to\infty} x_n$$

$$|\alpha - x_n| \leq \left[\frac{1}{2}\right]^n |b - a|$$

$$\lim_{n\to\infty} \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|} = \lim_{n\to\infty} \frac{2^{-(n+1)}|b - a|}{2^{-n}|b - a|} = \frac{1}{2}$$

ku $|b - a|$ është gjatësia e intervalit fillestar. Duke shfrytëzuar *Përkufizim 1.10* themi se algoritmi te Figura 1.1 konvergjon linearisht me koeficient te konvergjencës lineare $\frac{1}{2}$. Kjo nuk do të thotë domosdoshmërish se gabimi zvogëlohet për faktor $\frac{1}{2}$, por mesatarisht zvogëlohet për faktor $\frac{1}{2}$. Kjo do të thotë se nevojiten përafërsisht 3.32 iteracione për të llogaritur vetëm një shifër $(\log_2 10)$.

Pengesa kryesore e këtij algoritmi është shumë i ngadaltë, krahasuar veçanërisht me metodat e tjera të përshkruara në pjesët vijuese. Nga ana tjetër, metoda e përgjysmimit ka disa përparësi. E para prej tyre është që është e garantuar të konvergjojë nëse plotësohen parakushtet. E dyta është ekzistenca e një kufiri të arsyeshëm të gabimit. Kjo metodë siguron kufijtë e sipërm dhe të poshtëm të rrënjës në çdo iteracion dhe i përket klasës së metodave të quajtura *metodat e mbylljes* [6].

1.3.2 Metoda e Newton-it për përafrim të rrënjëve

Metoda e Newton-it (nganjëherë e quajtur edhe metoda Newton-Raphson sa për ta dalluar nga metoda e Newton-it për izolimin e rrënjëve), është një tjetër metodë për të gjetur përafrim më të mirë të një rrënje të një funksioni real polinomial (ose në përgjithësi, çdo funksion me vlera reale). Ideja themelore e metodës bazohet në faktin se duke pasur parasysh një vlerë fillestare x_0, që është mjaftueshëm afër rrënjës, mund të përafroni funksionin duke llogaritur tangjenten e tij në pikën $(x_0, P(x_0))$. Pastaj, mund të përdoret pikëprerja e tangjentës me boshtin x, i cili zakonisht ofron një përafrim më të mirë dhe përsërit këtë proces pafundësisht [6] (ose derisa të arrihet saktësia e dëshiruar).

Supozojmë se funksioni real $P(x)$ është i diferencueshëm në intervalin $[a, b]$ dhe se kemi një përafrim x_0. Duke shfrytëzuar kalkulusin mund të gjejmë një përafrim më të mirë për x_1 si në vazhdim.

Në fillim e njehsojmë koeficientin e drejtimit m të tangjentës

$$m = \frac{P(x_1) - P(x_0)}{x_1 - x_0}$$

Pasi e kërkojmë pikëprerjen e tangjentës dhe boshtit x zëvendësojmë te $P(x_1)$ me 0 dhe fitojmë

$$x_1 = x_0 - \frac{P(x_0)}{m}$$

Pasi që m është koeficienti i drejtimit të tangjentës, këtë mund ta shënojmë edhe si derivati i P në pikën x_0 nga fitojmë

$$x_1 = x_0 - \frac{P(x_0)}{P'(x_0)}$$

Duke e gjeneralizuar këtë proces fitojmë

$$x_{n+1} = x_n - \frac{P(x_n)}{P'(x_n)} \tag{1.2}$$

Meqenëse të gjithë funksionet polinomiale janë të diferencueshëm, ne mund ta aplikojmë këtë formulë për problemet tona. Metoda e Njutonit është jashtëzakonisht e fuqishme dhe një nga teknikat më të njohura për gjetjen e rrënjëve, pasi është mjaft e lehtë për t'u zbatuar dhe konvergjon shumë shpejt.

Teoremë 1.2 *[6]* Supozojmë se $P(x), P'(x)$ dhe $P''(x)$ janë të vazhdueshëm për çdo x në rrethinën $I = [\alpha - \varepsilon, \alpha + \varepsilon]$ e α, ku α është një rrënjë e P dhe $P'(x) \neq \forall x \in I$. Atëherë nëse x_0 është mjaftueshëm afër α, atëherë iteracionet $x_n, n \geq 0$ te (1.2) do të konvergjojnë në α. Gjithashtu

$$\lim_{n\to\infty} \frac{(\alpha - x_{n+1})}{(\alpha - x_n)^2} = \frac{-P''(\alpha)}{2P'(\alpha)} > 0$$

tregon se konvergjenca është kuadratike.

Sidoqoftë, kjo metodë ka disa mangësi.

Së pari, vetëm duke vëzhguar ekuacionin, mund të shohim se nëse x_n është i maksimum ose minimum lokal, atëherë x_{n+1} është i papërcaktuar, pasi $P'(x_n) = 0$ (që do të thotë se tangjente është paralele me boshtin x).

Së dyti, për ndonjë funksion polinomial, metoda mund të hyjë në një cikël të pafundëm. Kjo ndodh nëse, për shembull, x_1 jep si rezultat $x_2 = x_0$, duke bërë që metoda të alternohet midis këtyre dy rezultateve pafundësisht.

Së treti, nëse supozimi fillestar nuk është mjaft i afërt ose derivati i parë nuk sillet mirë në afërsi të një rrethine specifike, metoda mund ta kalojë këtë rrënjë pasi tangjente do ta pret boshtin x afër një rrënje tjetër. Kjo mund të paraqesë një problem, nëse jemi duke kërkuar një rrënjë që gjendet në një interval $[a, b]$, por nuk paraqet problem nëse thjesht kërkojmë ndonjë rrënjë të një polinomi.

Së fundmi, nëse rrënja ka shumëfishitet më të madhe se një (d.m.th. derivati i parë i funksionit polinomial në rrënjë është gjithashtu zero), atëherë konvergjenca në këtë rrënjë është lineare, gjë që nuk na ndihmon sepse mund ta shfrytëzojmë metodën e përgjysmimit pasi ishte më e thjeshtë

Teoremë 1.3 Nëse rrënja e një polinomi $P(x)$ ka shumëfishitet më të madh se një atëherë konvergjenca është lineare

Vërtetim: Le të jetë $P(x)$ një funksion polinomial me një rrënjë α që ka shumëfishitet $m > 1$, pra $P(x) = (x - \alpha)^m g(x)$. Supozojmë se x_0 është mjaftueshëm afër α siç në *Teoremë 1.2* ,atëherë

$$x_{n+1} = x_n - \frac{(x_n - \alpha)^m g(x_n)}{m(x_n - \alpha)^{m-1} g(x_n) + (x_n - \alpha)^m g'^{(x_n)}}$$

$$= x_n - \frac{(x_n - \alpha) g(x_n)}{m g(x_n) + (x_n - \alpha) g'(x_n)}$$

$$= \frac{x_n\big(m g(x_n) + (x_n - \alpha) g'(x_n)\big) - (x_n - \alpha) g(x_n)}{m g(x_n) + (x_n - \alpha) g'(x_n)}$$

$$= \frac{x_n(m g(x_n) - 1) + x_n(x_n - \alpha) g'(x_n) - \alpha g(x_n)}{m g(x_n) + (x_n - \alpha) g'(x_n)}$$

që kur afrohemi afër rrënjës $x_n \approx \alpha$ na jep

$$x_{n-1} \approx x_n \frac{m-1}{m} + \frac{\alpha}{m}$$

ose

$x_{n-1} \approx (x_n - \alpha) \frac{m-1}{m} + \alpha$

prej nga konkludojmë se

$$\lim_{n \to \infty} \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|} = \frac{m-1}{m}$$

që sipas *Përkufizim 1.10* kemi se konvergjon linearisht.

Për të vlerësuar gabimin dhe për të siguruar kufijtë e gabimit të llogaritjes sonë, do të duhet të përdorim një variacion të teoremës së vlerës mesme.

Teoremë 1.4 (Teorema e vlerës së mesme *[7]*) Le të jetë $P(x)$ një polinom real dhe a dhe b dy numra real ashtu që $a < b$. Atëherë ekziston një $c \in (a, b)$ ashtu që

$$P'(c) = \frac{f(b) - f(a)}{b - a}$$

Thënë ndryshe, teorema thotë se në një lakore, e cila fillon në pikën a dhe përfundon në pikën b, ekziston të paktën një pikë në të cilën tangjenta e tangjenta është paralele me sekantën që lidh dy pikat e skajshme. Duke përdorur këtë teoremë në një funksion polinomial $=P$ me një rrënjë α dhe një përafrim $x_n, \alpha < x_n$ marrim

$$P'(\zeta_n) = \frac{P(x_n) - P(\alpha)}{x_n - \alpha}$$

ku ζ_n gjendet në mes α dhe x_n. Pasiqë α është rrënjë e P ekuacioni i mësipërm mund të shprehet si

$$\alpha - x_n = \frac{-P(x_n)}{P'(\zeta_n)}$$

Ngjashëm nëse $x_n < \alpha$ do të arrijmë te e njëjta formulë. Nëse $P'(x)$ nuk ndryshon vazhdimisht në mes x_n dhe α atëherë $P'(\zeta_n) \approx P'(x_n)$. Duke e kombinuar me përkufizimin e metodës së Newton-it kemi

$$\alpha - x_n \approx \frac{-P(x_n)}{P'(x_n)} = x_{n+1} - x_n$$

që na jep gabimin absolut

$$\alpha - x_n \approx x_{n+1} - x_n \tag{1.3}$$

dhe gabimin relativ

$$\frac{\alpha - x_n}{\alpha} \approx \frac{x_{n+1} - x_n}{x_{n+1}} \tag{1.4}$$

Duke shfrytëzuar formulën (1.2) dhe ekuacionet (1.3) dhe (1.4) mund të krijojmë algoritmin e metodës së Newton-it (shih *Figura 1.3*)

Në pjesën e katërt kemi një shembull të koduar në Matlab, (shih *Shembull 4.2*)

Në

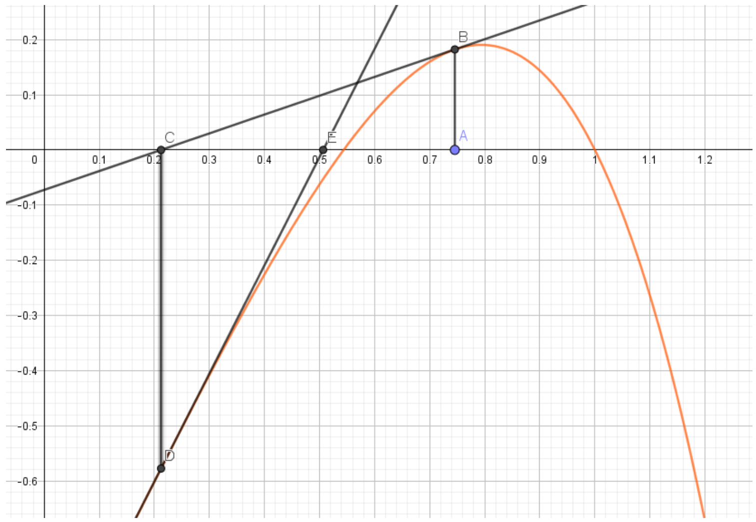

Figura 1.2: Paraqitja grafike e metodës së Newton-it për përafrimin e rrënjëve

funksion përgjysmim $(f, f', a, b, \varepsilon, max)$
  $x \leftarrow \frac{a+b}{2}$
  $err \leftarrow 1$
  për $i \leftarrow 1, max$
    emëruesi $\leftarrow f'(x)$
    nëse emëruesi=0 **atëherë**
      $err \leftarrow 1$
      kthe err, x_0
    $x_1 \leftarrow x_0 \frac{f(x_0)}{emëruesi}$
    nëse $\left(\left|\frac{x_1 - x_0}{x_1}\right| \leq \varepsilon\right)$ **dhe** $(|x_1 - x_0| \leq \varepsilon)$ **atëherë**
      $err \leftarrow 0$
      kthe err, x_1
    nëse $x_1 < a$ **ose** $x_1 > b$ **atëherë**
      $err \leftarrow 3$
      kthe err, x_1
    $x_0 \leftarrow x_1$
  kthe err, x_1

Figura 1.3: Algoritmi i metodës së Newton-it

1.3.3 Metoda e Halley-it

Është emëruar sipas Edmond Halley, është një metodë që përveç derivatit të parë shfrytëzon edhe derivatin e dytë. Përderisa metoda e Newton-it mund të shprehet gjeometrikisht si një seri tangjentash rrënjët e të cilave konvergjojnë në rrënjën e funksionit, për metodën e Halley nuk ka interpretim kaq të qartë. Sidoqoftë, metoda e Newton-it mund të përfitohet nga polinomi i rendit të parë të Taylor-it, metoda e Halley mund të nxirret përfitohet nga polinomi i rendit të dytë Taylor-it [8].

$$y(x) = P(x_n) + P'(x_n)(x - x_n) + \frac{P''(x_n)}{2!}(x - x_n)^2$$

ku x_n është një përafrim i x ashtu që $P(x) = 0$. Pasi që objektivi është të njehsojmë pikën x_{n+1} ku funksioni y e pret boshtin x, e marrim $y(x) = 0$ pastaj e zgjedhim ekuacionin

$$0 = P(x_n) + P'(x_n)(x_{n+1} - x_n) + \frac{P''(x_n)}{2!}(x_{n+1} - x_n)^2$$

për x_{n+1}. Nga [8] kemi dhe nga thjeshtimi i ekuacionit kemi se

$$x_{n+1} = x_n - \frac{P(x_n)}{P'(x_n) + \frac{P''(x_n)}{2}(x_{n+1} - x_n)}$$

duke zëvendësuar $x_{n+1} - x_n$ me $-\frac{P(x_n)}{P('x_n)}$ nga (1.2), kemi

$$x_{n+1} = x_n - \frac{2P(x_n)P'(x_n)}{2P'(x_n)^2 + 2P(x_n)P''(x_n)} \tag{1.5}$$

që njihet si metoda e Halley-it.

