Numeričko deriviranje, primjene i implementacija u Pythonu

Vratarić, Matej

Sveučilište u Rijeci

Fakultet za matematiku

Sveučilišni prijediplomski studij Matematika

Matej Vratarić

Numeričko deriviranje, primjene i implementacija u Pythonu

Završni rad

Rijeka, srpanj 2024.

Sveučilište u Rijeci

Fakultet za matematiku

Sveučilišni prijediplomski studij Matematika

Matej Vratarić

Numeričko deriviranje, primjene i implementacija u Pythonu

Završni rad

Mentor: doc. dr. sc. Sanda Bujačić Babić

Rijeka, srpanj 2024.

Sažetak

Ovaj završni rad uvodi osnovnu teoriju numeričkog deriviranja, izvodi neke osnovne metode i pristupe numeričkog deriviranja, opisuje njihovu točnost, te ih implementira u programskom jeziku Python. Kroz matematičke izvode, primjere i programski kod, daje se uvid u efikasnost, pouzdanost i primjenjivost numeričkog deriviranja.

Ključne riječi: numeričko deriviranje, analiza podataka, podijeljene i konačne razlike, analiza greške

Sadržaj

1 Uvod i vrste grešaka

U matematici se često susrećemo sa situacijama u kojima egzaktna rješenja nisu moguća ili je jednostavno njihov postupak rješavanja krajnje nepraktičan. Tu do izražaja dolaze numeričke metode, nudeći postupke kojima smo u mogućnosti dobiti zadovoljavajuće aproksimacije rješenja, a numerička matematika je grana matematike koja se bavi istraživanjem i razvijanjem te optimizacijom takvih metoda. Numeričku matematiku karakteriziraju aproksimativne metode koje se koriste za najrazličitije probleme: od interpolacije funkcije do određivanja rješenja diferencijalnih jednadnadžbi. Jedna od poznatih i vrlo korištenih zadaća numeričke matematike je numeričko, odnosno približno deriviranje funkcije u točki, i to najčešće u onim slučajevima u kojima je egzaktno deriviranje neizvedivo ili analitički zahtjevno, odnosno u onim slučajevima u kojima nemamo zadanu funkciju, već samo diskretan skup točaka koje nepoznata funkcija zadovoljava. Za mnoge probleme u numeričkoj matematici nerijetko postoji i više različitih metoda koje onda često nisu sve uvijek jednako precizne. Naime, vrlo često u ovisnosti o prirodi problema, biramo najpogodniju metodu tako da ona ima najmanju grešku. Neki od osnovnih pojmova u numeričkoj matematici stoga su apsolutna i relativna greška koje nam daju uvid u korektnost konstruiranih metoda:

Definicija 1. *Neka je* $a \in \mathbb{R}\backslash\{0\}$ *te* $\overline{a}$ *aproksimacija vrijednosti* a*. Broj* $|\overline{a} - a|$ *nazivamo apsolutnom, te* $\frac{|\overline{a}-a|}{|a|}$ *relativnom greškom aproksimacije* $\overline{a}$*.*

Uočimo da relativna greška daje nešto bolju informaciju o aproksimaciji jer daje postotak odstupanja aproksimacije vrijednosti od egzaktne vrijednosti, a ne apsolutnu vrijednost razlike stvarne i aproksimirane vrijednosti što je slučaj kod apsolutne greške.

Primjer 1. *Broj* π *je iracionalan broj i svaki prikaz broja* π *s konačno mnogo decimala je aproksimacija njegove stvarne vrijednosti. Na primjer,* 3.14 *je aproksimacija broja* π *s apsolutnom greškom približno jednakom* $|\pi - 3.14| \approx 0.00159265$ *te relativnom greškom* $\frac{|\pi-3.14|}{\pi} \approx 0.000506957$*. U ovom slučaju vrijednost aproksimacije odstupa od stvarne vrijednosti za približno* 0.05%*.*

