Web aplikacija za rješavanje nelinearne jednadžbe s jednom nepoznanicom

Minić, Nikola

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:929754

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-12**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Stručni prijediplomski studij Računarstvo

WEB APLIKACIJA ZA RJEŠAVANJE NELINEARNE JEDNADŽBE S JEDNOM NEPOZNANICOM

Završni rad

Nikola Minić

Osijek, 2024

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1S: Obrazac za ocjenu završnog rada na stručnom prijediplomskom studiju

Ocjena završnog rada na stručnom prijediplomskom studiju

Ime i prezime pristupnika:	Nikola Minić
Studij, smjer:	Stručni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	ARV 22, 22.09.2021.
JMBAG:	0165092454
Mentor:	Ivan Hrehorović, prof.
Sumentor:	izv. prof. dr. sc. Zdravko Krpić
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	izv. prof. dr. sc. Alfonzo Baumgartner
Član Povjerenstva 1:	Ivan Hrehorović, prof.
Član Povjerenstva 2:	doc. dr. sc. Tomislav Rudec
Naslov završnog rada:	Web aplikacija za rješavanje nelinearne jednadžbe s jednom nepoznanicom
Znanstvena grana završnog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada:	Napraviti web aplikaciju koja će za zadanu nelinearnu jednadžbu s jednom nepoznanicom doći do traženih rješenja jednom od numeričkih metoda. Sumentor s FERIT-a: Zdravko, Krpić

Datum ocjene pismenog dijela završnog rada od strane mentora:	12.09.2024.
Ocjena pismenog dijela završnog rada od strane mentora:	Izvrstan (5)
Datum obrane završnog rada:	26.09.2024.
Ocjena usmenog dijela završnog rada (obrane):	Vrlo dobar (4)
Ukupna ocjena završnog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio stručni prijediplomski studij:	08.10.2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O IZVORNOSTI RADA

Osijek, 08.10.2024.

Ime i prezime Pristupnika:	Nikola Minić
Studij:	Stručni prijediplomski studij Računarstvo
Mat. br. Pristupnika, godina upisa:	ARV 22, 22.09.2021.
Turnitin podudaranje [%]:	11

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za rješavanje nelinearne jednadžbe s jednom nepoznanicom**

izrađen pod vodstvom mentora Ivan Hrehorović, prof.

i sumentora izv. prof. dr. sc. Zdravko Krpić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD

Tema ovog završnog rada je razvoj web aplikacije za rješavanje nelinearnih jednadžbi s jednom nepoznanicom, koristeći numeričke metode. Nelinearne jednažbe nije uvijek moguće riješiti pomoću formula, u takvim slučajevima koriste se numeričke metode koje omogućuju približno rješenje problema. U ovom radu, web aplikacija će omogućiti rješavanje nelinearnih jednadžbi pomoću dvije numeričke metode: metode bisekcije i Newtonove metode. Ove metode pružaju različite pristupe za nalaženje rješenja i odabrane su zbog svoje učinkovitosti i široke primjene u rješavanju nelinearnih jednadžbi. Metoda bisekcije je pouzdana metoda koja se koristi za pronalaženje rješenja unutar zadanog intervala, dok je Newtonova metoda brža, ali zahtijeva dobru početnu pretpostavku za konvergenciju.

Cilj ovog završnog rada je izraditi web aplikaciju koja će biti dostupna svima, omogućujući korisnicima jednostavano rješavanje nelinearnih jednadžbi. Aplikacija će imati jednostavno sučelje u kojem će korisnici moći unijeti jednadžbu, definirati interval ili početnu točku, kao i razinu točnosti rješenja.

U drugom poglavlju bit će prikazana osnovna teorija nelinearnih jednadžbi, uključujući različite vrste jednadžbi, njihove karakteristike, te primjeri problema gdje se nelinearne jednadžbe pojavljuju. Također će biti objašnjene metode za njihovo analitičko i numeričko rješavanje, te će se detaljnije obraditi metoda bisekcije i Newtonova metoda, koje će biti implementirane u aplikaciji. Treće poglavlje bit će posvećeno izradi web aplikacije. Bit će opisane tehnologije korištene za razvoj aplikacije, te će biti dan detaljan opis funkcionalnosti aplikacije, uključujući algoritme za rješavanje jednadžbi. U četvrtom poglavlju bit će prikazana analiza funkcionalnosti aplikacije kroz konkretne primjere rješavanja nelinearnih jednadžbi, te će biti obavljena evaluacija točnosti i učinkovitosti korištenih numeričkih metoda.

1.1. Zadatak završnog rada

U ovom završnom radu je potrebno analizirati nelinearne jednadžbe s jednom nepoznanicom i napraviti web aplikaciju za rješavanje jednadžbi jednom od numeričkih metoda.

2. NELINEARNE JEDNADŽBE

Nelinearne jednadžbe su jednadžbe u kojima se pojavljuje najmanje jedan od članova koji je nelinearan. Zbog toga su ove jednadžbe puno složenije i teže za rješavanje za razliku od linearnih jednadžbi.

Imamo različite vrste nelinearnih jednadžbi, a to su:

1. **Algebarske nelinearne jednadžbe** – jednadžbe koje u sebi imaju polinome sa nepoznatim varijablama, gdje su eksponenti varijabli nelinearni, tj. veći od jedan.
$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0, a_n \neq 0$$
2. **Transcedentne nelinearne jednadžbe –** jednadžbe koje nisu algebarske, kao što su trigonometrijske, eksponencijalne i logaritamske funkcije. Za razliku od algebarskih jednadžbi, koje se mogu riješiti formulama do određenog stupnja (n ≤ 4), transcendentne jednadžbe u većini slučajeva nemogu, pa se za rješavanje složenih algebarskih i transcendentnih jednadžbi koriste numeričke metode koje rješenje daju s određenom točnošću.
3. **Diferencijalne nelinearne jednadžbe –** jednadžbe koje uključuju nepoznate funkcije i njihove derivacije, pri čemu su funkcije i derivacije nelinearno povezane. Zbog te složenosti, nemaju opća analitička rješenja, pa se često rješavaju numeričkim metodama.

Primjeri nelinearnih jednadžbi:

$x^2 - 9 = 0$ → rješenja su $x_1 = -3$ i $x_2 = 3$

$x^2 + 1 = 0$ → nema rješenja

Naš cilj je pronaći (realna) rješenja nelinearnih jednadžbi oblika $f(x) = 0$

Ako je vrijednost $\boldsymbol{x}$ takva da je $f(x) = 0$ kažemo da je:

1. x je rješenje jednadžbe $f(x) = 0$
2. x je nultočka funkcije f

2.1. Rješavanje nelinearnih jednadžbi

Da bismo eksplicitno riješili nelinearne jednadžbe, osim linearnih, one predstavljaju veliki izazov jer nemamo formule za rješavanje. Na primjer, Galoisova (*Elvariste Galois, 1811 - 1832*) teorija je pokazala da ne postoji konačan skup operacija koji bi nam omogućio da dobijemo točna rješenja u slučaju polinoma viših stupnjeva. U srednjoj školi često smo rješavali trigonometrijske, eksponencijalne i logaritamske jednadžbe, ali one su bile postavljene tako da su se mogle riješiti primjenom poznatih metoda i tehnika. Na primjer, znamo riješiti jednadžbu $x^2 = 4$ jer je to kvadratna jednadžba s jednostavnim rješenjem $x_1 = 2$ i $x_2 = -2$, no kako bismo riješili jednadžbu poput $x^3+x^2 = 5$ ili $e^x = x$?

Općenito, rješavanje bilo koje jednadžbe s jednom nepoznanicom svodi se na pronalaženje nultočaka odgovarajuće funkcije. Stoga umjesto traženja eksplicitnog oblika rješenja često koristimo numeričke metode za aproksimaciju tih nultočki. To uključuje iterativne postupke poput Newtonove metode, metode bisekcije ili druge metode koje konvergiraju prema rješenju dane jednadžbe. Ove metode omogućuju dobivanje približnih vrijednosti za rješenja s određenom točnošću.

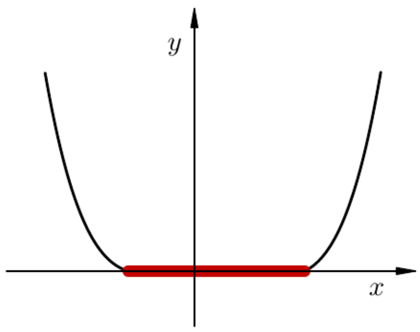
Neka je zadana nelinearna funkcija $f : I \rightarrow \mathbb{R}$, gdje je I neki interval. Tražimo sve one točke $x \in I$ za koje je $f(x) = 0$

Takve točke x zovu se **rješenja** ili **korijeni** pripadne jednadžbe, odnosno **nultočke** funkcije $\boldsymbol{f}$.