Shpejtësia e konvergjencës, metoda e Halley-it konvergjon në rrënjën në mënyrë kubike [9] [10], në krahasim me metodën e Njutonit, e cila konvergjon në mënyrë kuadratike. Vërtetimi i konvergjencës kubike është i ngjashëm me vërtetimin e metodës së Newton-it, duke përdorur zgjerimin e serisë Taylor. Konvergjenca kubike arrihet me koston e llogaritjeve më komplekse në çdo iteracion. Përveç kësaj, nëse derivati i dytë është afër apo saktësisht 0 atëherë shpejtësia e konvergjencës është shumë e ngjashme me atë të metodës së Newton-it. Algoritmi i Halley është i ngjashëm me atë të Newton-it, veçse kemi si funksion (1.4). Në pjesën e katërt kemi një shembull (shih *Shembull 4.3*.)

2. Disa metoda për izolim të rrënjëve

Përderisa në kapitullin e mëparshëm u përqendruam në arritjen e një përafrimi më të mirë të një rrënje të vetme, në këtë kapitull do të përqendrohemi në zgjidhjen e funksioneve polinom dhe izolimin e rrënjëve.

Le të jetë (1.1) një polinom i shkallës n.

Njehsimi i rrënjëve të tij mund të bëhet me metoda të ndryshme të diskutuara më sipër, por ekzistojnë edhe metoda tjera që janë më të përshtatshme për polinomet algjebrike. Do të kufizohemi në njërën prej tyre duke supozuar se koeficientët $a_0, a_1, \dots, a_n$ janë numra real. Do ti kërkojmë vetëm rrënjët reale të këtyre polinomeve por edhe rrënjët komplekse mund te gjinden me pak ndryshime. Së pari po i kujtojmë disa veti të polinomeve algjebrike:

1. Çdo polinom i fuqisë n i ka gjithsej n rrënjë reale ose komplekse duke numëruar rrënjët e shumëfishta aq herë sa ç'kanë shumëfishitetin.
2. Në qoftë se $x_1, x_2, \dots, x_m$ janë rrënjë të polinomit (1.1) me shumëfishitet përkatësisht $p_1, p_2, \dots p_m$ atëherë ai shkruhet

$$P_n(x) = a_0(x - x_1)^{p_1}(x - x_2)^{p_2} \dots (x - x_m)^{p_m}$$

   ku $p_1 + p_2 + \cdots + p_m = n$

3. Në qoftë se të gjithë koeficientët $a_0, a_1, \dots, a_n$ janë numra realë atëherë rrënjët komplekse të (1.1) janë dy nga dy të konjuguara.
4. Polinomi

$$Q_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

   që merret duke ndërruar renditje e koeficientëve të (1.1) në të kundërt, ka për rrënjë të anasjellat e $P_n(x)$, domethënë, në qoftë se $x = x_0 \neq 0$ është rrënjë e $P_n(x)$ atëherë $\frac{1}{x_0}$ do të jetë rrënjë e $Q_n(x)$ dhe anasjelltas.

5. Në qoftë se $P_n(x)$ i ka rrënjët si tek vetia 2, atëherë polinomi

$$R_n(x) = a_0(x - x_1)^{p_1 - 1}(x - x_2)^{p_2 - 1} \dots (x - x_m)^{p_m - 1}$$

   Është pjesëtuesi më i madh i përbashkët i $P_n(x)$dhe $P'_n(x)$. Në këto kushte polinomi

$$T(x) = \frac{P_n(x)}{R_n(x)}$$

ka koeficient realë dhe rrënjët $x_1, x_2, \dots, x_m$ të thjeshta.

6. Le të jenë $P(x)$ dhe $Q(x)$ polinome të fuqisë të shumtën n. Në qoftë se $x_1, x_2, \dots, x_m$ ku $m > n$, janë numra të ndryshëm dhe

$$P(x_i) = Q(x_i) \text{ për } i = 1,2, \dots, m$$

atëherë

$$P(x) = Q(x)$$

për të gjitha vlerat e x-it.

7. Le të jenë $x_1, x_2, \dots, x_n$, n rrënjët e (1.1), atëherë janë të vërteta këto barazime

$$\begin{cases} a_1 = -a_0 \sum_{i=1}^{n} x_i \\ a_2 = a_0 \sum_{i<j}^{n} x_i, x_j = a_0\left(x_1x_2 + x_1x_3 + \cdots + x_ix_j + \cdots + x_{n-1}x_n\right) \\ a_n = (-1)^n a_0 x_1 x_2 \dots x_n \end{cases}$$

8. Shënojmë $A = \max\{|a_1|, |a_2|, \dots, |a_n|\}$, $B = \max\{|a_0|, |a_1|, \dots, |a_{n-1}|\}$ atëherë të gjitha rrënjët e$x_k$të $P_n(x)$ plotësojnë kushtin

$$r = \frac{1}{1 + \frac{B}{|a_n|}} < |x_k| < 1 + \frac{A}{|a_0|} = R$$

9. Vetia e Lagrange-it. Supozojmë që $a_i, i = 0,1,2, \dots, n$ janë numra realë dhe $a_0 > 0, a\_k (k \geq 1)$ është koeficienti i parë negativ i polinomit $P_n(x)$ atëherë numri

$$R = 1 + \sqrt[k]{\frac{C}{a_0}}$$

ku C është koeficienti negativ më i madh në vlerë absolute, shërben si kufi i sipërm për rrënjët pozitive

10. Vetia e Newton-it. Në qoftë se $x = C$ polinomi $P(x)$ e derivatet e tij $P'(x), P''(x), \dots, P^{(n)}(x)$ janë jonegativë dhe $P^n(C) > 0$, atëherë C mund të merret si kufi sipërm për rrënjët pozitive $P(x)$.
11. Vetia e Descartes-it. Numri i rrënjëve pozitive të (1.1) është sa numri i ndërrimit të shenjave në vargun e koeficientëve $a_0, a_1, \dots, a_n$. Rrënjët e shumëfishta numërohen aq herë sa shumëfishiteti i tyre dhe koeficientët e barabartë me zero nuk numërohen.

Për të gjetur rrënjët reale të ekuacionit me një saktësi të dëshiruar, me anë të ndonjërës metodë iterative së pari duhet të gjejmë një përafrim.

Një përafrim fillestar gjendet zakonisht duke lokalizuar rrënjën me ndonjërën nga metodat të cilët mundësojnë atë.

Si metoda të tilla njihen:

2.1 Metoda grafike për izolimin e rrënjëve.

Shpeshherë kjo metodë është më e përshtatur për gjetjen e intervalit në të cilën gjenden zerot e funksionit. Kjo metodë bazohet në faktin që rrënjë e funksionit është pikërisht pika ku grafiku i funksionit e pret boshtin x. Në rastin kur të vizatuarit e grafikut është më e komplikuar, atëherë funksionin $P(x) = 0$ e shkruajmë në trajtën $P_1(x) = P_2(x)$ dhe duke studiuar grafikët e funksioneve $y_1 = P_1(x)$ dhe $y_2 = P_2(x)$ gjejmë intervalin në të cilën gjenden rrënjët e funksionit.

Shembull 2.1 Të lokalizohen rrënjët e ekuacioneve me metodën grafike

a) $x^3 - 8x + 5 = 0$
b) $x^3 + 2x - 4 = 0$

Zgjidhje:

a) Në bazë të asaj që e cekëm më lartë, ekuacionin $x^3 - 8x + 5 = 0$ mund ta shënojmë në formën $x^3 = 8x - 5$. Në këtë rast ne duhet të vizatojmë në sistemin kënddrejtë koordinativ lakoren e funksionit $P(x) = x^3$ dhe drejtëzën e funksionit linear $Q(x) = 8x - 5$. Vlerat për funksionet $P(x)$ dhe $Q(x)$ janë paraqitur në *Tabela 2.1*

Tabela 2.1: Vlerat për funksionet $P(x) = x^3$ dhe $Q(x) = 8x - 5$

$\boldsymbol{x}$	-3	-2	-1	0	1	2	3
$\boldsymbol{P(x) = x^3}$	-27	-8	-1	0	1	8	27
$\boldsymbol{Q(x) = 8x - 5}$	-29	-21	-13	-5	3	11	19

Në *Figura 2.1* janë paraqitur grafikisht funksionet $P(x)$ dhe $Q(x)$

Figura 2.1: Paraqitja grafike e $P(x)$ dhe $Q(x)$

Nga grafiqet e funksioneve shohim se ekuacioni $x^3 - 8x + 5 = 0$ ka tre rrënjë dhe ato ndodhen në intervalet $(-3.5, -3)$, $(0,1)$ dhe $(2,3)$.

c) Ngjashëm si shembullin paraprak, ekuacionin $x^3 + 2x - 4 = 0$ mund ta shënojmë në formën $x^3 = -2x + 4$. Vizatojmë tani në sistemin kënddrejtë koordinativ lakoren e funksionit $P(x) = x^3$ dhe drejtëzën e funksionit linear $Q(x) = -2x + 4$. Vlerat për funksionet $P(x)$ dhe $Q(x)$ janë paraqitur në *Tabela 2.2*

Tabela 2.2: Vlerat për funksionet $P(x) = x^3$ dhe $Q(x) = -2x + 4$

$\boldsymbol{x}$	**-2**	**-1**	**0**	**1**	**2**
$\boldsymbol{P(x) = x^3}$	**-8**	**-1**	**0**	**1**	**8**
$\boldsymbol{Q(x) = -2x + 4}$	**8**	**6**	**4**	**2**	**0**

Në *Figura 2.2* janë paraqitur grafikisht funksionet $P(x)$ dhe $Q(x)$

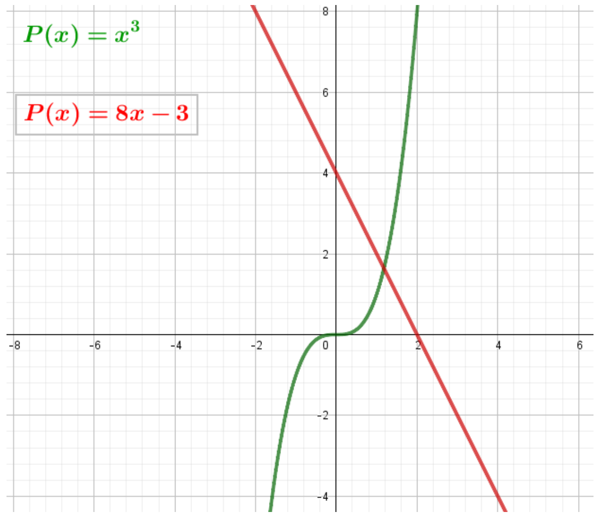

Figura 2.2: Paraqitja grafike e $P(x)$ dhe $Q(x)$

Nga grafiku shohim se ekuacioni $x^3 + 2x - 4 = 0$ ka një rrënjë reale në intervalin $(1, 2)$ dhe dy rrënjë komplekse të konjuguara (meqë $D > 0$).

2.2 Metoda analitike për lokalizimin e rrënjëve

Në qoftë se funksioni $P(x) = 0$ është funksion i vazhdueshëm në intervalin (a, b) dhe në qoftë se në skajet e intervalit merë vlera me shenja të kundërta, d.m.th $P(a)Q(b) < 0$, atëherë intervali (a, b) përmban të paktën një rrënjë të $P(x)$. Në qoftë se $P'(x)$ ruan edhe shenjën në intervalin (a, b) atëherë rrënja është e vetme.

Shembull 2.2 Të lokalizohen rrënjët e ekuacioneve me metodën analitike

a) $x^3 - 8x + 5 = 0$
b) $x^3 + 2x - 4 = 0$

Zgjidhje:

a) Meqë ekuacioni $x^3 - 8x + 5 = 0$ është ekuacion polinomial ai është edhe i vazhdueshëm i cili ka derivat $P'(x) = 3x^2 - 8$. Shqyrtojmë shenjat në tabelën në *Tabela 2.3.*

Tabela 2.3: Tabela e shenjave për disa vlera të $x^3 - 8x + 5$ dhe $3x^2 - 8$

x	-4	-3	-2	-1.5	-1	0	1	1.5	2	3	4
Sgn ($x^3 - 8x + 5$)	-	+	+	+	+	+	-	-	-	+	+
Sgn ($3x^2 - 8$)	+	+	+	-	-	-	-	-	+	+	+

Nga tabela shihet se ekuacioni $x^3 - 8x + 5 = 0$ ka rrënjë në intervalet $(-4, -3)$, $(0, 1)$ dhe $(2, 3)$. Meqë $P'(x) = 3x^2 - 8$ ruan edhe shenjën në këto intervale, atëherë konstatojmë se në këto intervale ka rrënjë të vetme.

b) Meqë ekuacioni $x^3 + 2x - 4 = 0$ është ekuacion polinomial ai është edhe i vazhdueshëm i cili ka derivat $P'(x) = 3x^2 + 2$. Shqyrtojmë shenjat në tabelën në *Tabela 2.4.*

Tabela 2.4: Tabela e shenjave për $x^3 + 2x - 4$ dhe $3x^2 + 2$

x	-3	-2	-1	0	1	2	3
Sgn ($x^3 + 2x - 4$)	-	-	-	-	-	+	+
Sgn ($3x^2 + 2$)	-	+	+	+	+	+	+

Nga tabela shihet se ekuacioni $x^3 + 2x - 4 = 0$ ndron shenjën në intervalin $(1, 2)$, prandaj në këtë interval ai ka rrënjë. Meqë $P'(x) = 3x^2 + 2$ ruan edhe shenjën në këtë interval, atëherë konstatojmë se në këtë interval ka rrënjë të vetme.