Konačna aritmetika računala i greška zaokruživanja

Jednu od ključnih uloga u numeričkoj matematici imaju računala. Njihovom uporabom omogućeno nam je učinkovitije rješavanje složenijih i vremenski zahtjevnijih zadataka, u okviru kojih je uglavnom riječ o metodama koje karakterizira velik broj koraka, odnosno operacija potrebnih za dostizanje prihvatljive točnosti. Uz sve prednosti koje postoje u upotrebi računala, iznimno je važno napomenuti i izazove koje donosi uporaba ograničenih resursa računalnih sustava. Prilikom spominjanja računalnih ograničenja koja postoje u današnjim računalima, najčešće se spominje limitirani raspon brojeva koji se pohranjuju u računalu te činjenica da se zbog računalne arhitekture može pohraniti samo konačan broj brojeva. Naime, prikaz realnih brojeva na računalu sastoji se od tri dijela i svaki od njih ima svoju duljinu, odnosno broj bitova (oznaka b) predviđenih za prikaz: predznak s, koji uvijek zauzima jedan bit, signikant f koji zauzima sljedećih t bitova i eksponent e. [1]

Primjer 2. *Sljedeća tablica opisuje 32-bitne i 64-bitne računalne sustave te odgovarajuće preciznosti:*

tip	binary32	binary64
duljina	$32b$	$64b$
s	$1b$	$1b$
e	$8b$	$11b$
t	$23b$	$52b$
ULP	2^{-24}	2^{-53}
raspon	$10^{\pm 38}$	$10^{\pm 308}$

Tablica 1: IEEE-754 standard. [1]

ULP (eng. Unit of Least Precision) je jedinična strojna greška koju interpretiramo kao razmak između dva "susjedna" broja koja su prikaziva u računalu. Svaki broj koji nije moguće egzaktno reprezentirati u računalu zaokružuje se na najbliži broj koji se može reprezentirati. Jasno je da tako zaokružen broj mora imati grešku manju od 1/2 ULP-a. [1]

Primjer 3. *Brojeve u bazi 2 reprezentiramo putem izraza* $(-1)^s \cdot f \cdot 2^e$. *Neka je na primjer* $t = 5$. *Tada* f *može poprimiti vrijednosti između 1.00000 i 1.11111, a najmanja promjena moguća je 0.00001.*

Katastrofalno kraćenje

Katastrofalno kraćenje događa se u slučaju kada za neke $x, y \in \mathbb{R}$ vrijedi

$$|x - y| \ll |x|, |y|,$$

odnosno kada je rezultat oduzimanja brojeva istog predznaka broj koji je po apsolutnoj vrijednosti puno manji od ulaznih podataka.

Uzmimo proizvoljne $x, y \in \mathbb{R}$ istoga predznaka. Definiramo:

$$\begin{aligned} \epsilon_x &:= \frac{\overline{x} - x}{x} \implies \overline{x} = x(1 + \epsilon_x), \\ \epsilon_y &:= \frac{\overline{y} - y}{y} \implies \overline{y} = y(1 + \epsilon_y). \end{aligned}$$

Sada vrijedi:

$$\begin{aligned} \overline{x} - \overline{y} &= x(1 + \epsilon_x) - y(1 + \epsilon_y) \\ &= x - y + (x - y)\frac{x\epsilon_x - y\epsilon_y}{x - y} \\ &= (x - y)(1 + \frac{x\epsilon_x - y\epsilon_y}{x - y}). \end{aligned}$$

Dakle, relativna greška operacije oduzimanja je:

$$\epsilon := |\frac{x\epsilon_x - y\epsilon_y}{x - y}| \leq |\frac{x}{x - y}||\epsilon_x| + |\frac{y}{x - y}||\epsilon_y|,$$

iz čega je uočljivo da zbog $|x - y| \ll |x|, |y|$ izrazi

$$|\frac{x}{x - y}|, |\frac{y}{x - y}|$$

mogu biti proizvoljno veliki pa je relativna greška operacije oduzimanja proizvoljno velika. [1]