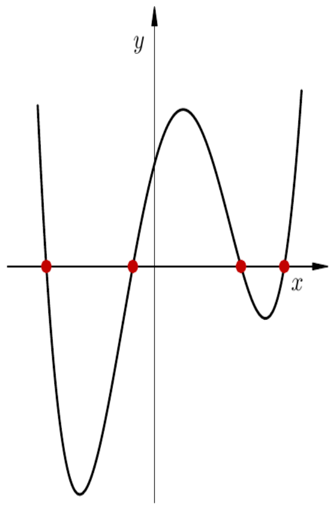
Pretpostavljamo:

- Funkcija $\boldsymbol{f}$ je neprekidna na $\boldsymbol{I}$
- Nultočke funkcije $\boldsymbol{f}$ su izolirane

Nultočka $\alpha \epsilon I$ funkcije f izolirana ako postoji okolina od α u kojoj ne postoji neka druga nultočka od f. [2]



Slika 2.1 *Funkcija ima cijeli interval nultočki, niti jedna nije izolirana* [2]



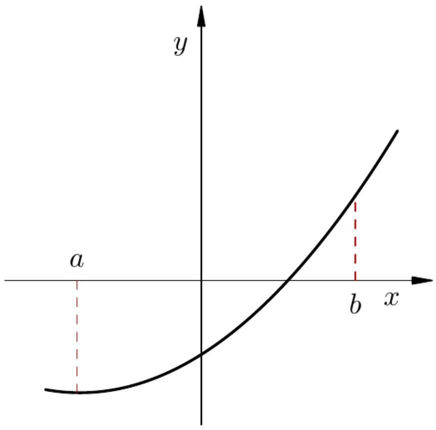
Slika 2.2 *Funkcija ima 4 izolirane nultočke* [2]

Neprekidnost funkcije $\boldsymbol{f}$ se često koristi za određivanje intervala u kojem se nalazi nultočka. Ako za neki segment [a,b] vrijedi:

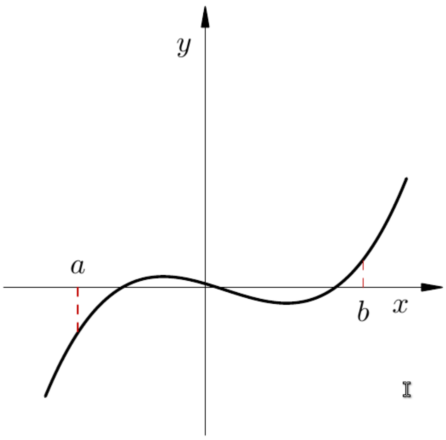
$$f(a) \cdot f(b) < 0 \tag{2-1}$$

i ako je funkcija $\boldsymbol{f}$ neprekidna, tada nužno postoji nultočka unutar intervala [a,b]. Naime, ili je $f(a) < 0 < f(b)$ ili je $f(b) < 0 < f(a)$, što prema poznatom rezultatu iz analize znači da neprekidna funkcija prelazi sve vrijednosti između $f(a)\ i\ f(b)$. Dakle, postoji točka α (alpha) u [a,b] za koju vrijedi $f(\alpha) = 0$.

Treba napomenuti da unutar intervala [a,b] može postojati više nultočaka funkcije f, što je prikazano na slikama 2.3 i 2.4. U oba primjera vrijedi $f(a) \cdot f(b) < 0$.



Slika 2.3. *Funkcija s jednom nultočkom u intervalu [a,b]* [2]

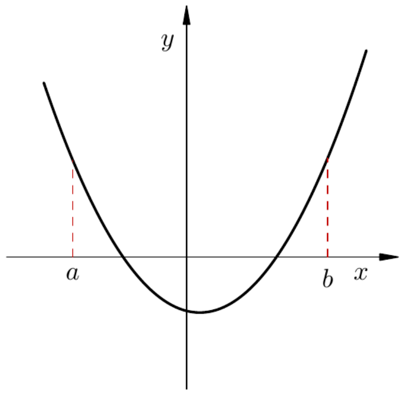


Slika 2.1. *Funkcija s više nultočki u intervalu [a,b]* [2]

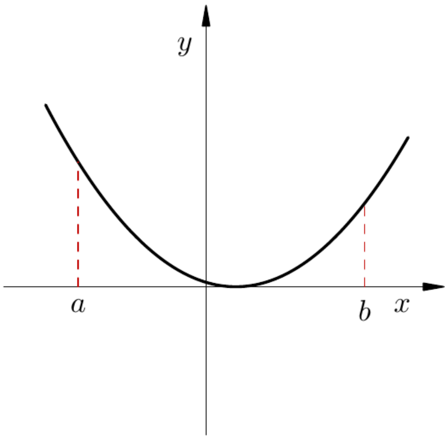
Za svaki interval [a,b] za koji vrijedi $f(a) \cdot f(b) > 0$, oni također mogu (ali ne moraju) sadržavati nultočke funkcije $\boldsymbol{f}$. Da bi smo ih pronašli, potrebno je detaljnije analizirati funkciju $\boldsymbol{f}$ i odabrati manji početni interval. Primjetimo da na slici 2.3. postoje mali segmenti [a,b] oko svake nultočke gdje vrijedi $f(a) \cdot f(b) < 0$. Nasuprot tome na slici 2.4. ne može se pronaći takav interval jer je nultočka parna. Problem pronalaženja takvih nultočki može biti numerički nestabilan: za funkciju $\boldsymbol{f}$ prikazanu na slici 2.4., čak i za vrlo mali $\varepsilon > 0$, funkcija $f(x) + \varepsilon$ nema nultočke. To nije slučaj na slici 2.3. niti za prethodne slike.

Ako su sve nultočke neprekidne funkcije $\boldsymbol{f}$ na segmentu [a,b] izolirane, onda takvih nultočaka može biti samo konačno mnogo.

Ako to nije slučaj, tada postoji beskonačan niz nultočaka $\alpha_1, \alpha_2, \ldots$ unutar [a,b]. S obzirom da je [a,b] ograničen interval, ovaj niz nultočaka ima gomilište $\alpha \in [a, b]$. Ova točka gomilišta bi također bila nultočka funkcije $\boldsymbol{f}$ zbog njene neprekidnosti.



Slika 2.2. *Prikaz funkcije koja ima nultočku unutar intervala [a,b]* [2]



Slika 2.3. *Prikaz funkcije koja nema nultočku unutar intervala [a,b]* [2]

Cilj nam je, za zadanu točnost ε > 0 pronaći aproksimacije $\hat{\alpha}_1, \hat{\alpha}_2 \ldots \hat{\alpha}_k$ za sve nultočke $\alpha_1, \alpha_2, \ldots \alpha_k$ funkcije $\boldsymbol{f}$ tako da za apsolutnu pogrešku aproksimacije vrijedi:

$$|\hat{\alpha}_i - \alpha_i| \leq \varepsilon$$

Postupak se sastoji od dvije faze:

1. Pronalaženje segmenata [a,b] koji sadrže barem jednu nultočku. Ovo je izazovan zadatak koji se obavlja analizom toka funkcije.
2. Primjena iterativnog algoritma za određivanje nultočaka s traženom točnošću ε

2.2. Metoda bisekcije

Metoda bisekcije predstavlja jedan od najjednostavnijih postupaka za pronalaženje riješenja nelinearnih jednadžbi, tj. nultočke. Ako je funkcija f(x) neprekidna u intervalu [a,b] i zadovoljava uvjet $f(a) \cdot f(b) < 0$, tada se metoda bisekcije može koristiti za sigurno konvergiranje prema rješenju α. [4]

Kako funkcionira metoda bisekcije?

Postupak se sastoji od sljedećih koraka:

1. Prvo se određuje srednja točka c_1 intervala [a,b] prema formuli

   $$c_1 = \frac{a_1+b_1}{2}$$

   gdje su: $a_1 = a$, $b_1 = b$.

   Ako je $f(c_1) = 0$ tada je c_1 traženo rješenje.