2.3 Skema e Horner-it

Metoda e Horner-it mbështetet në atë që njihet si *themelore teorema e algjebrës*.

Teoremë 2.1 (Teorema themelore e algjebrës *[11]*) Për çdo $n \geq 1$ dhe për çdo polinomi njëvariabëlsh P me koeficientë komplekse të tillë që shkalla e P është n, polinomi P ka të paktën një rrënjë komplekse.

Prej kësaj teoreme kemi disa rrjedhime, si:

Rrjedhim 2.1 Çdo polinom njëvariabëlsh P i shkallës $n \geq 1$ ka saktësisht n rrënjë

Rrjedhim 2.2 Çdo polinom njëvariabëlsh P i shkallës $n \geq 1$ mund të faktorizohet si $P = (x - z_1)(x - z_2) \dots (x - z_n)$ ku $z_1, z_2, \dots, z_n$ janë rrënjët komplekse të P

Rrjedhim 2.3 Çdo polinom real njëvariabëlsh P i shkallës $n \geq 1$ mund të faktorizohet si $P = (x - z_1)(x - z_2) \dots (x - z_m)g(x)$ ku $z_1, z_2, \dots, z_m$ janë rrënjët reale të P, $m \leq n$ dhe $g(x)$ është i pazbërthyeshëm në $\mathbb{R}$.

Implikimi i *Rrjedhim 2.3* është se ne mund të fillojmë duke gjetur ndonjë rrënjë α të $P(x)$, pastaj pjesëtojmë $P(x)$ me faktorin $(x-\alpha)$ për të fituar $P_1(x)$ i tillë që $P(x) = (x-\alpha)P_1(x)$ dhe përsëritni procesin për $P_1(x)$ derisa të arrijmë në $P_m(x)$ të tillë që të jetë i pazbërthyeshëm në $\mathbb{R}$ (d.m.th. të mos ketë rrënjë reale). Për shkak se pas çdo hapi pjesëtojmë me një faktor linear, mund të shfrytëzojmë një teknikë efikase për pjesëtim që njihet si Metoda e Horner-it e përshkruar në [12].

Përparësitë e algoritmit të metodës së Horner-it janë efikasiteti i pjesëtimit dhe thjeshtësia, të cilat në kombinim me metodën e Newton-it rezultojnë në një algoritëm të shpejtë. Nga ana tjetër, pjesëtimi në secilin iteracion prodhon një gabim vogël në precizion, që pas disa iteracioneve mund të shkaktojë mospërputhje më të madhe në rrënjët mëfundme sesa që dëshironim.

Në qoftë se kërkohet vlera e polinomit në një pikë x ajo mund të njehsohet drejtpërdrejtë duke shfrytëzuar metodën e tangjentës (Metoda e Newton-it për përafrimin e rrënjëve) [1], duke kryer $2n-1$ veprime shumëzimi dhe n veprime mbledhjeje gjithsej $3n-1$ veprime. Për të pakësuar numrin e veprimeve, gjë e cila ka rëndësi sepse vlera e polinomit njehsohet shumë herë, $P_n(x)$ vihet në trajtën:

$$P_n(x) = (\dots(a_0x + a_1)x + a_2)x + \cdots + a_{n-1})x + a_n$$

Po të shënojmë me $b_0 = a_0$ dhe $b_1, b_2, \dots b_{n-1}$ përkatësisht kllapat e vogla, nga më e brendshmja tek më e jashtmja, kemi

$$b_0 = a_0$$

$$b_1 = a_1 + b_0x$$

$$\dots$$

$$b_n = a_n + b_{n-1}x$$

dhe b_n-ja jep vlerën e polinomit në pikën x. Për përdorim praktik algoritmi i mësipërm paraqitet në trajtën:

$$b_0 = a_0$$

$$b_k = a_k + b_{k-1}x \quad \text{për } \mathrm{k} = 1,2, \dots, \mathrm{n}$$

Ky algoritëm quhet edhe skema e Horner-it, për njehsimin e vlerës së polinomit me këtë metodë duhen kryer n shumëzime dhe n mbledhje, pra gjithsej $2n$ veprime që është $\frac{2}{3}$ e numrit të veprimeve të kryera me metodën klasike.

Skema e Horner-it, mund të përdoret edhe për pjesëtimin e një polinomi me një faktor linear $x - a$. Le ta zëmë se (1.1) pjesëtohet me $x - a$ atëherë

$$P_n(x) = (x - a)Q_{n-1}(x) + R$$

ku $Q_{n-1}(x)$ është polinom i fuqisë $n - 1$ dhe R konstante.

Në qoftë se shënojmë $b_0, b_1, \dots, b_{n-1}$ koeficientët e $Q_{n-1}(x)$ atëherë

$$a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n = (x - a)(b_0x^{n-1} + b_1x^{n-2} + \cdots + b_{n-2}x + b_{n-1}) + R$$

Barazojmë koeficientët pranë kufizave me fuqi të njëjtë

$$a_0 = b_0$$

$$a_1 = b_1 - b_0a$$

$$\dots$$

$$a_{n-1} = b_{n-1} - b_{n-2}a$$

$$a_n = R - b_{n-1}a$$

Shënojmë tani $R = b_n$, dhe nxjerrim b_i në funksion të a_i fitohet përsëri skema e mësipërme e Horner-it për njehsimin e $b_k(k = 1,2, \dots, n - 1)$ dhe R-së.

2.4 Metoda e Müller-it

Kjo metodë është në një farë kuptimi, përgjithësimi i asaj të prerëses sepse vargu i përafrimeve të rrënjëve ndërtohet duke marrë prerjen e një parabole që kalon nga tri pika me boshtin e x-ve, në vend të prerëses që kalonte në dy pika.

Supozojmë se njihen tri pika të vijës $y = P(x)$

$$(x_{i-2}, P_{i-2}), (x_{i-1}, P_{i-1}) \text{ dhe } (x_i, P_i)$$

ku $P_i = P(x_i)$ për $i = 1,2, \ldots$ dhe $P(x)$ është polinomi (1.1) por me koeficient ndoshta edhe kompleksë.

Le të jetë

$$y = ax^2 + bx + c \tag{2.1}$$

ekuacioni i parabolës që kalon nëpër tri pikat e dhëna. Atëherë kanë ven barazimet

$$\begin{aligned} ax_{i-2}^2 + bx_{i-2}^2 + c &= P_{i-2} \\ ax_{i-1}^2 + bx_{i-1}^2 + c &= P_{i-1} \\ ax_i^2 + bx_i^2 + c &= P_i \end{aligned} \tag{2.2}$$

Ky sistem ekuacionesh ka një zgjidhje të vetme, sepse përcaktori i tij (determinanta) është e tipit të Vandermond-it pra i ndryshëm nga zero

$$Det = (x_{i-1} - x_{i-2})(x_i - x_{i-2})(x_{i-1} - x_i) \neq 0$$

Duke zgjidhur (2.2) me rregullat e Kramer-it, gjejmë:

$$a = \frac{\left((x_{i-1} - x_i)P_{i-2} + (x_i - x_{i-2})P_{i-1} + (x_{i-2} - x_{i-1})P_i\right)}{Det}$$

$$b = \frac{\left((x_i^2 - x_{i-1}^2)P_{i-1} + (x_{i-2}^2 - x_i^2)P_{i-1} + (x_{i-1}^2 - x_{i-2}^2)P_i\right)}{Det}$$

$$c = \frac{\left((x_{i-1}^2 - x_i^2 x_{i-1})P_{i-2} + (x_i^2 x_{i-2} - x_i x_{i-2}^2)P_{i-1} + (x_{i-2}^2 x_{i-1} - x_{i-1}^2 - x_{i-1}^2 x_{i-2})P_i\right)}{Det}$$

Zëvendësojmë a, b, c tek (2.1)

$$\begin{aligned} y = \frac{(x - x_i)(x - x_{i-1})}{(x_{i-1} - x_{i-2})(x_i - x_{i-2})} P_{i-2} + \frac{(x - x_{i-2})(x - x_i)}{(x_{i-1} - x_{i-2})(x_{i-1} - x_i)} P_{i-1} \\ + \frac{(x - x_{i-2})(x - x_{i-1})}{(x_i - x_{i-2})(x_i - x_{i-1})} P_i \end{aligned} \tag{2.3}$$

Bëjmë këto shënime me u dhe q:

$$u = \frac{(x - x_i)}{(x_i - x_{i-1})}$$

$$q = \frac{(x_i - x_{i-1})}{(x_{i-1} - x_{i-2})}$$

që nga

$$1 + u = \frac{(x - x_{i-1})}{(x_i - x_{i-1})}$$

$$1 + q = \frac{(x_i - x_{i-2})}{(x_{i-1} - x_{i-2})}$$

Dhe më tej

$$\frac{(x - x_i)(x - x_{i-1})}{(x_{i-1} - x_{i-2})(x_i - x_{i-2})} = \frac{u(1+u)q^2}{1+q}$$

$$\frac{(x - x_{i-2})(x - x_i)}{(x_{i-1} - x_{i-2})(x_{i-1} - x_i)} = -u(1 + q + uq) \qquad (2.4)$$

$$\frac{(x - x_{i-2})(x - x_{i-1})}{(x_i - x_{i-2})(x_i - x_{i-1})} = (1+u)\frac{1 + q + uq}{1+q}$$

I zëvendësojmë shprehjet (2.4) tek ekuacioni (2.3)

$$y = u^2(q^2 P_{i-2} - q(1-q)P_{i-1} + (1+2q)P_i) + u(q^2 P_{i-2} - (1+q)^2 P_{i-1} + (1+2q)P_i) + (1+q)P_i$$

Si pikë të re x_{i+1} do marrim prerjen e kësaj parabole me boshtin Ox.

Duke shënuar atëherë me

$$A = (q^2 P_{i-2} - q(1-q)P_{i-1} + (1+2q)P_i)$$

$$B = (q^2 P_{i-2} - (1+q)^2 P_{i-1} + (1+2q)P_i)$$

$$C = (1+q)P_i$$

merret metoda e Müller-it ose ndryshe e njohur si metoda e parabolave:

$$x_{i+1} = x_i - (x_i - x_{i-1})\frac{2C}{B \pm \sqrt{B^2 - 4AC}} \qquad (2.5)$$

Shenja pranë rrënjës katrore zgjidhet në mënyrë të tillë që emëruesi të ketë verë absolute më të madhe. Kjo zgjedhje shmang gabimin e fshirjes dhe siguron që x_{i+1} të jetë më afër vlerës x_i. Procesi vazhdon duke ndërtuar një parabolë, tani me anë të pikave x_{i-1}, x_i, x_{i+1} dhe vlerave përkatëse të polinomit (1.1). Duket që metoda e Müller-it është trehapëshe dhe kërkon tre vlera fillestare për të filluar. Zakonisht si të tilla merren $x_0 = -1$,$x_1 = 1$ dhe $x_2 = 0$ dhe vlerat përkatëse të polinomit (1.1) d.m.th.

$$P(x_0) = a_{n-2} - a_{n-1} + a_n, P(x_1) = a_{n-2} + a_{n-1} + a_n, P(x_2) = a_n$$

Kësaj zgjedhjeje i përgjigjet parabola prerëse:

$$P_2(x) = a_{n-2}x^2 + a_{n-1}x + a_n$$

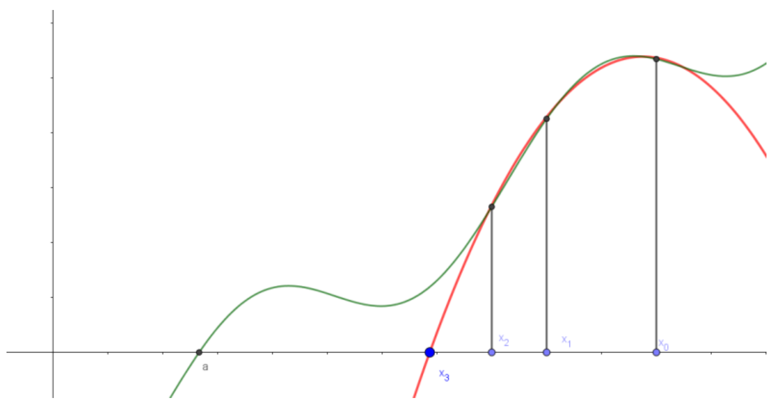

Figura 2.3: Metoda e Müller-it

Metoda e Müller-it është e përshtatshme për polinome me shkallë të lartë, por ajo mund të përdoret me sukses edhe për funksione që nuk janë polinome, në rastin kor derivatet njehsohen me vështirësi, mbasi këto të fundit nuk ndërhyjnë në formulën (2.5).

2.5 Metoda e Newton-it për lokalizimin e rrënjëve

Bazë e kësaj metode është pohimi i mëposhtëm:

Pohim 2.1: Në qoftë se për një vlerë $x = a$, polinomi $P(x)$ dhe të gjithë derivatet e tij $P'(x), P''(x), \dots, P^{(n)}(x)$ janë pozitive atëherë numri $x = a$ është kufi i sipërm i rrënjëve pozitive të polinomit $P(x)$.