Primjer 4. *Promatrajmo aritmetiku "računala" koje sprema brojeve u bazi 10 (umjesto u bazi 2). Pretpostavimo da to "računalo" zaokružuje sve brojeve na tri decimale, odnosno da je eksponent 3. Uzmimo sada* $x = 8.8866 \cdot 10^0$ *i* $y = 8.8844 \cdot 10^0$. *Neka naše računalo iz ovog primjera ta dva broja pohranjuje kao* $\overline{x} = 8.887 \cdot 10^0$, *i* $\overline{y} = 8.884 \cdot 10^0$. *Uočimo sada:*

$$\overline{x} - \overline{y} = 8.8866 \cdot 10^0 - 8.8844 \cdot 10^0 = 0.003 \cdot 10^0 = 3.??? \cdot 10^{-3},$$

dok je

$$x - y = 8.8866 \cdot 10^0 - 8.8844 \cdot 10^0 = 0.0022 \cdot 10^0 = 2.2 \cdot 10^{-3}. \ [1]$$

Primjetimo da je relativna greška ove aproksimacije vrlo velika.

2 Numeričko deriviranje

Deriviranje funkcija ima široku primjenu u raznim znanstvenim i tehničkim granama: kod rješavanja diferencijalnih jednadžbi, određivanja minimuma ili maksimuma funkcija, kod analize stopa promjena cijena, kamatnih stopa i drugih ekonomskih pokazatelja. Prije nego objasnimo metodu numeričkog deriviranja, nužno je uvesti par osnovnih definicija.

Definicija 2. *Za skup $U \subseteq \mathbb{R}$ kažemo da je otvoren ako za svaki $x \in U$ postoji $\epsilon > 0$ t.d. $\{r \in \mathbb{R} : |r - x| < \epsilon\} \subseteq U$. Otvoreni podskup U od $\mathbb{R}$ označavamo s $U \overset{otv.}{\subseteq} \mathbb{R}$.*

Definicija 3. *Neka je $f : U \overset{otv.}{\subseteq} \mathbb{R} \to \mathbb{R}$ neka proizvoljna funkcija. Kažemo da je ta funkcija diferencijabilna u točki $a \in U$ ako postoji limes*

$$\lim_{h \to 0} \frac{f(a+h) - f(a)}{h},$$

kojeg u tom slučaju označavamo s $f'(a)$ te ga nazivamo derivacijom funkcije f u točki a. Funkcija f je diferencijabilna ako je ona diferencijabilna u svakoj točki domene.

Definicija 4. *Realna funkcija f realne varijable je klase C^k, za $k \in \mathbb{N}_0$ ako postoji k-ta neprekidna derivacija f^k.*

Definicija 5. *Za proizvoljnu funkciju f neka je $f[x] := f(x)$ za sve x iz domene te funkcije. Za čvorove $(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_n, f(x_n))$ uvodimo podijeljenu razliku n-tog reda funkcije f kao:*

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0}.$$

Podijeljene razlike možemo na pregledan način zapisivati i iščitavati uz pomoć tablice:

$$
\begin{array}{llllll}
x_0 & f[x_0] & & & & \\
 & & f[x_0,x_1] & & & \\
x_1 & f[x_1] & & f[x_0,x_1,x_2] & & \\
 & & f[x_1,x_2] & & & \\
x_2 & f[x_2] & & f[x_1,x_2,x_3] & & \\
 & & f[x_2,x_3] & \vdots & \dots & f[x_0,x_1,\dots,x_n] \\
x_3 & f[x_3] & \vdots & & & \\
\vdots & & & f[x_{n-2},x_{n-1},x_n] & & \\
 & \vdots & f[x_{n-1},x_n] & & & \\
x_n & f[x_n] & & & &
\end{array}
$$

Tablica 2: Podijeljene razlike. [1]