2. Ako je $f(c_1) \neq 0$ i vrijedi $f(a_1) \cdot f(c_1) < 0$, tada se novi interval smanjuje na [a_1, c_1] i ponovo se računa srednja točka novog intervala. Ako je $f(a_1) \cdot f(c_1) > 0$, tada je korijen u intervalu [c_1, b_1].
3. Ovaj postupak se ponavlja sve dok se duljina intervala ne smanji na željenu točnost ε ili se ne dostigne unaprijed određeni broj koraka [5]

Matematički, točnost aproksimacije može se izraziti kao:

$$|\alpha - x_n| \leq \frac{1}{2^n}(b-a) \leq \varepsilon \tag{2-2}$$

Iz formule (2-2) slijedi da broj koraka n mora zadovoljiti:

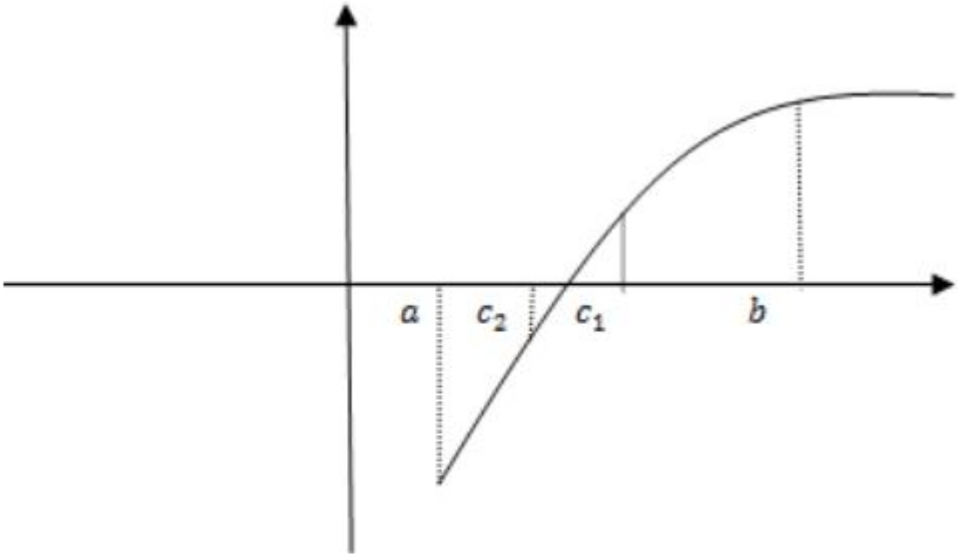
$$n \geq \frac{log(b-a) - log\,\varepsilon}{log\,2} \tag{2-3}$$

Ako funkcija f(x) ima neprekidnu prvu derivaciju, tada se dodatno može koristiti sljedeća procjena:

$$|\alpha - x_n| \leq \frac{|f_{x_n}|}{m_1} \leq \varepsilon \tag{2-4}$$

Gdje je m_1 najmanja vrijednost prve derivacije f(x) na intervalu [a,b].

Na slici 2.5 ispod je prikazana funkcija f(x) i postupak određivanja srednje točke intervala i sužavanja intervala.



Slika 2.4. *Metoda bisekcije* [4]

Primjer 1.: Riješimo metodom bisekcije jednadžbu $x^2 - 4 = 0$ na intervalu [1,3] s toćnošću 0.1.

Rješenje:

U ovom primjeru koristimo metodu bisekcije za rješavanje zadane jednadžbe. U nastavku su prikazani koraci rješavanja.

Imamo zadani interval [a,b] = [1,3]. Izračunamo vrijednosti funkcije na krajevima intervala:

$$f(1) = 1^2 - 4 = -3$$

$$f(3) = 3^2 - 4 = 5$$

Vidimo da vrijednosti funkcije na krajevima intervala imaju različit predznak što znači da unutar intervala postoji bar jedna nultočka funkcije.

1. **Prvo izračunavamo srednju točku intervala:**

   $$c_1 = \frac{a+b}{2} = \frac{1+3}{2} = 2$$

2. **Izračunamo vrijednost funkcije u toj točki:**

   $f(c_1) = f(2) = 2^2 - 4 = 0$

   Vidimo da je $f(c_1) = 0$, to znači da je c_1 traženo rješenje.

   Da bi pokazali postupak sa više iteracije pretpostavićemo da je $f(c_1) \neq 0$

3. **Računamo broj potrebnih iteracija za postizanje točnosti $\boldsymbol{\varepsilon = 0,1}$:**

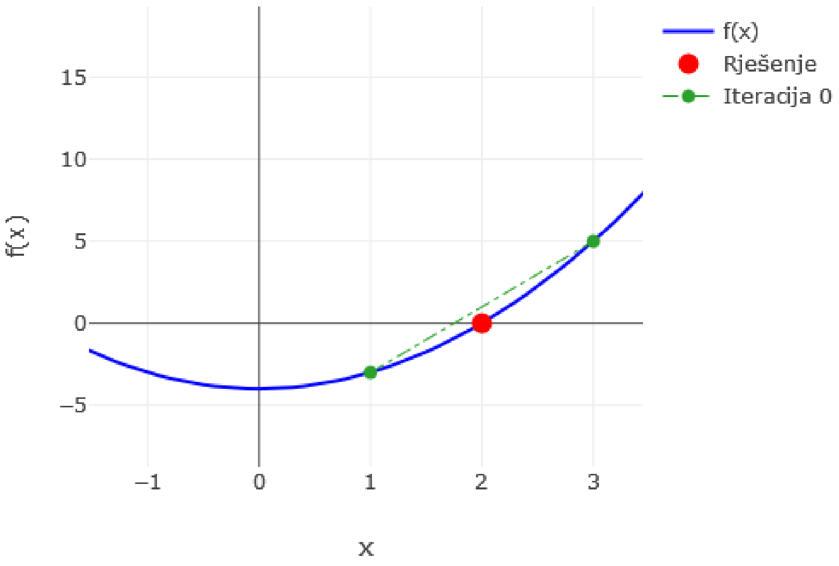
$$\mathrm{n} \geq \frac{\log(\mathrm{b}-\mathrm{a}) - \log \varepsilon}{\log 2}$$

Gdje su: a = 1, b = 3 i $\varepsilon = 0.1$

$$n \geq \frac{log(3-1) - log\,0,1}{log\,2}$$

$$\mathrm{n} \geq 4.32$$

Iz izračuna vidimo da je potrebno najmanje 5 iteracija.



Slika 2.5. *Grafički prikaz funkcije*

Na slici 2.5. je prikazan graf funkcije sa označenim intervalom i srednjom točkom, tj. u ovom slučaju rješenje.

4. **Tablica iteracija**

Tablica 2.1 *Tablica iteracija*

Iteracija	a	b	c	$\mathrm{f(a)}$	$\mathrm{f(b)}$	$\mathrm{f(c)}$	Novi interval
1.	1	3	2	-3	5	0	[1,3]
2.	1	2	1.5	-3	0	-1.75	[1.5,3]
3.	1.5	3	2.25	-1.75	5	1.0625	[1.5,2.25]
4.	1.5	2.25	1.875	-1.75	1.0625	-0.484375	[1.875,2.25]
5.	1.875	2.25	2.0625	-0.4844	1.0625	0.25390625	[1.875, 2.0625]

Rješenje jednadžbe $x^2 - 4 = 0$ na intervalu [1,3] s točnošću $\varepsilon = 0.1$ je približno $\mathbf{x \approx 2,0625}$. Vrijednost funkcije $f(x)$ na toj točki je približno **0.25390625** što je dovoljno blizu nuli s obzirom na zadanu točnost.

Primjer 2.: Riješimo metodom bisekcije jednadžbu lnx + x – 3 = 0, s točnošću 0.005

Rješenje:

Pošto nije zadan interval morat ćemo ga sami odrediti tako što ćemo provjeriti vrijednosti funkcije u nekoliko točaka.

$$f(1) = \ln 1 + 1 - 3 = 0 + 1 - 3 = -2$$

$$f(2) = \ln 2 + 2 - 3 \approx 0.693 + 2 - 3 = -0.307$$

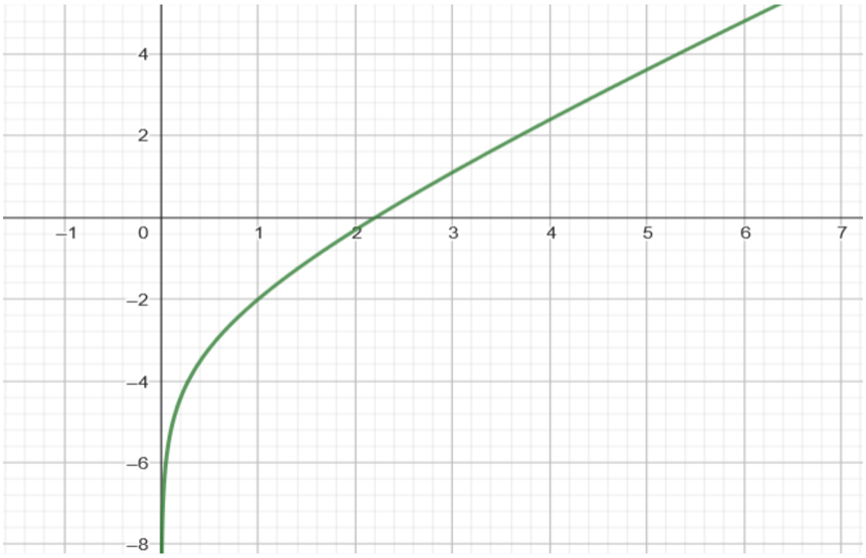
$$f(3) = \ln 3 + 3 - 3 \approx 1.099 + 3 - 3 = 1.099$$

Iz ovoga vidimo da je $f(2) < 0$ i $f(3) > 0$, tako da se rješenje nalazi u intervalu [2,3].