Për të caktuar kufirin e poshtëm të rrënjëve pozitive të polinomit $P(x)$ e shfrytëzojmë zëvendësimin $x = \frac{1}{y}$, nga ku fitojmë

$$P(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n$$

$$P\left(\frac{1}{y}\right) = a_0 \left(\frac{1}{y}\right)^n + a_1 \left(\frac{1}{y}\right)^{n-1} + \cdots + a_{n-1}\left(\frac{1}{y}\right) + a_n \qquad /\cdot y^n$$

$$y^n \cdot f\left(\frac{1}{y}\right) = a_0 + a_1 y + \cdots + a_{n-1} y^{n-1} + a_n y^n = Q(y)$$

Në qoftë se b është kufi i sipërm i rrënjëve pozitive të polinomit $Q(y)$ atëherë $\frac{1}{b}$ është kufi i poshtëm i rrënjës pozitive të polinomit $P(x)$ pra rrënjët pozitive do të shtrihen në intervalin $\left(\frac{1}{b}, a\right)$.

Nëse në polinomin $P(x)$ zëvendësojmë $x = -\frac{1}{z}$ atëherë fitojmë polinomin

$$h(z) = z^n \cdot P\left(-\frac{1}{z}\right)$$

Në qoftë se d është kufi i sipërm i rrënjëve reale pozitive të polinomit $h(z)$ atëherë $-\frac{1}{d}$ është kufi i sipërm i rrënjës negative të polinomit $P(x)$.

Nëse në polinomin $P(x)$ zëvendësojmë $x = -u$ atëherë fitojmë polinomin $k(u)$. Në qoftë se c është kufi i sipërm i rrënjëve reale pozitive të polinomit $k(u)$ atëherë $-c$ është kufi i poshtëm i rrënjës negative të polinomit $P(x)$. Kështu rrënjët negative do të shtrihen në intervalin $(-c, -\frac{1}{d})$.

Shembull 2.3 Të lokalizohen rrënjët e ekuacionit $P(x) = x^3 + 2x^2 - 5x - 6$ me metodën e Newton-it.

Zgjidhje:

Gjejmë derivatet e $f(x)$

$$P'(x) = 3x^2 + 4x - 5$$

$$P''(x) = 6x + 4$$

$$P'''(x) = 6$$

Gjejmë një vlerë të x për të cilën do të vlejë $P(x) > 0$, $P'(x) > 0$, $P''(x) > 0$ dhe $P'''(x) > 0$.

Për $x = 1$ kemi $P(1) < 0$

Për $x = 2$ kemi $P(2) = 0$

Për $x = 3$ kemi $P(3) > 0$, $P'(3) > 0$, $P''(3) > 0$ dhe $P'''(3) > 0$, nga rrjedhë se $x = a = 3$, pra kufi i sipërm i rrënjëve reale pozitive të polinomit $P(x)$ është 3.

Për të caktuar kufirin e poshtëm të rrënjëve pozitive të polinomit $f(x)$ zëvendësojmë $x = \frac{1}{y}$ pra:

$$P\left(\frac{1}{y}\right) = \left(\frac{1}{y}\right)^3 + 2\left(\frac{1}{y}\right)^2 - 5\left(\frac{1}{y}\right) - 6 \qquad /\cdot y^3$$

$$y^3 P\left(\frac{1}{y}\right) = 1 + 2y - 5y^2 - 6y^3 \qquad /\cdot(-1)$$

$$-y^3 P\left(\frac{1}{y}\right) = 6y^3 + 5y^2 - 2y - 1 = Q(y)$$

$$Q(y) = 6y^3 + 5y^2 - 2y - 1$$

Gjejmë derivatet e $Q(y)$

$$Q'(y) = 18y^2 + 10y - 2$$

$$Q''(y) = 36y + 10$$

$$Q'''(y) = 36$$

Për $y = 1$ kemi $Q(1) > 0$, $Q'(1) > 0$, $Q''(1) > 0$ dhe $Q'''(1) > 0$ nga rrjedhë se $y = b = 1$, pra kufi i poshtëm i rrënjëve reale pozitive të polinomit $P(x)$ është $\frac{1}{b} = \frac{1}{1} = 1$.

Pra rrënjët pozitive do të shtrihen në intervalin $\left(\frac{1}{b}, a\right) = (1,3)$.

Për të gjetur rrënjët e sipërme negative të polinomit $P(x)$ së pari duhet të marrim zëvendësimin $x = -\frac{1}{z}$ dhe kemi

$$P\left(-\frac{1}{z}\right) = \left(-\frac{1}{z}\right)^3 + 2\left(-\frac{1}{z}\right)^2 - 5\left(-\frac{1}{z}\right) - 6$$

$$P\left(-\frac{1}{z}\right) = -\frac{1}{z^3} + \frac{2}{z^2} + \frac{5}{z} - 6 \qquad / \cdot \ (-z^3)$$

$$-z^3 \cdot P\left(-\frac{1}{z}\right) = 6z^3 - 5z^2 - 2z + 1 = h(z)$$

$$h(z) = 6z^3 - 5z^2 - 2z + 1$$

Gjejmë derivatet e $h(z)$

$$h'(z) = 18z^2 - 10z - 2$$

$$h''(z) = 36z - 10$$

$$h'''(z) = 36$$

Për $z = 1$ kemi $h(1) = 0$

Për $z = 2$ kemi $h(2) > 0$, $h'(2) > 0$, $h''(2) > 0$ dhe $h'''(2) > 0$ nga rrjedhë se $z = d = 2$, pra kufi i sipërm i rrënjëve reale negative të polinomit $P(x)$ është $-\frac{1}{d} = -\frac{1}{2}$.

Për të gjetur rrënjët e poshtëm negative të polinomit $P(x)$ së pari duhet të marim zavendësimin $x = -u$ dhe kemi

$$P(-u) = (-u)^3 + 2(-u)^2 - 5(-u) - 6$$

$$P(-u) = -u^3 + 2u^2 + 5u - 6 \qquad /\cdot\ (-1)$$

$$P(-u) = u^3 - 2u^2 - 5u + 6 = k(u)$$

$$k(u) = u^3 - 2u^2 - 5u + 6$$

Gjejmë derivatet e $k(u)$

$$k'(u) = 3u^2 - 4u - 5$$

$$k''(u) = 6u - 4$$

$$k'''(u) = 6$$

Për $u = 1$ kemi $k(1) = 0$

Për $u = 2$ kemi $k(2) < 0$

Për $u = 3$ kemi $k(3) = 0$

Për $u = 4$ kemi $k(4) > 0$, $k'(4) > 0$, $k''(4) > 0$ dhe $k'''(4) > 0$ nga rrjedhë se $u = c = 4$. Në qoftë se c është kufi i sipërm i rrënjëve reale pozitive të polinomit $k(u)$ atëherë $-c = -4$ është kufi i poshtëm i rrënjës negative të polinomit $P(x)$

Kështu rrënjët negative do të shtrihen në intervalin $\left(-4, -\frac{1}{2}\right)$. (shih *Shembull 4.5*)

2.6 Rregulla e Descartes-it për shenjat

Rregulli i Descartes-it përdoret për të përcaktuar numrin e zerove reale të një funksioni polinom. Kjo rregull na tregon se numri i zero pozitive reale në një funksion polinomial $P(x)$, që është i njëjtë me numrin e ndryshimeve të shenjave të koeficientëve ose më i vogël se ajo vlerë për një numër çift. Numri i zerove reale negative të $P(x)$ është i barabartë me numrin e ndryshimeve në shenjën e koeficientëve të termave të $P(-x)$ ose më pak se kjo për një numër çift.

3. Metoda e Sturm-it dhe Vincent-it

Në këtë pjesë, do të paraqesim tre algoritme që synojnë të izolojnë rrënjët në intervale më të vogla të ndara, secili përmban saktësisht një rrënjë. Pastaj, duke përdorur kombinimin e algoritmeve të paraqitur në kapitullin e mëparshëm, do të përafrojmë rrënjët me tolerancën e lejuar, duke arritur kështu rezultatin e dëshiruar.

3.1 Metoda e Sturm-it

Përkufizim 3.1 Zinxhir i Sturm-it apo varg i Sturm-it i një polinomi P është një varg i fundëm i polinomeve $P_0, P_1, \dots P_m$ me shkallë rënëse, me veti:

1. $P = P_0$
2. P ka rrënjë të thjeshtë(pra të njëfishtë)
3. Nëse $P(\alpha) = 0$ atëherë $sgn(P_1(\alpha\,)) = sgn\big(P'(\alpha)\big)$
4. $\forall 0 < i < m,$ nëse $f_i(\alpha) = 0$ atëherë $sgn\big(P_{i-1}(\alpha)\big) = -sgn\big(P_{i+1}(\alpha)\big)$
5. $sgn\big(P_m(x)\big)$ është konstante për çdo $x \in \mathbb{R}$.

Vargu i Sturm-it mund të krijohet duke e shfrytëzuar algoritmin e Euklid-it për $P(x)$ dhe $P'(x)$ si më poshtë

$$P_0(x) = P(x)$$

$$P_1(x) = P'(x)$$

$$P_2(x) = -r(P_0, P_1) = P_1(x)q_0(x) - P_0(x)$$

$$P_2(x) = -r(P_1, P_2) = P_2(x)q_1(x) - P_1(x)$$

$$\vdots$$

$$P_m(x) = -r(P_{m-2}, P_{m-1}) = P_{m-1}(x)q_{m-2}(x) - P_{m-2}(x)$$

$$0 = -r(P_{m-1}, P_m)$$

ku $r(P_i, P_{i+1})$ paraqet mbetjen dhe $q_i(x)$ paraqet herësin gjatë pjesëtimit të polinomeve $P_i(x) = q_i(x)P_{i+1}(x) + r(P_i, P_{i+1})$ për çdo $0 \leq i < m$. Pasi që shkalla më e madhe $\deg\big(r(P_i, P_{i+1})\big) \leq \deg(P_{i+1}) - 1$, kemi që gjatësia maksimale e vargjeve të Sturm-it $m \leq \deg(P_0)$.

Përkufizim 3.2 Le të jetë $\lambda_0, \lambda_1, \lambda_2, \ldots$ një varg real i fundëm apo i pafundëm. Supozojmë që $a < b; a, b \in \mathbb{N}$ dhe kushtet e mëposhtme mbajnë

$$sgn(\lambda_a) = -sgn(\lambda_b)$$

dhe

$$\forall i, a < i < b: \lambda_i = 0$$

quhet një ndryshim shenje mes a dhe b.

Për ta formuluar teoremën e Sturm-it le të jetë $\sigma(P, \zeta)$ funksion i cili paraqet numrin e ndryshimeve të shenjave në vargun $P_0(\zeta), P_1(\zeta), \ldots, P_m(\zeta)$ ku $P_0, P_1, \ldots, P_m$ është vargu i Sturm-it për P.

Teoremë 3.1 (Sturm *[13]*) Le të jetë $P(x)$ një polinom real pa rrënjë të shumëfishta dhe $a, b \in \mathbb{R}$ ashtu që $a < b, P(a) \neq 0, P(b) \neq 0$ atëherë numri i rrënjëve reale $(a, b]$ është i barabartë me $\sigma(P, a) - \sigma(P, b)$.

Shënim: Teorema mund të përforcohet (si në [14]) për çdo polinom, përfshirë e dhe ato me rrënjë të shumëfishta. Atëherë përsëri numri i rrënjëve $\sigma(P, a) - \sigma(P, b)$ na jep numrin e rrënjëve të ndryshme që shtrihen në intervalin $(a, b]$.

Duke e shfrytëzuar *Teoremë 3.1* me metodën e Newton-it dhe përgjysmimit mund të krijojmë një algoritëm. (shih *Shembull 4.6.*)

```
funksion Sturm(P, P′, a, b)
    σ_a ← 0
    σ_b ← 0
    P_0 ← P
    P_1 ← P′
    nëse sgn(P_0(a)) ≠ sgn(P_1(a)) atëherë
        σ_a ← σ_a + 1
    nëse sgn(P_0(b)) ≠ sgn(P_1(b)) atëherë
        σ_b ← σ_b + 1
    derisa P_1 ≠ 0
        r = P_0 mod P_1
        P_0 = P_1
        P_1 = −r
        nëse sgn(P_0(a)) ≠ sgn(P_1(a)) atëherë
            σ_a ← σ_a + 1
        nëse sgn(P_0(b)) ≠ sgn(P_1(b)) atëherë
            σ_b ← σ_b + 1
    kthe σ_a − σ_b
```

Figura 3.1: Algoritmi i Sturm-it

Gjithashtu kemi edhe një algoritëm (të Sturm-it) për gjetjen e rrënjëve

```
funksion Gjetja_e_rrënjëve_Sturm(P, P′, a, b, ε, max)
    rrënjët ← ∅
    numërimi i rrënjëve ← Sturm(P, P′, a, b)
    nëse numërimi i rrënjëve > 0 atëherë
        nëse numërimi i rrënjëve = 1 atëherë
            error, rrënjë ← Newton(P, P′, a, b, ε, max)
            nëse error = 0 atëherë
                rrënjët ← rrënjët ∪ rrënjë
                kthe rrënjët
        c = (a+b)/2
        rrënjët ← rrënjët ∪ Gjetja_e_rrënjëve_Sturm(P, P′, a, c, ε, max)
        rrënjët ← rrënjët ∪ Gjetja_e_rrënjëve_Sturm(P, P′, c, b, ε, max)
    kthe rrënjët
```

Figura 3.2: Algoritmi i Sturm-it (gjetja e rrënjëve)

(shih *Shembull 4.7.*)

Algoritmi ka disa disavantazhe. Dobësia kryesore e këtij algoritmi është kompleksiteti kohor i llogaritjes së polinomeve të vargut të Sturm-it në pikat a dhe b për të llogaritur numrin e ndryshimit të shenjave. Meqenëse mund të ketë deri në n polinome në varg, ku n është shkalla e P, kjo mund të rezultojë deri në $O(n^2)$ shumëzime dhe mbledhje në çdo iteracion të përgjysmores.