Ekvidistantnim čvorovima zovemo skup čvorova kod kojih je razlika vrijednosti x-koordinata između susjednih čvorova konstantna. Promotrimo podijeljene razlike u slučaju kada imamo takve čvorove. Njih možemo zapisati u obliku:

$$x_{i+1} = x_i + h, \ \ i = 0, \dots, n-1,$$

gdje je h konstantan razmak (korak) između dva susjedna čvora. U ovom slučaju podijeljene razlike zovemo konačne razlike i određujemo ih rekurzivno:

$$\Delta^k f(x_i) = \Delta^{k-1} f(x_{i+1}) - \Delta^{k-1} f(x_i), \ \ k = 1, \dots, n, \quad \Delta^0 f(x_i) := f(x_i).$$

Vrijedi:

$$
\begin{array}{llllll}
x_0 & f(x_0) & & & & \\
 & & \Delta f(x_0) & & & \\
x_1 & f(x_1) & & \Delta^2 f(x_0) & & \\
 & & \Delta f(x_1) & & & \\
x_2 & f(x_2) & & \Delta^2 f(x_1) & & \\
 & & \Delta f(x_2) & \vdots & \dots & \Delta^n f(x_0) \\
x_3 & f(x_3) & \vdots & & & \\
\vdots & & & \Delta^2 f(x_{n-2}) & & \\
 & \vdots & \Delta f(x_{n-1}) & & & \\
x_n & f(x_n) & & & &
\end{array}
$$

Tablica 3: Konačne razlike. [1]

Sljedeća lema nam daje poveznicu između podijeljenih i konačnih razlika i znatno skraćuje postupak računanja podijeljenih razlika i izbjegava potencijalnu propagaciju greške koja nastaje prilikom dijeljenja, dakako samo u slučaju kada su čvorovi ekvidistantni.

Lema 1. *Za ekvidistantne čvorove* $(x_i, f(x_i))$, $x_{i+1} - x_i = h$, $i = 0, 1, \ldots, n-1$ *vrijedi:*

$$f[x_0, \ldots, x_n] = \frac{\Delta^n f(x_0)}{n! h^n}.[1]$$

Lema 1 se dokazuje indukcijom uz korištenje nekih od navedenih jednakosti.

2.1 Newtonov interpolacijski polinom

Pod pojmom interpolacije podrazumijevamo zahtjev da tražena funkcija mora zadovoljavati određeni fiksni broj točaka (čvorova). U tom slučaju te čvorove nazivamo interpolacijskim čvorima, a tu funkciju nazivamo funkcijom interpolacije. Interpolacijske funkcije mogu biti razne: polinomi, splajnovi, itd., a interpolacijski polinom je polinom koji je određen tako da interpolacijski čvorovi zadovoljavaju jednadžbu polinoma. Jedna forma takvog polinoma je Newtonov interpolacijski polinom.
Neka je $p_{n-1}(x)$ interpolacijski polinom koji interpolira funkciju f kroz čvorove $(x_i, f(x_i))$, $i = 0, \ldots, n-1$. Ako želimo dodati još jedan čvor, x_n, tada ćemo dobiti interpolacijski polinom $p_n(x)$ koji interpolira funkciju $f(x)$ u čvorovima x_i, $i = 0, \ldots, n$, odnosno:

$$p_n(x) = p_{n-1}(x) + c(x), \tag{1}$$

gdje moramo odrediti polinom n-tog stupnja $c(x)$. Taj polinom mora biti oblika:

$$c(x) = a_n(x - x_0)(x - x_1) \ldots (x - x_{n-1}).$$

Definicija 6. *Za funkciju* $f(x)$ *te čvorove* (x_i, y_i), $i = 0, 1, \ldots, n$, *polinom*

$$P_n(x) = f[x_0] + (x - x_0) f[x_0, x_1] + \cdots + (x - x_0) \ldots (x - x_{n-1}) f[x_0, \ldots, x_n]$$

naziva se Newtonov interpolacijski polinom funkcije f *kroz* (x_i, y_i), $i = 0, 1, \ldots, n$.

Sada ako postavimo $a_n := f[x_0, \ldots, x_n]$, polinom iz Definicije 6 odgovara uvjetu (1). Može se pokazati da takav polinom uvijek postoji i jedinstven je za zadani skup čvorova.