Izračunamo sredinu intervala: $c = \frac{a+b}{2} = \frac{2+3}{2} = 2,5$

Izračunamo vrijednost funkcije u sredini: $f(2,5) = \ln 2,5 + 2,5 - 3 \approx 0.9163 + 2,5 - 3 = 0.4163$

Računamo broj potrebnih iteracija: $n \geq \frac{log(3-2) - log(0,005)}{log\,2} \approx 7,6439 \rightarrow n = 8$



Slika 2.6. Graf funkcije ln(x) + x -3

Tablica 2.2. *Tablica iteracija*

Iteracija	a	b	c	f(a)	f(b)	f(c)	Novi interval
1.	2	3	2,5	-0,307	1,099	0,4163	[2,2.5]
2.	2	2,5	2,25	-0.307	0,4163	0,0609	[2,2.25]
3.	2	2,25	2,125	-0.307	0,0609	-0,1212	[2.125,2.25]
4.	2,125	2,25	2,1875	-0,1212	0,0609	-0,0297	[2.1875,2.25]
5.	2,1875	2,25	2,2188	-0.0297	0.0609	0.0155	[2.1875,2.2188]
6.	2,1875	2,2188	2,2031	-0,0297	0,0155	-0,0071	[2.2031,2.2188]
7.	2,2031	2,2188	2,2109	-0,0071	0,0155	0,0042	[2.2031,2.2109]
8.	2,2031	2,2109	2,2070	-0,0071	0,0042	-0,0013	[2.2070,2.2109]

Nakon 8 iteracija suzili smo interval tako da odgovara zadanoj točnosti. **Konačno rješenje je 2,207.**

2.3. Newtonova metoda (metoda tangente)

[5] Newtonova metoda, također poznata kao Newton-Raphsonova metoda je moćna tehnika za numeričko rješavanje jednadžbi. Ova se metoda često koristi za aproksimaciju riješenja realnih funkcija. Razvili su je Isac Newton i Joseph Raphson po kojima metoda nosi ime.

Newtnova metoda formira niz aproksimacija $\{x_n\}_n \in \mathbb{N}_0$

Kako radi Newtonova metoda?

Newtonova metoda se temelji na iterativnom poboljšanju početne pretpostavke kako bi se konvergiralo prema željenom rješenju. Pretpostavimo da je funkcija neprekidna na intervalu [a,b] i da ima neprekidnu prvu derivaciju.

Ako je $f(a) \cdot f(b) < 0,$ funkcija f ima riješenje u $[a, b]$.

Newtonova metoda formira niz aproksimacija $\{x_n\}_n \in \mathbb{N}_0$ oblika:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2-5}$$

Gdje je:

- x_n je trenutna aproksimacija rješenja
- $f(x_n)$ je vrijednost funkcije f u točki x_n.
- $f'(x_n)$ je vrijednost prve derivacije funkcije f u točki x_n.

Ocjena pogreške u Newtonovoj metodi je oblika:

$$|\alpha - x_n| \leq \frac{M_2}{2m_1}(x_n - x_{n-1})^2 \tag{2-6}$$

Gdje su:

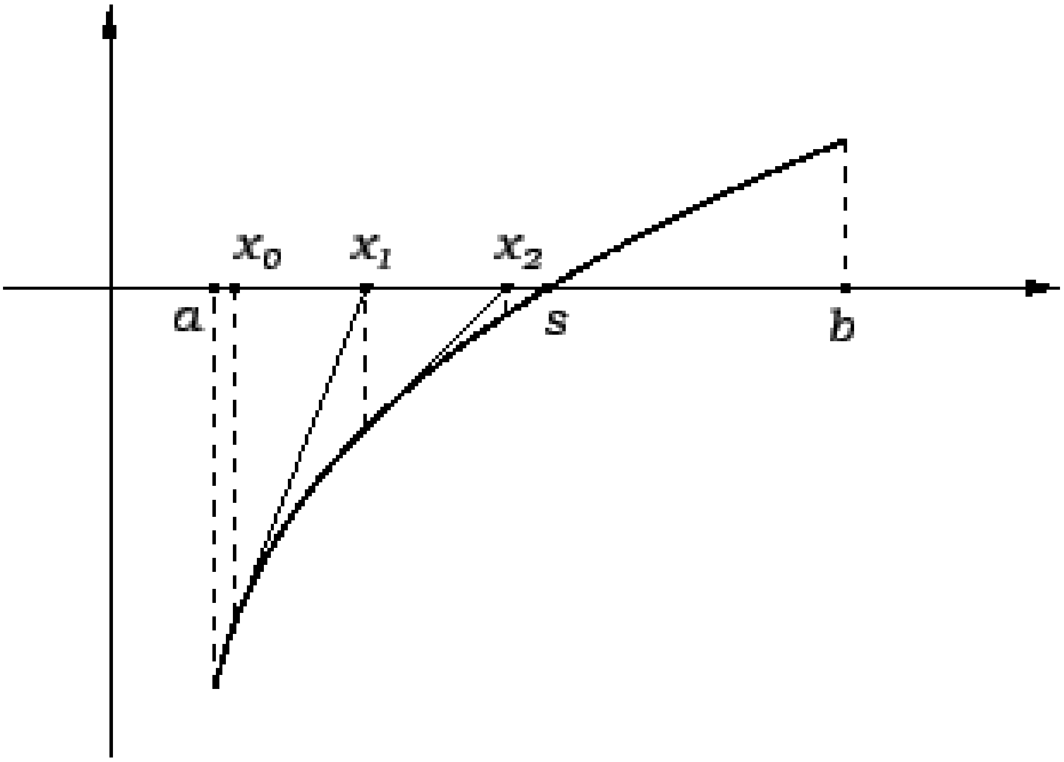
- α svarno rješenje funkcije
- M_2 je maksimalna vrijednost apsolutne vrijednosti druge derivacije funkcije na intervalu
- m_1 je minimaln vrijednost apsolutne vrijednosti prve derivacije funkcije na intervalu

Ako želimo da pogreška u krajnjoj aproksimaciji bude manja ili jednaka $\varepsilon,\ dovoljno\ \ je$:

$$|x_n - x_{n-1}| \leq \sqrt{\frac{2m_1\varepsilon}{M_2}} \tag{2-7}$$

U nekim slučajevima Newtnova metoda sigurno konvergira. Uvjeti za sigurnu konvergenciju uključuju:

- Prva i druga derivacija funkcije ne mijenjaju predznak na intervalu [a,b]
- Početna točka je odabrana bliže onom kraju intervala gdje je apsolutna vrijednost derivacije veća.



Slika 2.7. *Newtnova metoda*

Primjer 3. Riješimo jednadžbu $e^{-x}-2+x = 0$ Newton-Raphsonovom metodom s točnošću 0.01.

Rješenje:

1. **Određivanje m_1 i M_2.**

$f'(x) = -e^{-x}+1 > 0$ *za* $x \in [1,2] \Rightarrow m_1 = v(1) = -e^{-1}+1 = 1 - \frac{1}{e} = 0.632121$

$f''(x) = e^{-x} > 0$ *za* $x \in [1,2] \Rightarrow M_2 = f''(1) = \frac{1}{e} = 0.367879$

$f'''(x) = -e^{-x} < 0 \rightarrow$ treba nam za lakše određivanje M_2

Iz ovoga vidimo da je f' rastuća jer je $f'' > 0$ i pozitivna pa je min u lijevoj granici intervala.

f'' je padajuća jer jer $f''' < 0$ i pozitivna pa je max u lijevoj granici intervala.