Një çështje tjetër është se gjatë krijimit të vargut të Sturm-it, në varësisht nga zbatimi, nëse kemi një gabim gjatë pjesëtimit të polinomeve, kjo do të na krijonte efektin e topit të borës në pjesëtimet pasuese, gjë që mund të rezultojë në një numër të ndryshëm të polinomeve në vargun e llogaritur sesa numri aktual i polinomeve në varg. Kjo mund të ndodhë nëse koeficientët paraqiten si floating-points me saktësi të caktuar, në vend që t'i përfaqësojnë ato si thyesa të numrave të plotë (e cila shpesh është jopraktike). Kjo mund të rezultojë që numri i ndryshimeve të shenjave $\sigma(\zeta)$të jetë i pasaktë, duke çuar në numrin e rrënjëve të vendosura në interval (a, b) të jetë i gabuar dhe të shkaktojë që disa rrënjë të mos gjenden. Megjithatë, kjo mund të shmanget duke zvogëluar tolerancën e gabimit në mënyrë të mjaftueshme.

Metoda e Sturm-it merret si metodë teorikisht e përkryer për përcaktimin e numrit të rrënjëve reale dhe izolimin e tyre nëpër interval.

Shembull 3.1 Me anë të metodës së Sturm-it të caktohet numri i rrënjëve reale dhe të veçohen ato në interval me gjatësi jo më të madhe se 1, për polinomin $P(x) = x^3 + x^2 - 2x - 1$.

Zgjidhje: Formojmë vargun e funksioneve të Sturm-it $P(x)$, $P'(x)$, $P_1(x)$ dhe $P_2(x)$.

$P(x) = x^3 + x^2 - 2x - 1$

$P'(x) = 3x^2 + 2x - 2$

Në bazë të algoritmit të Euklidit kemi:

$P_1(x) = 2x - 1$

$P_2(x) = 1$

Me metodën e Newton-it gjejmë se rrënjët reale të polinomit $P(x) = x^3 + x^2 - 2x - 1$ ndodhen në intervalet $\left(-2, -\frac{1}{3}\right)$ dhe $(1,2)$ d.m.th. rrënjët gjenden në intervalin $(-2,2)$.

Për të caktuar numrin e rrënjëve reale të polinomit në këtë interval përpilojmë Tabela 3.1.

Nga Tabela 3.1 konstatojmë se polinomi i dhënë i ka të gjitha rrënjët reale (sepse $\sigma(-2) - \sigma(2) = 3$).

Tabela 3.1: Tabela e Sturm-it për $P(x) = x^3 + x^2 - 2x - 1$

$\boldsymbol{x}$	$\boldsymbol{P(x)}$	$\boldsymbol{P'(x)}$	$\boldsymbol{P_1(x)}$	$\boldsymbol{P_2(x)}$	$\boldsymbol{W(x)}$	$\boldsymbol{\sigma(-2) - \sigma(2)}$
$\boldsymbol{-2}$	-	+	-	+	3	3
2	+	+	+	+	0	

Me qëllim që t'i veçojmë rrënjët në intervalet e veçanta, tabelën e më sipërme e zgjerojmë në këtë mënyrë:

Tabela 3.2: Tabela e zgjeruar e Sturm-it për $P(x) = x^3 + x^2 - 2x - 1$

$\boldsymbol{x}$	$\boldsymbol{P(x)}$	$\boldsymbol{P'(x)}$	$\boldsymbol{P_1(x)}$	$\boldsymbol{P_2(x)}$	$\boldsymbol{\sigma(x)}$	$\boldsymbol{\sigma}$ $(\boldsymbol{x_k})$- $\boldsymbol{\sigma}$ $(\boldsymbol{x_{k+1}})$
$\boldsymbol{-2}$	$-$	$+$	$-$	$+$	3	$3 - 2 = 1$
$\boldsymbol{-1}$	$+$	$-$	$-$	$+$	2	$2 - 1 = 1$
0	$-$	$-$	$-$	$+$	1	$1 - 1 = 0$
1	$-$	$+$	$+$	$+$	1	$1 - 0 = 1$
2	$+$	$+$	$+$	$+$	0	

Nga shtylla e fundit e kësaj tabele shohim se polinomi i ka rrënjët në intervalet $(-2, -1)$, $(-1,0)$ dhe $(1,2)$.

3.2 Metoda e Vincent-it

Vincent-i e botoi teoremën e tij në shekullin e XIX, por për shkak të publikimit të teoremës së Sturm-it, ajo u harrua deri në fund të shekullit të XX, kur u soll përsëri nga. Kjo çoi në krijimin e një versioni të dytë të teoremës në.

Teoremë 3.2 (e Vincent-it, versioni me thyesa të vazhdueshme *[15] [16]*) Le të jetë $P(x)$ një polinom me koeficient racional dhe pa rrënjë të shumëfishta në qoftë se i o kryejmë këto zëvendësime

$$x \leftarrow \alpha_1 + \frac{1}{x}, x \leftarrow \alpha_2 + \frac{1}{x}, x \leftarrow \alpha_3 + \frac{1}{x}, \dots$$

ku $\alpha_1 \geq 0$ është një numër i plotë jonegativ arbitrar, dhe $\alpha_2, \alpha_3, \dots$ janë numra pozitiv arbitrar $\alpha_i > 0, i > 1$ atëherë polinomi i fituar ose nuk ka asnjë ndryshim shenje (shih *Përkufizim 3.2*) ose vetëm një ndryshim shenje. Në rastin e parë s'ka asnjë rrënjë pozitive kurse në rastin e dytë ka vetëm një rrënjë pozitive, që si thyesë e vazhdueshme paraqitet

$$\alpha_1 + \cfrac{1}{\alpha_2 + \cfrac{1}{\alpha_3 + \cfrac{1}{\ddots}}}$$

Rrënjët negative mund të gjinde në të njëjtën mënyre thjeshtë duke marrë zëvendësimin $x \leftarrow -x$ në polinomin P. Kushti që polinomi nuk duhet të ketë rrënjë të shumëfishta nuk e kufizon gjeneralitetin e teoremës, në raste se polinomi ka rrënjë të shumëfishta mund ta pjesëtojmë P me $P.M.P(P,P')$ që do ti sillte të gjitha rrënjët e shumëfishta në rrënjë të njëfishta dhe pastaj t'i izolojmë rrënjët.

Teoremë 3.3 (Teorema e Vincent-it(me përgjysmim) *[17]*) Le të jetë $P(x)$ një polinom real i shkallës n që ka vetëm rrënjë të thjeshta(pra të njëfishta). Është e mundur që të përcaktojmë një $\delta > 0$ ashtu që për çdo çift numrash real a, b ku $|b-a| < \delta$, çdo polinom i transformuar në formën

$$\phi(x) = (1+x)^n P\left(\frac{a+bx}{1+x}\right)$$

ka saktësisht zero ose një ndryshim shenje. Rasti i dytë është i mundur nëse dhe vetëm nëse $P(x)$ ka rrënjë të vetme në intervalin (a,b).

Ngjashëm si me metoda e mësipërme, rrënjët negative mund ti fitojmë nëse e marrim zëvendësimin $x \leftarrow -x$, kurse problemin me rrënjët e shumëfishta mund ta zgjedhim si paraprakisht në raste se polinomi ka rrënjë të shumëfishta mund ta pjesëtojmë P me $P.M.P(P,P')$ që do ti sillte të gjitha rrënjët e shumëfishta në rrënjë të njëfishta dhe pastaj t'i izolojmë rrënjët.

Duke shfrytëzuar këtë teoremë mund të krijojmë një test për kufirin e sipërm të rrënjëve pozitive brenda intervalit (a,b)

$$\sigma_{(a,b)}(P) = \sigma\left((1+x)^n P\left(\frac{ax+b}{1+x}\right)\right) = \sigma_{(b,a)}(P) \qquad (3.1)$$

ku $\sigma(\zeta)$ është numri i ndryshimeve të shenjave të vargut të numrave $a_0, a_1, \dots a_n$ në ζ.

Kemi dy algoritme të bazuar në versione të ndryshme të kësaj teoreme, siç janë të përshkruar në vazhdim.

3.2.1 Metoda e Vincent-Akritas-Strzebonski (VAS)

Transformimi i Möbius-it

$$M(x) = \frac{ax + b}{cx + d}, a, b, c, d \in \mathbb{N}, ad - bc \neq 0$$

paraqet thyesën e vazhdueshme të fundme që zëvendëson x në $P(x)$, ashtu që zëvendësimi rezulton në polinomin e transformuar

$$Q(x) = (cx + d)^{\deg P} P\left(\frac{ax + b}{cx + d}\right)$$

që ka vetëm një ndryshim shenje. Atëherë rrënja pozitive e $P(x)$ korrespondon me rrënjën e $Q(x)$ në $\left(\min\left(\frac{a}{c}, \frac{b}{d}\right), \max\left(\frac{a}{c}, \frac{b}{d}\right)\right)$. Kështu, për të izoluar rrënjët pozitive, duhet të njehsohen a, b, c, d tillë që ato të rezultojnë në polinom $Q(x)$ me një ndryshim shenje. Rregulli i Dekartit për shenjat, thotë se numri i rrënjëve pozitive të $P(x)$ është i barabartë me numrin e ndryshimeve të shenjave të koeficientëve $\sigma(P)$ ose më pak se ajo me një numër çift. Ne mund ta përdorim këtë test të thjeshtë për të përcaktuar nëse P ka saktësisht zero ose një rrënjë pozitive ose nëse ka më shumë. Nëse ka më shumë, ne së pari transformojmë polinomin duke zhvendosur rrënjë me te vogël pozitive më afër 0 dhe pastaj e ndajmë polinomin në dy polinome të transformuar, i pari përfaqëson intervalin midis $(0, 1)$ dhe i dyti përfaqëson intervalin $(1, \infty)$. Sa herë që kryejmë një zëvendësim në P, kryejmë zëvendësimin përkatës në M. Kjo na mundëson që në mënyrë sukseseive t'i njehsojmë intervalet në polinomin origjinal.

Kështu që algoritmi VAS na jep intervale të tillë që çdo interval përmban vetëm një rrënjë pozitive. Përsëri njëjtë si tek metodat e mësipërme, rrënjët negative do t'i izolojmë duke shfrytëzuar zëvendësimin $x \leftarrow -x$, pas kësaj mund ta ekzekutojmë edhe një herë algoritmin dhe në fund e gjejmë unionin e intervaleve. Kurse për përafrim të rrënjëve mund të shfrytëzojmë ndonjërën nga metodat e shënuara më sipër(p.sh metoda e Newton-it, përgjysmores) për çdo interval.

Kufiri i poshtëm i P (do ta shënojmë kp) mund të njehsohet duke llogaritur kufirin e sipërm (do ta shënojmë ks), $Q(x) = x^{\deg P} P\left(\frac{1}{x}\right)$ dhe pastaj $kp_P = \frac{1}{ks_Q}$ [18].

Nga [18] kemi edhe dy algoritme për kufij e sipërm

funksion $VAS\ (P, e)$
  nëse $\sigma(P) = 0$ **atëherë**
    kthe $\emptyset$
  nëse $\sigma(P) = 1$ **atëherë**
    $a \leftarrow \min(M(0), M(\infty))$ ku $M(\infty) = \frac{a}{c}$ nëse $c \neq 0$
    $b \leftarrow \max(M(0), M(\infty))$ dhe nëse $c = 0$ atëherë $M(\infty)$ është një kufi i sipërm i rrënjëve pozitive
    kthe (a, b)
  $kp \leftarrow$ një kufi i poshtëm i rrënjëve pozitive të P
  nëse $kp \geq 1$ **atëherë**
    $P \leftarrow P(x + kp)$
    $M \leftarrow M(x + kp)$
  $P_{01} \leftarrow (x+1)^{\deg P} P\left(\frac{1}{x+1}\right)$
  $M_{01} \leftarrow M\left(\frac{1}{x+1}\right)$
  $P_{1\infty} \leftarrow P(x+1)$
  $M_{1\infty} \leftarrow M(x+1)$
  $m = M(1)$
  nëse $P(1) \neq 0$ **atëherë**
    kthe $VAS(P_{01}, M_{01}) \cup VAS(P_{1\infty}, M_{1\infty})$
  përndryshe
    kthe $VAS(P_{01}, M_{01}) \cup VAS(P_{1\infty}, M_{1\infty}) \cup \{(m, m)\}$

Figura 3.3: Algoritmi VAS

3.2.2 Metoda e Vincent-Collins-Akritas (VCA)

Përderisa algoritmi i mëparshëm kërkonte llogaritjen e kufirit të sipërm dhe të poshtëm të rrënjëve pozitive, ky algoritëm lejon që të fusim intervalin e dëshiruar të kërkimit (a, b). Kjo nuk ishte e mundur në rastin e mëparshëm, pasi rregulli i Dekartit jep kufirin e sipërm e të gjitha rrënjët pozitive. Gjithsesi testi (3.1)(i njohur si testi i Budan-it), na lejon ta specifikojmë intervalin në të cilin jemi të interesuar t'i kërkojmë rrënjët.

Testi i Budan-it është një version i testit (3.1) ku $a = 0, b = 1$ nga kemi se

$$\sigma_{(0,1)}(P) = \sigma\left((1+x)^n P\left(\frac{x}{1+x}\right)\right)$$

Nëse $\sigma_{(0,1)}(P) = 0$ do të thotë se nuk kemi rrënjë në intervalin $(0,1)$, nëse $\sigma_{(0,1)}(P) = 1$ do të thotë se kemi saktësisht një rrënjë në intervalin $(0,1)$. Por për çdo vlerë më të madhe se 1, testi na jep kufirin e sipërm të numri të rrënjëve që gjenden në intervalin $(0,1)$ ashtu që numri i vërtetë i rrënjëve që gjenden në atë interval është i barabartë me $\sigma_{(0,1)}(P)$ ose është më pakë për ndonjë numër çift, d.m.th nëse kufiri i sipërm është çift ka mundës që të mos kemi asnjë rrënjë në intervalin $(0,1)$.