2. **Dinamička ocjena pogreške**

$$|x_n - x_{n-1}| \leq \sqrt{\frac{2 \cdot (1 - \frac{1}{e}) \cdot 10^{-2}}{\frac{1}{e}}} = 0{,}18538$$

3. **Određivanje startne točke**

   $x_0 = 2$ jer je $f(2) \cdot f''(2) > 0$

4. **Konačno rješenje**

   $x_1 = 2 - \frac{f(2)}{f'(2)} = 2 - \frac{e^{-2}-2+2}{-e^{-2}+1}$ *=1.843482357* , $|x_1 - x_0|$ *=0.15652 ≤ 0.18538*

Primjer 4. Riješimo jednadžbu x^3- 2x + 1 = 0 Newton-Raphsonovom metodom s točnošću 0,0001.

Rješenje:

$f(x)=x^3-2x+1= 0$

Vidimo da je funkcija polinom trećeg stupnja tako da može imati jednu ili tri realne nultočke.

Prvo ćemo odrediti prvu i drugu derivaciju funkcije:

Prva derivacija:

$f'(x)=3x^2-2$ → funkcija je pozitivna kad *je* $x > \sqrt{\frac{2}{3}}$ i kad je $x < -\sqrt{\frac{2}{3}}$ dok je negativna između tih vrijednosti.

Druga derivacija:

$f''(x)=6x$ → druga derivacija je linearna pa funkcija prelazi iz negativne u pozitivnu

Funkcija *$f(x)=x^3-2x+1$* ima realno rješenje na intervalu [-2,2]. S obzirom da prva i druga derivacija moraju biti različite od nule, moramo suziti interval na $\left[0, \frac{4}{5}\right]$ a za početnu točku ćemo odabrati $x_0=0{,}5$.

Sada računamo m_1 i M_2.

Minimalna vrijednost m_1:

$f'(0) = 3 \cdot (0)^2 - 2 = 0 - 2 = -2$

$f'\left(\frac{4}{5}\right) = 3 \cdot \left(\frac{4}{5}\right)^2 - 2 = \frac{48}{25} - \frac{50}{25} = -\frac{2}{25}$

Minimalna apsolutna vrijednost je $\left|f'\left(\frac{4}{5}\right)\right| = \frac{2}{25}$, pa je $m_1 = \frac{2}{25}$

Maksimalna vrijednost M_2 :

$f''\left(\frac{4}{5}\right) = 6x = 6 \cdot \frac{4}{5} = \frac{24}{5}$

Maksimalna vrijednost na intervalu $\left[0, \frac{4}{5}\right]$ je u točki $x = \frac{4}{5}$, pa je $M_2 = \frac{24}{5}$

Dinamička ocjena pogreške:

$$|x_{n+1} - x_n| \leq \frac{M_2}{2m_1}|x_n - x_{n-1}|^2 = \frac{\frac{24}{5}}{2 \cdot \frac{2}{25}}|x_n - x_{n-1}|^2 = \frac{600}{20}|x_n - x_{n-1}|^2 = 30|x_n - x_{n-1}|^2$$

Iteracije:

$$f(0,5) = (0,5)^3 - 2 \cdot (0,5) + 1 = 0,125$$

$$f'(0,5) = 3 \cdot (0,5)^2 - 2 = -1,25$$

$$x_1 = 0,5 - \frac{0,125}{-1,25} = 0,6$$

$$f(0,6) = (0,6)^3 - 2 \cdot (0,6) + 1 = 0,016$$

$$f'(0,6) = 3 \cdot (0,6)^2 - 2 = -0,92$$

$$x_2 = 0,6 - \frac{0,016}{-0,92} = 0,6174$$

$$f(0,6174) = (0,6174)^3 - 2 \cdot (0,6174) + 1 = 0,0008$$

$$f'(0,6174) = 3 \cdot (0,6174)^2 - 2 = -0,8578$$

$$x_3 = 0,6174 - \frac{0,0008}{-0,8578} = 0,6183$$

Vidimo da Newtnova metoda brzo konvergira prema rješenju s zadanom točnošću. Konačno rješenje jednadžbe je $x = 0,6183$.

3. IZRADA WEB APLIKACIJE

U ovom poglavlju su prikazane tehnologije koje su korištene za izradu web aplikacije za rješavanje nelinearnih jednadžbi s jednom nepoznanicom jednom od numeričih metoda.

3.1. Korištene tehnologije

3.1.1. HTML

HTML (HyperText Markup Language) je jezik za stvaranje web stranica. Pomoću HTML-a oblikujemo sadržaj i stvaramo veze unutar web stranice. Jednostavan je za korištenje i brzo se uči, zato je jako popularan. HTML je besplatan i dostupan je svima što dodatno donosi njegovoj popularnosti.

Web preglednici omogućuju prikaz hipertekstualnih dokumenata koji se stvaraju pomoću HTML-a. Glavna uloga HTML-a je uputiti preglednik kako da prikažesadržaj stranice na ekranu, pri čemu se nastoji osigurati jednak prikaz bez obzira na korišteni preglednik,računalo ili operativni sustav.

HTML nije programski jezik i neomogućuje izvođenje operacija kao što su matematički izračuni. Umjesto toga, koristimo ga za opisivanje struktura i veza unutar naših web stranica. HTML datoteke su obično tekstualne datoteke s ekstenzijom .html.

Osnovni elementi svake HTML stranice su oznake (tags) koje definiraju kako će se elementi prikazati na web stranici. [6]

U ovom radu je koršten HTML5. Na slici 3.1. ispod je prikaz dio HTML koda.

```
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <a class="navbar-brand" href="#">Kalkulator za rješavanje nelinearnih jednadžbi</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle na
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item active">
                    <a class="nav-link" href="#">Početna <span class="sr-only">(current)</span></a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">O kalkulatoru</a>
                </li>
            </ul>
        </div>
    </nav>
    <div class="container">
        <h1 class="text-center">Rješavanje nelinearnih jednadžbi</h1>
        <p class="text-center">Kalkulator za rješavnje nelinearnih jednadžbi s jednom nepoznanicom numeričkim metodama.</p>
        <div class="row justify-content-center">
            <div class="col-md-6">
                <form id="equationForm">
                    <div class="form-group">
                        <label for="equationInput">Jednadžba:</label>
                        <input type="text" class="form-control" id="equationInput" placeholder="Unesite jednadžbu, npr. 2x^3-3">
                    </div>
                    <div class="form-group">
                        <label for="intervalInput">Interval:</label>
                        <input type="text" class="form-control" id="intervalInput" placeholder="Unesite interval, npr. 0,3">
```

Slika 3.1. *HTML kod od web aplikacije*

3.1.2. CSS

CSS(Cascading Style Sheets) je jezik za stiliziranje koji se koristi za opisivanje izgleda web stranica u HTM-u ili XML-u. Glavna uloga CSS-a je definirati kako će se elementi prikazivati na ekranu.

CSS je ključan jezik na otvorenom webu i standardiziran je prema specifikacijama W3C-a (World Wide Web Consortium).

Kroz CSS možemo kontrolirati boje, fontove, raspored elemenata i druge vizualne aspekte web stranica, čime pružamo korisnicima ugodno i konzistentno iskustvo pregledavanja. Primjena CSS-a omogućuje razdvajanje sadržaja od prezentacije, olakšavajući održavanje i poboljšavanje web stranice bez potrebe za mjenjanjem HTML-a. [7]

```
<style>
    body {
        background-color: #f8f9fa;
    }
    .container {
        margin-top: 50px;
    }
    .alert {
        margin-top: 20px;
    }
    #solutionDisplay {
        margin-top: 20px;
    }
    #plot {
        margin-top: 20px;
    }
    table {
        width: 100%;
    }
    th, td {
        text-align: center;
    }
</style>
```

Slika 3.2. *Primjer CSS koda*

3.1.3. JavaScript

JavaScript (JS) je lagani, interpretirani programski jezik s podrškom za funkcije prvog reda. Najpoznatiji je kao skriptni jezik za web stranice, ali se koristi i u mnogim drugim okruženjima izvan web preglednika, kao što su Node.js, Apache CouchDB i Adobe Acrobat. JavaScript je temeljen na prototipovima, podržava više paradigmi, dinamički jezik i podržava objektno orijentirani, imperativni te deklarativni stil programiranja.

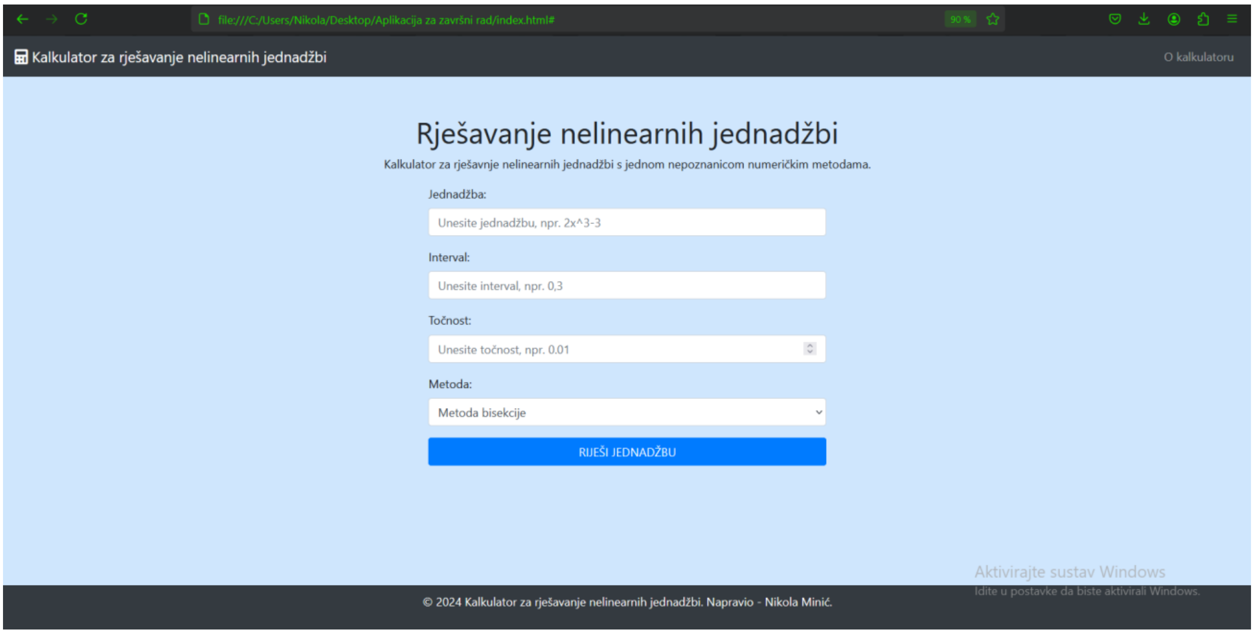
Važno je napomenuti da se JavaScript ne treba zamijeniti s programskim jezikom Java. Iako su oba naziva zaštitni znakovi Oracle-a, jezici imaju različitu sintaksu, semantiku i primjenu. JavaScript je ključan alat za moderni web razvoj i kontinuirano se prilagođava novim zahtjevima i tehnologijama. [8]

Za logiku web aplikacije koja se bavi rješavanjem nelinearnih jednadžbi korišten je JavaScript koji omogućuje obradu korisničkog unosa, izračune koji su potrebni za pronalazak rješenja jednadžbe te vizualizaciju rezultata.

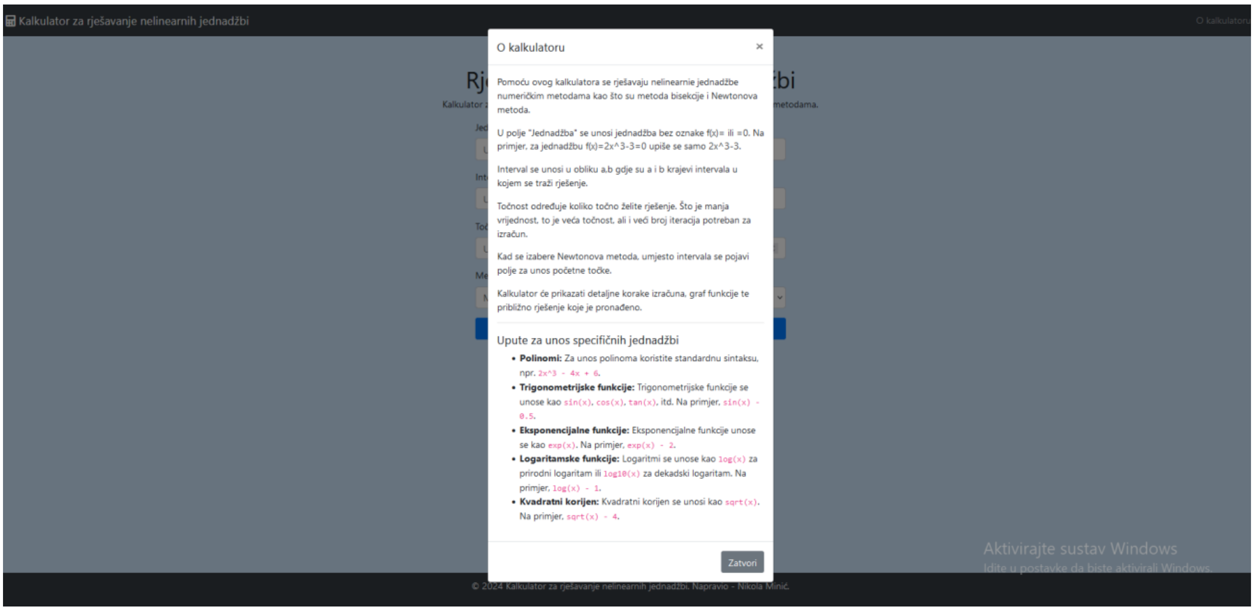
3.2. Opis aplikacije

U ovom poglavlju će biti opisana izrada web aplikacija. Struktura aplikacija se sastoji od datoteka index.html, u kojoj je html kod za polja za unos podataka, tekst na stranici i css koji daje izgled stranice i script.js u kome je napisana logika za rješavnje jednadžbi.

Na slici 3.2. ispod je prikazana početna stranica web aplikacije na kojoj imamo polja za unos jednadžbe, zadanog intervala, točnosti i mogućnost biranja metode rješavanja (Metoda bisekcije ili Newtnova metoda). U gornjem desnom kutu imamo „O kalkulatoru“, pritiskom na to pojaviće se prozor u kom je opisana upotreba kalkulatora što se vidi na slici.



Slika 3.2. *Početna stranica web aplikacije*



Slika 3.4. *Upute za upotrebu kalkulatora*

Slika 3.5. *DOMContentLoaded i prikaz stranice ovisno o izabranoj metodi*

Na slici 3.5. je prikazan dio koda koji je zadužen za događaje i logiku web stranice tj. korisnik može unijeti jednadžbu i odabrati numeričku metodu za računanje jednadžbe. Koristi se 'DOMContentLoaded' event listener koji osigurava da se skripta ne pokrene prije nego se učita cijela stranica. Konstante 'form', 'methodSelect', 'intervalGroup', 'initialGuessGroup' se koriste kao reference HTML elemenata na stranici, formu, padajući izbornik za izabir metode, polje za unos intervala i polje za unost početne vrijednosti.

Event listener na metodi 'methodSelect' služi da prikaže ili sakrije određena polja za unos ovisno o izabranoj metodi. Ako korisnik odabere Newtonovu metodu u sučelju će biti prikazano polje za unos početne točke, dok će se polje za unos intervala sakriti. Suprotno je ako se odabere metoda bisekcije.

```javascript
form.addEventListener('submit', function(event) {
    event.preventDefault();

    let equationStr = document.getElementById('equationInput').value;
    const tolerance = parseFloat(document.getElementById('toleranceInput').value);
    const method = document.getElementById('methodSelect').value;

    equationStr = equationStr.replace(/f\(x\) *= */g, '');
    equationStr = equationStr.replace(/= *0 */g, '');

    console.log("Unesena jednadžba:", equationStr);
    console.log("Unesena točnost:", tolerance);
    console.log("Odabrana metoda:", method);

    try {
        const equation = math.compile(equationStr);
        const derivative = math.derivative(equationStr, 'x').compile();

        let solution;
        if (method === 'bisection') {
            const intervalStr = document.getElementById('intervalInput').value;
            const [a, b] = intervalStr.split(',').map(Number);
            if (isNaN(a) || isNaN(b)) {
                displayError('Pogrešan unos intervala. Provjerite da su oba kraja intervala brojevi.');
                return;
            }
            solution = bisectionMethod(equation, a, b, tolerance);
        } else if (method === 'newton') {
            const initialGuess = parseFloat(document.getElementById('initialGuessInput').value);
            solution = newtonMethod(equation, derivative, initialGuess, tolerance);
        }

        displaySolution(solution, equation, method);
    } catch (error) {
        console.error("Greška:", error);
        displayError('Pogreška pri parsiranju jednadžbe: ' + error.message);
    }
});
```

Slika 3.6. *Event listener za obradu unesenih podataka*

Na slici 3.6 vidimo dio koda koji služi kada korisnik unese jednadžbu i odabere metodu, da obradi i pravilno parsira jednadžbu i ukloni nepotrebne dijelove ako su uneseni *'f(x) ='* ili *'= 0'*. Za kompajliranje i evaluiranje jednadžbe i izračune derivacija koristi se Math.js biblioteka.

```javascript
function bisectionMethod(equation, a, b, tol) {
  let fa = equation.evaluate({ x: a });
  let fb = equation.evaluate({ x: b });
  if (fa * fb > 0) {
      return 'Funkcija nema nultu točku u zadanom intervalu.';
  }

  let steps = [];
  let iteration = 0;
  while ((b - a) / 2 > tol) {
      const c = (a + b) / 2;
      const fc = equation.evaluate({ x: c });

      steps.push({ iteration, a, b, c, x: c, fx: fc });

      if (fc === 0) {
          return steps;
      } else if (fa * fc < 0) {
          b = c;
      } else {
          a = c;
          fa = fc;
      }
      iteration++;
  }

  const result = (a + b) / 2;
  steps.push({ iteration, a, b, c: result, x: result, fx: equation.evaluate({ x: result }) });
  return steps;
}
```

Slika 3.7. *Metoda bisekcije*

Na slici 3.7. vidimo kod za računanje pomoću metode bisekcije.