Atëherë hapi i parë që duet të bëjmë është që të sigurohemi që të gjitha rrënjët që duam ti izolojmë janë në intervalin (a, b). Së pari e marrim zëvendësimin $x \leftarrow x + a$ në P dhe fitojmë P_a, më pas zëvendësojmë $x \leftarrow x(b - a)$ në P_a për të fituar P_{ab} kjo na jep një bijeksion të intervalit (a, b) të P në intervalin $(0,1)$ të P_{ab}.

Më pas e aplikojmë testin e Budan-it për P_{ab}. Në qoftë se ka zero ose një rrënjë kërkimi ka mbaruar, përndryshe polinomin P_{ab} e ndajmë në dy polinome $P_{0\frac{1}{2}}$ dhe $P_{\frac{1}{2}1}$, ashtu që ka një bijeksion në mes të intervaleve $\left(0, \frac{1}{2}\right)$ dhe $\left(\frac{1}{2}, 1\right)$ të P_{ab} dhe intervaleve $(0,1)$ të dy polinomeve $P_{0\frac{1}{2}}$ dhe $P_{\frac{1}{2}1}$ respektivisht. Pastaj mund t'i shfrytëzojmë këto polinome në mënyre rekursive.

Nëse polinomi fillestar P ka rrënjë të shumëfishta, kjo, teorikisht, do të shkaktojë përsëritje pafundme të algoritmit, pasi testi i Budan-it do të kthejë gjithmonë një numër më të madh se një. Sidoqoftë, ekzistojnë 2 zgjidhje për këtë problem:

1. Njëjtë si te mënyrat e mësipërme, duhet t'i largojmë katrorët duke pjesëtuar P me $P.M.P(P, P')$, ose ndonjë metodë tjetër për mënjanimin e rrënjëve të shumëfishta.
2. Mund të caktojmë ndonjë tolerancë ε ashtu që $|a - b| < \varepsilon$, që do ta ndaloj algoritmin

Meqenëse zgjidhja e dytë është shumë më e thjeshtë dhe në përputhje me faktin se po i përafrojmë rrënjët me një precizitet të caktuar, ne e zbatojmë atë për të fituar algoritmin ë vazhdim. Nëse kufiri i sipërm është i tek, ekzistojnë ose rrënjë të shumëfishta ose një rrënjë e vetme me shumëfishitet tek. Nëse ka rrënjë të shumëfishta, atëherë kjo nuk ndikon në rezultat, pasi ato janë më afër njëra-tjetrës sesa toleranca e dëshiruar dhe kështu nuk jemi në gjendje t'i dallojmë ato. Nëse kufiri i sipërm është çift, atëherë nuk ka asnjë garancion se intervali përmban rrënjë. Për të provuar nëse ka një rrënjë, mund të provojmë të ekzekutojmë algoritmin e Newton-it në këtë interval. Sidoqoftë, nëse funksioni afrohet 0 shumë afër (d.m.th. më afër se ε), kjo mund të rezultojë të jetë e rreme.

Kështu, duhet të kemi kujdes gjatë përdorimit të këtij algoritmi në polinome me rrënjë të shumëfishta.

funksion $VCA\ (P, a, b, e)$

  $budan \leftarrow \sigma\left((1+x)^{\deg P} P\left(\frac{1}{1+x}\right)\right)$

  nëse $budan\ =\ 0$ **atëherë**

    kthe $\emptyset$

  nëse $budan\ =\ 1$ **atëherë**

    kthe $\{(a, b)\}$

  nëse $|a-b| < \varepsilon$ **atëherë**

    kthe $\{(a, b)\}$

  $P_{0\frac{1}{2}} \leftarrow 2^{\deg P} P\left(\frac{x}{2}\right)$

  $P_{\frac{1}{2}1} \leftarrow 2^{\deg P} P\left(\frac{x+1}{2}\right)$

  nëse $P\left(\frac{1}{2}\right) \neq 0$ **atëherë**

    kthe $VCA\left(P_{0\frac{1}{2}}, a, \frac{a+b}{2}, \varepsilon\right) \cup VCA\left(P_{\frac{1}{2},1}, \frac{a+b}{2}, b, \varepsilon\right)$

  përndryshe

    kthe $VCA\left(P_{0\frac{1}{2}}, a, \frac{a+b}{2}, \varepsilon\right) \cup VCA\left(P_{\frac{1}{2},1}, \frac{a+b}{2}, b, \varepsilon\right) \cup \{(\frac{a+b}{2}, \frac{a+b}{2})\}$

Figura 3.4: Algoritmi VCA

4. Implementimi i Matlab për izolimin dhe krahasimin e rrënjëve të polinomeve

Të gjitha metodat e paraqitura më sipër janë implementuar në Matlab.

4.1 Metoda e përgjysmimit

Shembull 4.1 Gjeni një rrënjë α të

$$P(x) = 2x^4 - 3x - 2$$

me precizitet $\varepsilon = 0.000001$

Zgjidhje: Më marrim një interval fillestar p.sh $1 < \alpha < 2$ pasi $P(1)P(2) = -3 \cdot 24 < 0$

Më poshtë kemi kodin në Matlab për metodën e Halley-it *Figura 4.4*

```
% metoda e pergjysmimit ne MATLAB
clear;

a =1;%= input('Jepe vleren e a=');
b =2;%= input('Jepe vleren e b=');
e =0.000001;%= input('Jepe vleren e tolerances e=');
numri_i_iteracioneve = 0;

[rrenja, numri_i_iteracioneve] = pergjysmorja(a, b, e,
numri_i_iteracioneve);
rrenja
numri_i_iteracioneve

function y = P(t)
    y = 2*t^4-3*t-2;
    return
end

function [x, numri_i_iteracioneve] = pergjysmorja(a, b, e,
numri_i_iteracioneve)
    x = (a+b)/2;
    numri_i_iteracioneve = numri_i_iteracioneve + 1;
    if (x-a)<=e
        return
    end
    if P(a)*P(x)<0
        [x, numri_i_iteracioneve] = pergjysmorja(a, x, e,
numri_i_iteracioneve);
        return
    else
```

```
        [x, numri_i_iteracioneve] = pergjysmorja(x, b, e,
numri_i_iteracioneve);
        return
    end
end
```

Figura 4.1: Metoda e përgjysmores në Matlab

e cila pas ekzekutimit fitojmë rezultatin e mëposhtëm, si në *Figura 4.2*

```
rrenja =

    1.3127


numri_i_iteracioneve =

    20
```

Figura 4.2: Rezultatet e metodës se përgjysmores

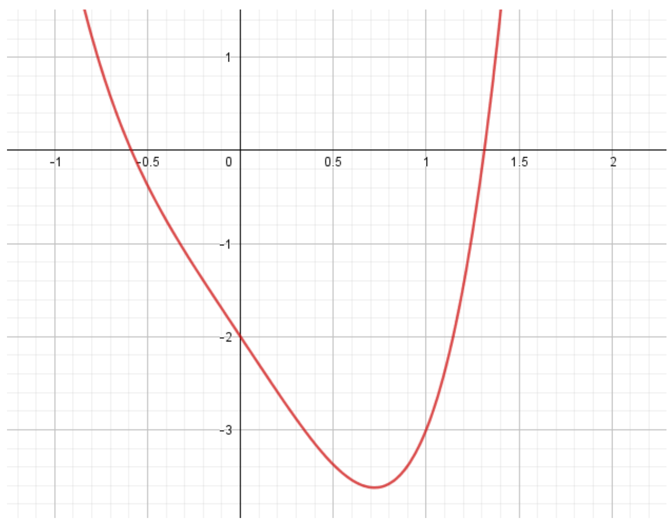

Figura 4.3: Paraqitja grafike e $P(x) = 2x^4 - 3x - 2$

4.2 Metoda e Newton-it për përafrimin e rrënjëve

Shembull 4.2 Gjeni një rrënjë α të

$$P(x) = 2x^4 - 3x - 2$$

me precizitet $\varepsilon = 0.000001$ duke shfrytëzuar metodën e Newton-it për përafrimin e rrënjëve.

Zgjidhje: Më marrim një interval fillestar p.sh $1 < \alpha < 2$, njëjtë si te metoda e përgjysmimit pasi $P(1)P(2) = -3 \cdot 24 < 0$

Më poshtë kemi kodin në Matlab për metodën e Halley-it Figura 4.4

```
% metoda e Newton-it ne MATLAB
clear;

a =1;%= input('Jepe vleren e a=');
b =2;%= input('Jepe vleren e b=');
e =0.000001;%= input('Jepe vleren e tolerances e=');
max = 20;

[err, x_0, numri_i_iteracioneve] = newton(a, b, e, max);
err;
x_0
numri_i_iteracioneve

function y = P(t)
    y = 2*t^4-3*t-2;
    return
end

function y = P_der(t)
    y = 8*t^3-3;
    return
end
function [err, x_1, numri_i_iteracioneve] = newton(a, b,
e, max)
    numri_i_iteracioneve=0;
    err = 1;
    x_0 = (a+b)/2;
    for i=1:max
        numri_i_iteracioneve = numri_i_iteracioneve + 1;
        emeruesi = P_der(x_0);
        if emeruesi == 0
            x_1 = x_0;
            break;
        end
        x_1 = x_0 - P(x_0)/emeruesi;
        if (abs((x_1-x_0)/x_1)<=e)&&(abs(x_1-x_0)<=e)
            err = 0;
            break;
        end
        if x_1<a || x_1>b
            err = 3;
            break;
        end
        x_0 = x_1;
    end
end
```

Figura 4.4: Metoda e Newton-it në Matlab

Pasi e ekzekutojmë kodin e mësipërm ftojmë rezultatin si në *Figura 4.5*

```
>> metoda_njutnit_kodi

x_0 =

    1.3127


numri_i_iteracioneve =

     5
```

Figura 4.5: Rezultati i metodës së Newton-it

Shih Figura 4.3 paraqitja grafike e funksionit $P(x) = 2x^4 - 3x - 2$

Me përafrimin e saktë që është $\alpha = 1.31265975467417$, gabimi pas iteracionet të fundit është $|\,\alpha - x_5\,| = 0.00000024532583$ që është brenda saktësisë së lejuar. Përderisa algoritmi i Newton-it zgjidh disa nga mangësitë e metodës së përgjysmimit, disa mbeten. Metoda ndonjëherë mund të dështojë (kur derivati i parë i x_n është zero) dhe konvergjenca për rrënjët e shumëfishta lineare.

Siç shihet qartë pas ekzekutimit të kodit metoda e Newton-it është shumë më e shpejtë se sa metoda e përgjysmimit pasi fituam saktësi të njëjtë për 5 iteracione kurse me metodën e përgjysmimit na nevojiteshin 20 iteracione. Kjo e bën metodën e Newton-it superiore pranë metodës së përgjysmimit në të shumtën e rasteve.

4.3 Metoda e Halley-it

Shembull 4.3 Gjeni një rrënjë α të

$$P(x) = 2x^4 - 3x - 2$$

me precizitet $\varepsilon = 0.000001$ duke shfrytëzuar metodën e Halley-it për përafrimin e rrënjëve.

Zgjidhje: Më marrim një interval fillestar p.sh $1 < \alpha < 2$, njëjtë si te metoda e përgjysmimit, dhe e Newton-it për ti krahasuar.

Më poshtë kemi kodin në Matlab për metodën e Halley-it *Figura 4.6*

```
% metoda e Halley-it ne MATLAB
clear;
clc;
a =1;%= input('Jepe vleren e a=');
b =2;%= input('Jepe vleren e b=');
e =0.000001;%= input('Jepe vleren e tolerances e=');
max = 20;

[err, x_0, numri_i_iteracioneve] = halley(a, b, e, max);
err;
x_0
numri_i_iteracioneve

function y = P(t)
    y = 2*t^4-3*t-2;
    return
end

function y = P_der(t)
    y = 8*t^3-3;
    return
end

function y = P_der2(t)
    y = 24*t^2;
    return
end

function [err, x_1, numri_i_iteracioneve] = halley(a, b,
e, max)
    numri_i_iteracioneve=0;
    err = 1;
    x_0 = (a+b)/2;
    for i=1:max
        numri_i_iteracioneve = numri_i_iteracioneve + 1;
        emeruesi = 2*(P_der(x_0))^2-P(x_0)*P_der2(x_0);
        if emeruesi == 0
            x_1 = x_0;
            break;
        end
        x_1 = x_0 - (2*P(x_0)*P_der(x_0))/emeruesi;
        if (abs((x_1-x_0)/x_1)<=e)&&(abs(x_1-x_0)<=e)
            err = 0;
            break;
        end
        if x_1<a || x_1>b
            err = 3;
```

```
                break;
            end
            x_0 = x_1;
        end
end
```

Figura 4.6: Metoda e Halley-it në Matlab

Pas ekzekutimit të kodit të mësipërm fitojmë rezultatin *Figura 4.7*

```
x_0 =

    1.3127


numri_i_iteracioneve =

     3
```

Figura 4.7: Rezultati i metodës së Halley-it

Shih Figura 4.3 paraqitja grafike e funksionit $P(x) = 2x^4 - 3x - 2$

Algoritmi përfundon me të njëjtin rezultat si rezultati i fituar nga algoritmi i përgjysmores dhe algoritmi i Newton-it ndërsa përfundon në vetëm 3 iteracione, për dallim nga 20 dhe 5 nga algoritmet e përgjysmores dhe Newton-it respektivisht. Algoritmi i Halley ka të njëjtat disavantazhe si algoritmi i Newton-it, ndërsa mundëson një shpejtësi më të lartë të konvergjencës, me koston e shtimit të kohës së llogaritjes pas çdo iteracioni. Nëse shkalla e funksionit polinomial është shumë e lartë, njehsimi i $P(x_n)$, $P'(x_n)$, $P''(x_n)$ merr një kohë të konsiderueshme. Kjo, e kombinuar me llogaritjet e tjera shtesë mund të rezultojë që koha totale e llogaritjes të jetë e ngjashme ose edhe më e madhe se te metoda e Newton-it. Për më tepër, nëse derivati i dytë është shumë afër 0, shkalla e konvergjencës është shumë e ngjashme me Newton-in. Kështu, nuk siguron avantazh të konsiderueshëm mbi metodën e Njutonit (dhe madje mund të rezultojë në më shumë kohë për llogaritje).