Metoda 'bisectionMethod' prima parametre: funkciju, krajeve intervala i točnost. Na početku odmah provjerava da li postoji nultočka u zadanom intervalu, ako rješenje postoji funkcija kreće sa iteracijama. Pomoću while petlje radi iteracije sve dok se nezadovolji uvjet. Metoda prvo računa vrijednosti funkcije na krajevima intervala. Ako je rezultat fa * fb veći od 0 to znači da nultočka nije u tom intervalu i funkcija vraća poruku: „Funkcija nema nultočku u zadanom intervalu“. Ako ne, onda metoda ulazi u while petlju koja se izvodi sve dok je polovina intervala veća od zadane točnosti. U svakoj iteraciji se računa srednja vrijednost intervala c i vrijednost funkcije fc. Ako je fc = 0 to znači da je pronađena nultočka i metoda ispisuje iteracije. Ako je fa*fc < 0, nultočka je između a i c pa se granica b postavlja na c ili suprotno nultočka je između c i b pa se granica a postavlja na c, a fa na fc. Petlja se zaustavlja kad je razlika između a i b manja od zadane točnosti i metoda ispisuje konačno rješenje, iteracije.

```javascript
function newtonMethod(equation, derivative, x0, tol) {
  try {
    let x = x0;
    let fx = equation.evaluate({ x });
    let dfx = derivative.evaluate({ x });
    let iteration = 0;
    let steps = [];

    while (Math.abs(fx) > tol && iteration < 1000) {
      if (dfx === 0) {
        return 'Derivacija je nula, metoda se ne može primijeniti.';
      }

      const x1 = x - fx / dfx;
      steps.push({ iteration, x, fx, dfx });

      x = x1;
      fx = equation.evaluate({ x });
      dfx = derivative.evaluate({ x });
      iteration++;
    }

    steps.push({ iteration, x, fx, dfx });
    return steps;
  } catch (error) {
    console.error("Greška u Newtonovoj metodi:", error);
    throw new Error('Greška pri računanju Newtonove metode: ' + error.message);
  }
}
```

Slika 3.8. *Newtnova metoda*

Na slici 3.8 vidimo kod za računanje pomoću newtnove metode.

Newtonova metoda kao ulazne parametre prima funkciju, prvu derivaciju, početnu točku x_0 i točnost. Početna točka se postavlja na vrijednost x_0, izračunava se vrijednost funkcije za x i derivacija funkcije. Broj iteracija se postavlja na 0 i stvara se niz koji će čuvati podatke o iteracijama. Metoda ima while petlju koja se izvršava sve dok je apsolutna vrijednost f(x) veća od točnosti i dok je broj iteracija manji od 1000. Prvo se provjerava da li je derivacija jednaka 0, ako je ispisuje se poruka „Derivacija je nula, metoda se ne može primjeniti.“. Ako nije 0 izračunava se nova aproksimacija x_1. Koraci se dodaju u niz i vrijednosti se ažuriraju. Petlja se zaustavlja kad je uvjet zadovoljen.

```
function displaySolution(steps, equation, method) {
    const solutionDisplay = document.getElementById('solutionDisplay');
    if (typeof steps === 'string') {
        solutionDisplay.innerHTML = `
            <div class="alert alert-danger" role="alert">
                ${steps}
            </div>
        `;
    } else {
        let tableRows;
        if (method === 'bisection') {
            tableRows = steps.map(step => `
                <tr>
                    <td>${step.iteration}</td>
                    <td>${step.a.toFixed(6)}</td>
                    <td>${step.b.toFixed(6)}</td>
                    <td>${step.c.toFixed(6)}</td>
                    <td>${step.fc.toFixed(6)}</td>
                </tr>
            `).join('');
        } else if (method === 'newton') {
            tableRows = steps.map(step => `
                <tr>
                    <td>${step.iteration}</td>
                    <td>${step.x !== undefined ? step.x.toFixed(6) : ''}</td>
                    <td>${step.fx !== undefined ? step.fx.toFixed(6) : ''}</td>
                    <td>${step.dfx !== undefined ? step.dfx.toFixed(6) : ''}</td>
                </tr>
            `).join('');
        }

        const finalResult = steps[steps.length - 1];
        const finalResultText = finalResult.c ? finalResult.c.toFixed(6) : finalResult.x.toFixed(6);
```

Aktiviraj
Idite u post

```
        solutionDisplay.innerHTML = `
            <div class="alert alert-success" role="alert">
                <h4 class="alert-heading">Konačno rješenje</h4>
                <p>Rješenje je približno: ${finalResultText}</p>
                <hr>
                <table class="table table-striped table-hover">
                    <thead>
                        <tr>
                            <th>Iteracija</th>
                            ${method === 'bisection' ? '<th>a</th><th>b</th><th>c</th><th>f(c)</th>' : '<th>x</th><th>f(x)</th><th>f\'(x)</th>'}
                        </tr>
                    </thead>
                    <tbody>
                        ${tableRows}
                    </tbody>
                </table>
            </div>
        `;

        plotFunction(equation, steps, method);
    }
}
```

Slika 3.9. *Prikaz rješenja*

Na slici 3.9 vidimo funkciju koja prikazuje rješenja jednadžbi. Ako funkcija dobije string, tj. poruku za grešku prikaže se crveni alert s porukom greške, a ako dobije niz koraka iteracija prikazuje se tablica u kojoj su detalji svake iteracije i prikazuje se konačno rješenje kao zeleni alert. Na kraju se poziva funkcija za crtanje grafa funkcije.

.

```
function displayError(message) {
  const solutionDisplay = document.getElementById('solutionDisplay');
  solutionDisplay.innerHTML = `
    <div class="alert alert-danger" role="alert">
      ${message}
    </div>
  `;
}
```

Slika 3.10. *Prikaz greške*

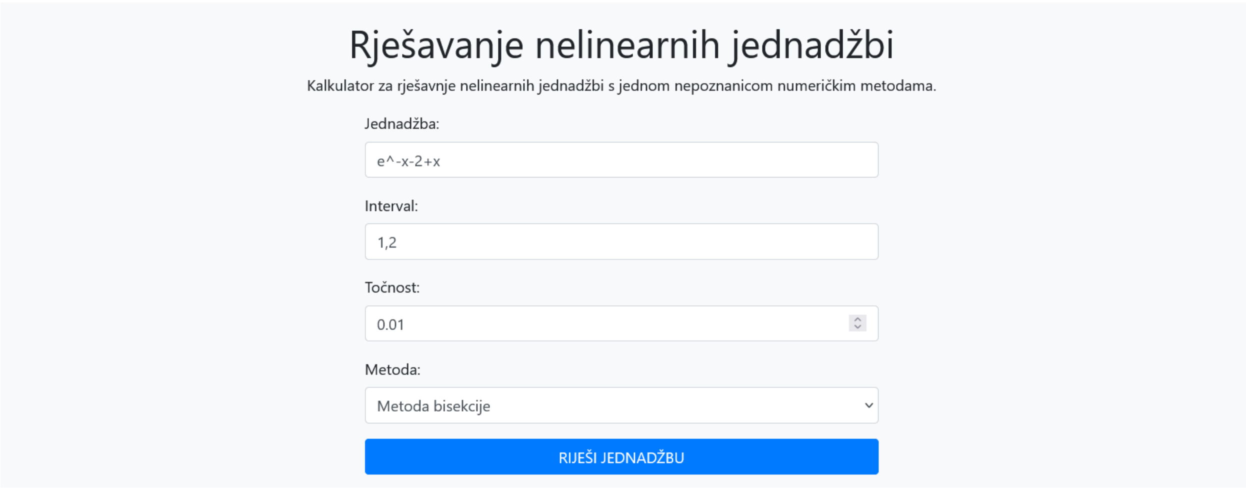
Ova funkcija prikazuje poruku greške u crvenom alert box-u. Koristi se za prikaz poruka grešaka koje se javljaju tijekom parsiranja jednadžbe ili izvođenja metoda.

```
function plotFunction(equation, root) {
  const xValues = math.range(-10, 10, 0.1).toArray();
  const yValues = xValues.map(x => equation.evaluate({ x }));

  const trace = {
    x: xValues,
    y: yValues,
    mode: 'lines',
    type: 'scatter',
    name: 'f(x)',
    line: { color: 'blue' }
  };

  const rootTrace = {
    x: [root],
    y: [equation.evaluate({ x: root })],
    mode: 'markers',
    marker: { color: 'red', size: 12 },
    name: 'Rješenje'
  };

  const layout = {
    title: 'Graf funkcije',
    xaxis: { title: 'x' },
    yaxis: { title: 'f(x)' }
  };

  Plotly.newPlot('plot', [trace, rootTrace], layout);
}
```

Slika 3.11. *Crtanje grafa*

Na slici 3.11 je metoda za crtanje grafa funkcije. Prikazuje graf funkcije zajedno sa iteracijama i konačnim rezultatom. Za vizualizaciju se koristi biblioteka Ploty koja omogućuje interaktivni prikaz.

4. ANALIZA I REZULTATI



Slika 4.1. *Primjer unesene jednadžbe, intervala i točnosti*

Na slici 4.1. vidimo primjer unesene jednadžbe, intervala, točnosti i metode kojom će se računati.

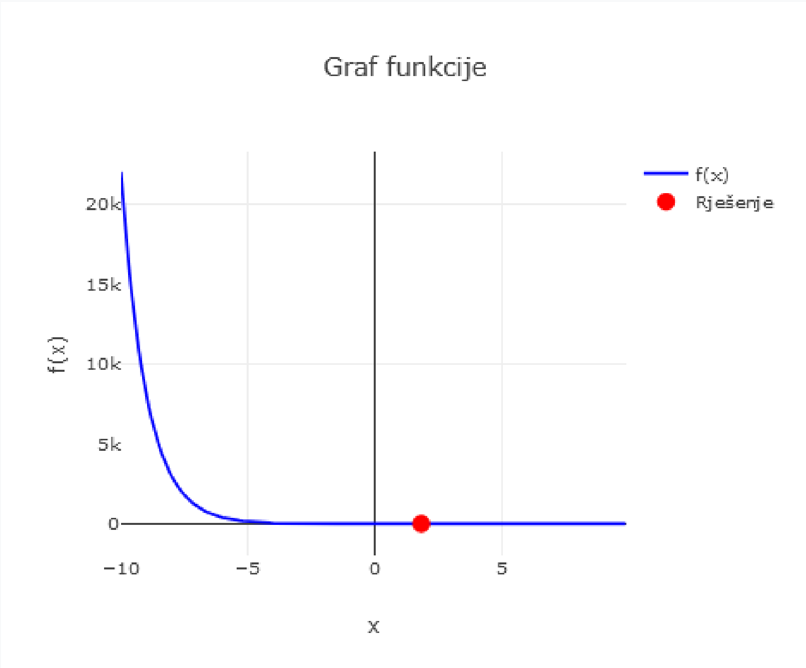
Kada se unesu potrebni podaci i klikom na plavi gumb „riješi jednadžbu“ aplikacija će izračunati konačno riješenje i prikazati tablicu sa iteracijama ovisno koju smo metodu odabrali i ispod graf funkcije što je prikazano na slici 4.2.

Primjenom metode bisekcije na jednadžbu $e^{-x} - 2 + x = 0$ unutar intervala [1,2] s točnošću 0.01 aplikacija je konvergirala na rješenje x = 1.835938 kroz 7 iteracija. Newtonova metoda, s početnom pretpostavkom x_0 = 1.5, postigla je isto rješenje kroz samo 3 iteracija, što pokazuje njenu učinkovitost u bržoj konvergenciji kada su zadovoljeni uvjeti za njenu primjenu.

Konačno rješenje

Rješenje je približno: 1.835938

Iteracija	a	b	c	f(c)
0	1.000000	2.000000	1.500000	-0.276870
1	1.500000	2.000000	1.750000	-0.076226
2	1.750000	2.000000	1.875000	0.028355
3	1.750000	1.875000	1.812500	-0.024254
4	1.812500	1.875000	1.843750	0.001973
5	1.812500	1.843750	1.828125	-0.011160
6	1.828125	1.843750	1.835938	-0.004599



Slika 4.2. *Prikaz konačnog rješenja, tablice i grafa funkcije*

5. ZAKLJUČAK

U ovom završnom radu su analizirane nelinearne jednadžbe s jednom nepoznanicom i njihovo rješavanje numeričkim metodama i na osnovu toga je napravljena web aplikacija za rješavanje istih. U radu je opisana teorija, korištene tehnologije i analiza rješenja.

Implementacija aplikacije omogućuje korisnicima rješavanje nelinearnih jednadžbi koristeći metodu bisekcije i Newtonovu metodu. Korisnici unose jednadžbu, interval i toleranciju, a aplikacija automatski parsira jednadžbu, izračunava derivaciju (za Newtonovu metodu), te prikazuje korake iteracija i konačno rješenje.

Obje metode imaju svoje prednosti: Metoda bisekcije je stabilnija i uvijek konvergira unutar intervala gdje funkcija mijenja predznak, dok Newtonova metoda brže konvergira ali zahtijeva dobru početnu pretpostavku i ne-nultu derivaciju. Preporučuje se korištenje metode bisekcije za jednostavnije jednadžbe ili kada je interval poznat, dok je Newtonova metoda pogodnija za jednadžbe gdje je derivacija dobro definirana i poznata početna točka je blizu rješenja.

LITERATURA

[1] 6. Nelinearne jednadzbe i sustavi Sveučilište u Zagrebu

[2] Rješavanje nelinearnih jednadžbi , Sveučilište u Zagrebu (https://web.math.pmf.unizg.hr/nastava/unm/skripta/html/06a_rjesavanje_nelinearnih_jednadzbi.html) [2.9.2024]

[3] Non-linear equations HELM (2008) , University of Sheffield (https://www.sheffield.ac.uk/) [2.9.2024]

[4] Približno rješavanje nelinearnih jednadžbi metoda bisekcije, Marijan Čančarević, Nataša Lončarić (https://hrcak.srce.hr/file/161432) [2.9.2024]

[5] Metoda bisekcije (https://www.scribd.com/doc/225074187/bisekcija) [2.9.2024]

[6] Newtnova metoda (https://www.scribd.com/doc/225074187/bisekcija) [2.9.2024]

[7] HTML (https://en.wikipedia.org/wiki/HTML) [2.9.2024]

[8] CSS (https://developer.mozilla.org/enUS/docs/Learn/Getting_started_with_the_web/CS_basics) [2.9.2024]

[9] JavaScript (https://developer.mozilla.org/en-US/docs/Web/JavaScript) [2.9.2024]

SAŽETAK

U ovom završnom radu je bio cilj istražiti numeričke metode za rješavanje nelinearnih jednadžbi s jednom nepoznanicom i izraditi web aplikaciju za rješavanje istih. Aplikacija omogućuje jednostavno i precizno rješavanje takvih jednadžbi. Za izradu same aplikacije su korištene tehnologije HTML, CSS i JavaScript. Pomoću JavaScripta je riješena logika rješavanja jednadžbi sa metodom bisekcije i Newtnovom metodom, ispis rješenja, grafički prikaz... , a HTML i CSS su korišteni za izgled aplikacije. U radu je opisana upotreba aplikacije i prikazani su primjeri riješenih zadataka.

Ključne riječi: nelinearne jednadžbe, metoda bisekcije, Newtnova metoda, web aplikacija

ABSTRACT

WEB APPLICATION FOR SOLVING NONLINEAR EQUATIONS WITH ONE UNKNOWN USING NUMERICAL METHODS

This thesis aims to explore numerical methods for solving nonlinear equations with one unknown and develop a web application for solving such equations. The application allows for simple and accurate solutions to these types of equations. The technologies used to develop the application include HTML, CSS, and JavaScript. JavaScript was employed to implement the logic for solving equations using the bisection method and Newton's method, as well as for displaying solutions and graphical representations, while HTML and CSS were used for the application's design. The thesis describes the use of the application and presents examples of solved problems.

Keywords: bisection method, Newton's method, nonlinear equations, web application

____

(Potpis studenta)

PRILOG 1 – REPOZITORIJI NA GITHUB-u

https://github.com/nminic02/Zavrsni-rad

PRILOG 2 – WEB APLIKACIJA

https://nikola-minic.netlify.app/