4.4 Metoda e Horner-it

Shembull 4.4 Gjeni një rrënjë α të

$$P(x) = 2x^4 - 3x - 2$$

me precizitet $\varepsilon = 0.000001$ duke shfrytëzuar metodën e Halley-it për përafrimin e rrënjëve.

Zgjidhje:

Më poshtë kemi kodin në Matlab për metodën e Horner-it *Figura 4.8*.

```
% metoda e Horner-it ne MATLAB
clear;

a =1;%= input('Jepe vleren e a=');
b =2;%= input('Jepe vleren e b=');
e =0.000001;%= input('Jepe vleren e tolerances e=');
max = 20;
rrenjet = [];
mbetja = 0;
error = 0;
P_0 = [2 0 0 -3 -2];

while mbetja == 0 && error == 0
    [error, rrenje] = newton(a, b, e, max);
    if error  == 0
        rrenjet = [rrenjet, rrenje];
        [P_1, mbetja] = horner(P_0, rrenje);
    end
    if P_1 == 1
        break
    end
    P_0 = P_1;
end

function y = P(t)
    y = 2*t^4-3*t-2;
    return
end

function y = P_der(t)
    y = 8*t^3-3;
    return
end

function [err, x_1] = newton(a, b, e, max)
    err = 1;
    x_0 = (a+b)/2;
    for i=1:max
        emeruesi = P_der(x_0);
        if emeruesi == 0
            x_1 = x_0;
            break;
        end
        x_1 = x_0 - f(x_0)/emeruesi;
        if (abs((x_1-x_0)/x_1)<=e)&&(abs(x_1-x_0)<=e)
            err = 0;
            break;
```

```
        end
        if x_1<a || x_1>b
            err = 3;
            break;
        end
        x_0 = x_1;
    end
end

function [P_1, mbetja] = horner(f_0, rrenja)
    n = length(P_0);
    rezultati = P_0(1)*ones(1,n);
    for j = 2:n
        rezultati(j) = rrenja * rezultati(j-1) + P_0(j);
    end
    mbetja = rezultati(n);
    P_1 = rezultati(1:n-1);
end
```

Figura 4.8: Kodi i metodës së Horner-it

Pas ekzekutimit të kodit të mësipërm fitojmë rezultatin *Figura 4.9*

```
>> horner_code

ans =

1.312659754674166
```

Figura 4.9: Rezultati i metodës së Horner-it

Shih Figura 4.3 paraqitja grafike e funksionit $P(x) = 2x^4 - 3x - 2$

4.5Metoda e Newton-it për lokalizimin e rrënjëve

Shembull 4.5 Gjeni një interval për rrënjët pozitive të polinomit $P(x) = 3x^2 + 4x - 5$

Zgjidhje: Më poshtë kemi kodin në Matlab për metodën e Newton-it për lokalizimin e rrënjëve (*Figura 4.10*) , ku p është polinomi dhe i kërkojmë rrënjët në intervalin $(0, 20)$, ky interval është marrë në mënyrë arbitrare, ne mund të merrnim një interval më të madh pastaj ta zvogëlojmë gradualisht, po ashtu edhe kufiri i poshtëm për rrënjët pozitive.

```
% Metoda e Newton-it për lokalizimin e rrënjëve
clear;
clc;
p = [3 4 -5];
q = -fliplr(p);
x = linspace(0, 3, 4000);
```

```
if sum(polyval(p,x)>0)>0
    x_pozitive = polyval(p,x)>0;
    rezultati = newton_lok(p, x, x_pozitive);
else
    disp('ska rrenje pozitive')
end
if sum(polyval(q,x)>0)>0
    xq_pozitive = polyval(q,x)>0;
    rezultati_q = newton_lok(q, x, xq_pozitive);
else
    disp('ska rrenje pozitive')
end
for i = 1:length(x)
    if rezultati(i) == 1
        kufiri_siperm = x(i);
        break
    end
end
for i = 1:length(x)
    if rezultati_q(i) == 1
        kufiri_poshtem = 1/x(i);
        break
    end
end
disp('Kufiri i Siperm:')
disp(kufiri_siperm)
disp('Kufiri i poshtem:')
disp(kufiri_poshtem)

function rezultati = newton_lok(p, x, x_pozitive)
    rezultati = x_pozitive;
    p_der = polyder(p);
    if sum(polyval(p_der, x)<=0)==length(x)
        return;
    else
        x_pozitive = rezultati .* sign(polyval(p_der,x));
        rezultati = newton_lok(p_der, x, x_pozitive);
    end
end
```

Figura 4.10: Metoda e Newton-it për lokalizimin e rrënjëve

Pas ekzekutimit të kodit të mësipërm fitojmë rezultatin

```
Kufiri i Siperm:
    0.7869

Kufiri i poshtem:
    0.7860
```

Figura 4.11: Rezultati i metodës së Newton-it për lokalizimin e rrënjëve

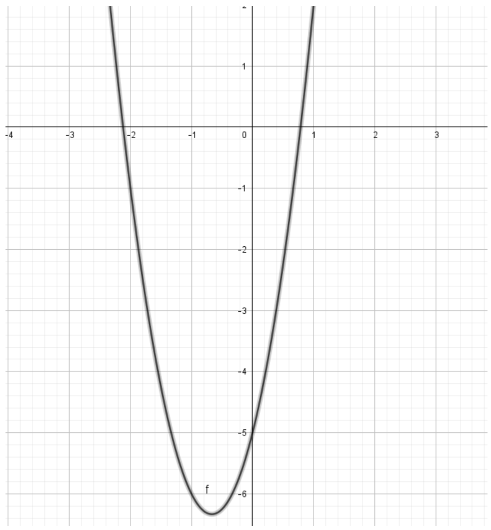

Figura 4.12: Paraqitja grafike e funksionit $P(x) = 3x^2 + 4x - 5$

4.6 Metoda e Sturm-it

Shembull 4.6 Gjeni numrin e rrënjëve të

$$P(x) = 2x^4 - 3x - 2$$

duke shfrytëzuar metodën e Sturm-it në intervalin $(-8,4)$.

Zgjidhje:

Intervali $(-8,4)$ është marrë në mënyrë arbitrare pasi që mund të merret çfarëdo intervali, këtë interval mund ta gjejmë duke shfrytëzuar .

Më poshtë kemi kodin në Matlab për metodën e Sturm-it, *Figura 4.13*

```
% metoda e Sturm-it ne MATLAB
clear;
clc;
a = -8;                         %kufijt e rrenjeve
b = 4;                          %kufijt e rrenjeve
s_a = 0;
```

```
s_b = 0;
q = 0;
mbetja = 0;
P = [2 0 0 -3 -2];            %shenojme vlerat e koeficienteve te 
polinomit P
P_der = polyder(P);           %jep derivatine e polinomit P
P_0 = P;
P_1 = P_der;
if sign(polyval(P_0,a)) ~= sign(polyval(P_1,a))
    s_a = s_a + 1;
end
if sign(polyval(P_0,b)) ~= sign(polyval(P_1,b))
    s_b = s_b + 1;
end
while sum(abs(P_1)) ~= 0
    [q, mbetja] = deconv(P_0, P_1);
    P_0 = pa_zero(P_1);
    P_1 = -pa_zero(mbetja);
    if sign(polyval(P_0,a)) ~= sign(polyval(P_1,a))
        s_a = s_a + 1;
    end
    if sign(polyval(P_0,b)) ~= sign(polyval(P_1,b))
        s_b = s_b + 1;
    end
end
numri_i_rrenjeve = s_a - s_b

function new = pa_zero(vektor)
    if sum(abs(vektor)) ~= 0
        i=1;
        L = length(vektor);
        while (vektor(i)==0)
            vektor = vektor(2:L);
            L = L-1;
        end
    end
    new = vektor;
end
```

Figura 4.13: Kodi i metodës në Sturm-it

Pas ekzekutimit të kodit të mësipërm fitojmë rezultatin *Figura 4.14*

```
numri_i_rrenjeve =

     2
```

Figura 4.14: Rezultati i metodës së Sturm-it

Shih Figura 4.3 paraqitja grafike e funksionit $P(x) = 2x^4 - 3x - 2$

Shembull 4.7 Gjeni rrënjët e

$$P(x) = 2x^4 - 3x - 2$$

duke shfrytëzuar metodën e Sturm-it në intervalin $(-2,3)$.

Zgjidhje: E shfrytëzojmë kodin e mëposhtëm në *Figura 4.15*

```
%gjetja e rrenjeve metoda e Sturm-it
clear;
clc;
a = -2;                     %kufijt e rrenjeve
b = 3;                      %kufijt e rrenjeve
e = 0.001;
max = 10;
P = [2 0 0 -3 -2];          %shenojme vlerat e
koeficienteve te polinomit P
P_der = polyder(P);         %jep derivatine e polinomit P

rrenjet = rrenjet_sturm(P, P_der, a, b, e, max);

function rrenjet = rrenjet_sturm(P, P_der, a, b, e, max)
    rrenjet = [];
    numerimi_i_rrenjeve = sturm(P, P_der, a, b);
    if numerimi_i_rrenjeve > 0
        if numerimi_i_rrenjeve == 1
            [error, rrenje] = Newton(P,P_der, a, b, e,
max);
            if error == 0
                rrenjet = [rrenjet, rrenje];
            end
        else
            c = (a+b)/2;
            rrenjet = [rrenjet, rrenjet_sturm(P, P_der, a,
c, e, max)];
            rrenjet = [rrenjet, rrenjet_sturm(P, P_der, c,
b, e, max)];
        end
    end
end

function numri_i_rrenjeve = sturm(P, P_der, a, b)
    s_a = 0;
    s_b = 0;
    P_0 = P;
    P_1 = P_der;
    if sign(polyval(P_0,a)) ~= sign(polyval(P_1,a))
        s_a = s_a + 1;
```

```
    end
    if sign(polyval(P_0,b)) ~= sign(polyval(P_1,b))
        s_b = s_b + 1;
    end
    while sum(abs(P_1)) ~= 0
        [q, mbetja] = deconv(P_0, P_1); %#ok<ASGLU>
        P_0 = pa_zero(P_1);
        P_1 = -pa_zero(mbetja);
        if sign(polyval(P_0,a)) ~= sign(polyval(P_1,a))
            s_a = s_a + 1;
        end
        if sign(polyval(P_0,b)) ~= sign(polyval(P_1,b))
            s_b = s_b + 1;
        end
    end
    numri_i_rrenjeve = s_a - s_b;
end

function [err, rrenje] = Newton(P,P_der, a, b, e, max)
    err = 1;
    x_0 = (a+b)/2;
    for i=1:max
        emeruesi = polyval(P_der, x_0);
        if emeruesi == 0
            x_1 = x_0;
            rrenje = x_0; %#ok<NASGU>
            break;
        end
        x_1 = x_0 - polyval(P,x_0)/emeruesi;
        if (abs((x_1-x_0)/x_1)<=e)&&(abs(x_1-x_0)<=e)
            err = 0;
            rrenje = x_1; %#ok<NASGU>
            break;
        end
        if x_1<a || x_1>b
            err = 3;
            rrenje = x_1; %#ok<NASGU>
            break;
        end
        x_0 = x_1;
    end
    rrenje = x_1;
end

function new = pa_zero(vektor)
    if sum(abs(vektor)) ~= 0
        i=1;
```

```
        L = length(vektor);
        while (vektor(i)==0)
            vektor = vektor(2:L);
            L = L-1;
        end
    end
    new = vektor;
end
```

Figura 4.15: Gjetja e rrënjëve me metodën e Sturm-it

Pas ekzekutimit të kodit të mësipërm fitojmë rezultatin *Figura 4.16*

```
>> rrenjet

rrenjet =

   -0.5873    1.3127
```

Figura 4.16: Rezultati i rrënjëve sipas metodës së Sturm-it

Shih Figura 4.3 paraqitja grafike e funksionit $P(x) = 2x^4 - 3x - 2$

4.7 Metoda e Vincent-it

Shembull 4.8 Izoloni rrënjët e polinomit

$$P(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$$

duke e shfrytëzuar algoritmin VAC

Zgjidhje: E shfrytëzojmë kodin e mëposhtëm në Figura 4.17

```
% gjetja e rrenjeve metoda e VCA
clear;
clc;
a = 0.2;                                  %kufijt e rrenjeve
b = 6.1;                                  %kufijt e rrenjeve
e = 0.000001;                              %toleranca e saktesise
P = [1 -10 35 -50 24];                  %shenojme vlerat e
koeficienteve te polinomit P

rrenjet = 1./VCAfunc(P, a, b, e);
rrenjet = fliplr(rrenjet);
disp("rrenjet gjenden ne intervalet")
disp("ne cdo interval kemi vetem nga nje rrenje")
for i = 1:2:length(rrenjet)
    disp(rrenjet(i:i+1))
```

```
end

function rrenjet = VCAfunc(P, a, b, e)
    rrenjet = [];

    budan = numri_rrenjeve(fliplr(P), a, b);
    if budan == 0
        return
    end
    if budan == 1
        rrenjet = [rrenjet, a, b];
        return
    end
    if abs(a-b)<e
        rrenjet = [rrenjet, a, b];
        return
    end

    if polyval(P, 1/2) ~= 0
        rrenjet = [rrenjet, VCAfunc(P, a, (a+b)/2, e)];
        rrenjet = [rrenjet, VCAfunc(P, (a+b)/2, b, e)];
        return
    else
        rrenjet = [rrenjet, VCAfunc(P_012, a, (a+b)/2, e)];
        rrenjet = [rrenjet, VCAfunc(P_121, (a+b)/2, b, e)];
        rrenjet = [rrenjet, ((a+b)/2), ((a+b)/2)];
        return
    end
end

function numri_i_rrenjeve = numri_rrenjeve(P, a, b)
    P_der = polyder(P);
    s_a = 0;
    s_b = 0;
    P_0 = P;
    P_1 = P_der;
    if sign(polyval(P_0,a)) ~= sign(polyval(P_1,a))
        s_a = s_a + 1;
    end
    if sign(polyval(P_0,b)) ~= sign(polyval(P_1,b))
        s_b = s_b + 1;
    end
    while sum(abs(P_1)) ~= 0
        [q, mbetja] = deconv(P_0, P_1); %#ok<ASGLU>
        P_0 = pa_zero(P_1);
        P_1 = -pa_zero(mbetja);
        if sign(polyval(P_0,a)) ~= sign(polyval(P_1,a))
```

```
            s_a = s_a + 1;
        end
        if sign(polyval(P_0,b)) ~= sign(polyval(P_1,b))
            s_b = s_b + 1;
        end
    end
    numri_i_rrenjeve = s_a - s_b;
end

function new = pa_zero(vektor)
    if sum(abs(vektor)) ~= 0
        i=1;
        L = length(vektor);
        while (vektor(i)==0)
            vektor = vektor(2:L);
            L = L-1;
        end
    end
    new = vektor;
end
```

Figura 4.17: Algoritmi VCA

Pas ekzekutimit të kodit të në Figura 4.17 fitojmë rezultatin

```
rrenjet gjenden ne intervalet
ne cdo interval kemi vetem nga nje rrenje
    0.5970    1.0667

    1.7582    2.6016

    2.6016    3.4225

    3.4225    5.0000
```

Figura 4.18: Rezultatet e kodit VCA

Le të shënojmë rrënjët e polinomit të dhënë me $\alpha_i, 1 \leq i \leq 4$, atëherë nga rezultati i fituar kemi se rrënjët shtrihen

$$0.5970 < \alpha_1 < 1.0667$$

$$1.7582 < \alpha_2 < 2.6016$$

$$2.6016 < \alpha_3 < 3.4225$$

$$3.4225 < \alpha_4 < 5$$

Për ta konfirmuar rezultatin e arritur polinomin e dhënë në *Shembull 4.8* do ta paraqesim grafikisht

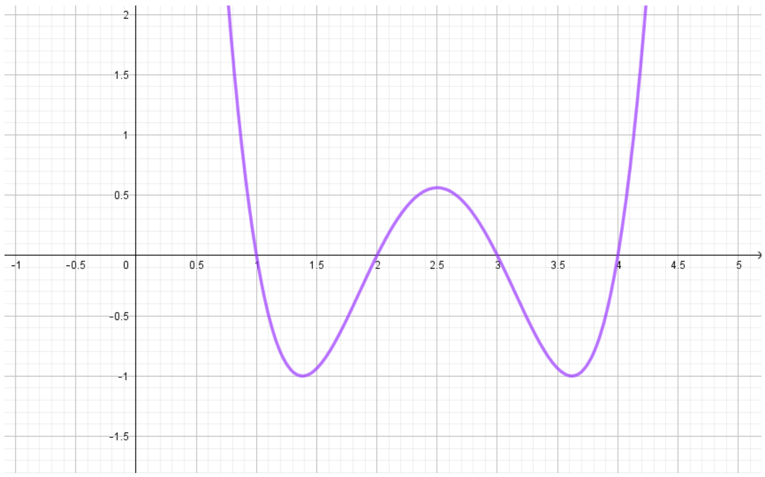

Figura 4.19: Paraqitja grafike e $P(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$

Shembull 4.9 Përafroni rrënjët e polinomit

$$P(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$$

duke shfrytëzuar metodën e përgjysmimit, dhe intervalet e fituara nga metoda e VCA (*Shembull 4.8*)

Zgjidhje: Duke marrë në kombinim kode e metodës VCA dhe metodës së përgjysmimit mund të formojmë kodin e mëposhtëm

```
%gjetja e rrenjeve metoda e VCA
clear;
clc;
a = 0.2;                                    %kufijt e rrenjeve
b = 6.1;                                    %kufijt e rrenjeve
e = 0.000001;                               %toleranca e saktesise
P = [1 -10 35 -50 24];                      %shenojme vlerat e
koeficienteve te polinomit P
rrenjet = 1./VCAfunc(P, a, b, e);
rrenjet = fliplr(rrenjet);
rrenja = [];
disp("rrenjet e polinomit jane")
for i = 1:2:length(rrenjet)
    A = rrenjet (i);
    B = rrenjet (i+1);
    rrenja = [rrenja pergjysmorja(P, A, B, e)];
end

disp(rrenja)

function rrenjet = VCAfunc(P, a, b, e)
```

```
    rrenjet = [];

    budan = numri_rrenjeve(fliplr(P), a, b);
    if budan == 0
        return
    end
    if budan == 1
        rrenjet = [rrenjet, a, b];
        return
    end
    if abs(a-b)<e
        rrenjet = [rrenjet, a, b];
        return
    end

    if polyval(P, 1/2) ~= 0
        rrenjet = [rrenjet, VCAfunc(P, a, (a+b)/2, e)];
        rrenjet = [rrenjet, VCAfunc(P, (a+b)/2, b, e)];
        return
    else
        rrenjet = [rrenjet, VCAfunc(P_012, a, (a+b)/2, e)];
        rrenjet = [rrenjet, VCAfunc(P_121, (a+b)/2, b, e)];
        rrenjet = [rrenjet, ((a+b)/2), ((a+b)/2)];
        return
    end
end

function numri_i_rrenjeve = numri_rrenjeve(P, a, b)
    P_der = polyder(P);
    s_a = 0;
    s_b = 0;
    P_0 = P;
    P_1 = P_der;
    if sign(polyval(P_0,a)) ~= sign(polyval(P_1,a))
        s_a = s_a + 1;
    end
    if sign(polyval(P_0,b)) ~= sign(polyval(P_1,b))
        s_b = s_b + 1;
    end
    while sum(abs(P_1)) ~= 0
        [q, mbetja] = deconv(P_0, P_1); %#ok<ASGLU>
        P_0 = pa_zero(P_1);
        P_1 = -pa_zero(mbetja);
        if sign(polyval(P_0,a)) ~= sign(polyval(P_1,a))
            s_a = s_a + 1;
        end
        if sign(polyval(P_0,b)) ~= sign(polyval(P_1,b))
```

```
            s_b = s_b + 1;
        end
    end
    numri_i_rrenjeve = s_a - s_b;
end

function new = pa_zero(vektor)
    if sum(abs(vektor)) ~= 0
        i=1;
        L = length(vektor);
        while (vektor(i)==0)
            vektor = vektor(2:L);
            L = L-1;
        end
    end
    new = vektor;
end

function x = pergjysmorja(P, a, b, e)
    x = (a+b)/2;
    if (x-a)<=e
        return
    end
    if polyval(P,a)*polyval(P,x)<0
        x = pergjysmorja(P, a, x, e);
        return
    else
        x = pergjysmorja(P, x, b, e);
        return
    end
end
```

Figura 4.20: Izolimi i rrënjëve me metodën VCA, dhe përafrimi me metodën e përgjysmimit

Shih Figura 4.19 paraqitja grafike e funksionit $P(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$

Zgjedhja e metodës së përgjysmimit ishte në mënyre arbitrare pasi që kishim mundësi të merrnim cilëndo nga metodat përafruese të përshkruara në pjesën e dytë.

Përfundimi

Në këtë pjesë të punimit do të bëjmë një rezyme mbi atë që është paraqitur gjatë hartimit të temës.

Fillimisht, gjegjësisht në kapitullin e parë, dhamë disa koncepte dhe veti themelore për polinomet, duke i konkretizuar ato përmes shembujve.

Pastaj, për të vazhduar në kapitullin e dytë dhe të tretë me trajtimin e disa metodave më të përhapura në studimin, përkatësisht izolimin e rrënjëve të polinomeve. Ajo që është me rëndësi të potencohet është se bëmë një studim komparativ mes metodave, duke i vënë në pah përparësitë dhe mangësitë e tyre, duke treguar se në cilat raste cila është më efikase. Përveç mënyrës teorike, këto janë dëftuar edhe përmes shembujve, ku mund të arrihet deri te një konceptim më i qartë i asaj që është trajtuar teorikisht.

Kapitulli fundit, në fakt ky është kapitulli që , në një farë mënyre, dëshmon se roli i softuerëve matematik në zhvillimin e matematikës është krucial, pasi që përveç zgjidhjeve me saktësi të lartë, na kursejnë mund dhe kohë, me një fjalë ndikon në zhvillimin e teorisë në fjalë, në veçanti, e matematikës në përgjithësi.

Mund të konkludojmë se çdonjëra metodë e përshkrua më sipër ka përparësitë dhe mangësitë e veta të cilat janë të shqyrtuara në kapitujt korrespondues mund të themi se metoda më efikase për përafrimin e rrënjëve do të ishte metoda e Newton-it por për raste specifike kjo metodë mund të hyj në cikël të pafundëm apo edhe mund të divergjoj, kështu që do të ishte më e sigurte nëse do të shfrytëzonim një kombinim të metodës së përgjysmimit dhe metodës së Newton-it që na jep rezultat në më pak iteracione dhe për kohë të shkurtër, në të shumtën e rasteve. Përsëri e cekim për ndonjë problem mund të ishte më efikase të shfrytëzojmë ndonjërën nga metodat tjera.

Poashtu edhe metodat për izolimin e rrënjëve kishin përparësitë dhe dobësitë e tyre, shpesh duhej të zgjidhnim mes metodës më të shpejtë po e cila nuk na jep si rezultat të gjitha rrënjët e polinomit, pra nuk na jep intervale ku gjinden vetëm nga një rrënjë dhe metodave të cilat na jepnin më shumë ose të gjitha rrënjët e poilinomit por nevojiten forcë më e madhe kalkuluese. Përsëri dukej të këmbenim mes shpejtësisë dhe "kualitetit". Nëse duhej të izolonim rrënjët e një polinomi të shkallës relativisht të vogël do të ishte më e lehtë për ne që të shfrytëzonim metodën e Vincent-it, pasiqë na jepte intervale në të cilat shtrihen vetëm nga një rrënjë, që më pas mund të kombinohet me metodën e Newton-it ose me metodën e përgjysmores për të fituar një përafrim të kërkuar të të gjitha rrënjëve.

Përmes softuerit Matlab, kemi gjetur, respektivisht izoluar rrënjët e një polinomi, por me të gjitha metodat që janë dhënë paraprakisht, duke dhënë një vështrim kritik për secilën prej tyre.

Le të jetë e ardhmja, koha kur do të shohin dritën edhe shumë punime tjera të këtij lloji, kështu do të zgjerojmë diapazonin e njohurive mbi këtë lëmi sa të rëndësishme, aq interesante.

Bibliografia

[1] F. Hoxha, Metoda të analizës numerike (Algoritme numerike), Tiranë: Infobotues, 2012.

[2] K. T. Leung, I. A. C. Mok dhe S. N. Suen, Polynomials and Equations, Hong Kong: Hong Kong University Press, 1992.

[3] J. Rosickz, Algebra, Brno, 2007.

[4] K. G. Binmore, Mathematical Analysis: A Straightforward Approach, Cambridge: Cambridge University Press, 1997.

[5] K. Miller, Polynomials are continuous functions, Berkeley: The University of California, 2014.

[6] K. Atkinson, An Introduction to Numerical Analysis, 2. ed, Re., Iowa: John Wiley & Sons, 1989.

[7] H. Jeffreys and B. Jeffreys, Methods of Mathematical Physics, 3rd ed ed., Cambridge: Cambridge University Press, 2000.

[8] T. R. Scavo dhe J. B. Thoo, On the Geometry of Halley's Method, vëll. i 102, The American Mathematical Monthly, 1995, pp. 417-426.

[9] J. H. Mathews, Theory and Proof for Halley's Method, 2003.

[10] G. Alefeld, On the Convergence of Halley's Method, vëll. i 88, The American Mathematical Monthly, 1981, pp. 530-536.

[11] A. L. Cauchy, Analyse Algebrique, Paris: Jacques Gabay, 1989.

[12] L. Fan, A Generalization of Synthetic Division and A General Theorem of Division of Polynomials, vëll. i 30, Mathematical Medle, 2003, pp. 30-37.

[13] J. C. Sturm, Mémoire sur la résolution des équations numériques, vëll. i 11, Bulletin des Sciences de Férussac, 1829, pp. 419-425.

[14] P. Barlett, Finding All the Roots: Sturm's Theorem, 2013.

[15] A. J. Vincent, Mémoire sur la résolution des équations numériques, Lille: Mémoires de la Société Royale des Sciences, de l' Agriculture et des Arts, de Lille, 1834, pp. 1-34.

[16] A. J. Vincent, Sur la résolution des équations numériques, vëll. i 1, Journal de Mathématiques Pures et Appliquées, 1834, pp. 341-374.

[17] A. Alesina dhe M. Galuzzi, Vincent's Theorem from a Modern Point of View, Rendiconti del Circolo Matematico di Palermo Serie II, 2000, p. 179–191.

[18] P. Vigklas, Upper bounds on the values of the positive roots of polynomials, Volos, 2010.