Primjer 5. *Odredimo Newtonov interpolacijski polinom kroz čvorove* $(x_0, y_0) = (1, 2), (x_1, y_1) = (2, 3), (x_2, y_2) = (3, 5), (x_3, y_3) = (4, 9)$. *Izračunajmo podijeljene razlike:*

$$f[x_0, x_1] = \frac{y_1 - y_0}{x_1 - x_0} = \frac{3-2}{2-1} = 1, \quad f[x_1, x_2] = \frac{5-3}{3-2} = 2, \quad f[x_2, x_3] = \frac{9-5}{4-3} = 4,$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{2-1}{3-1} = \frac{1}{2}, \quad f[x_1, x_2, x_3] = \frac{4-2}{4-2} = 1,$$

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} = \frac{1 - \frac{1}{2}}{4-1} = \frac{1}{6},$$

te uvrstimo:

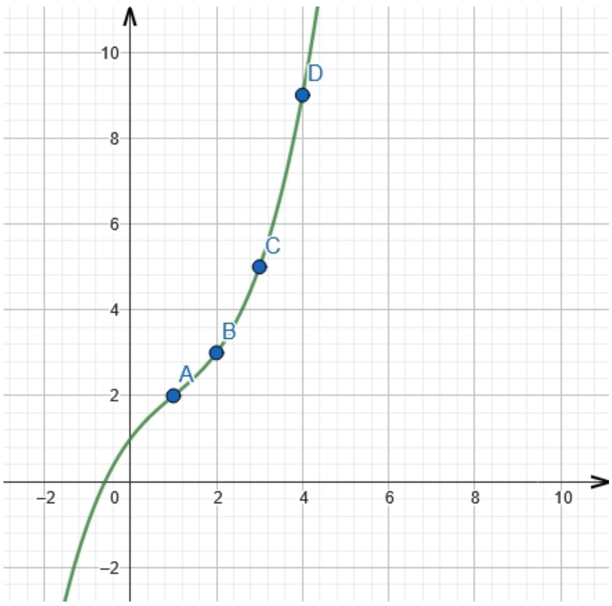
$$p_3(x) = 2 + (x-1)1 + (x-1)(x-2)\frac{1}{2} + (x-1)(x-2)(x-3)\frac{1}{6} = \frac{1}{6}x^3 - \frac{1}{2}x^2 + \frac{4}{3}x + 1$$

Kako je riječ o ekvidistantnim čvorovima $(h = 1)$ *mogli smo primjeniti i Lemu 1 te na taj način dobiti potpuno iste vrijednosti koeficijenata:*

$$f[x_0, x_1] = \frac{\Delta y_0}{1! \cdot 1} = y_1 - y_0 = 3 - 2 = 1, \; f[x_1, x_2] = y_2 - y_1 = 5 - 3 = 2, \; f[x_2, x_3] = 9 - 5 = 4,$$

$$f[x_0, x_1, x_2] = \frac{\Delta^2 y_0}{2! \cdot 1} = \frac{1}{2}(\Delta y_1 - \Delta y_0) = \frac{1}{2}(2-1) = \frac{1}{2}, \quad f[x_1, x_2, x_3] = \frac{1}{2}(4-2) = 1,$$

$$f[x_0, x_1, x_2, x_3] = \frac{\Delta^3 y_0}{3! \cdot 1} = \frac{1}{6}(\Delta^2 y_1 - \Delta^2 y_0) = \frac{1}{6}(2-1) = \frac{1}{6}.$$



Slika 1: Newtonov interpolacijski polinom $y = \frac{1}{6}x^3 - \frac{1}{2}x^2 + \frac{4}{3}x + 1$.

Vrlo je važan i sljedeći teorem koji daje izgled funkcije greške interpolacijskog polinoma:

Teorem 1. *Pretpostavimo da za funkciju $f(x)$ postoji $f^{n+1}(x)$ za svaki $x \in [x_0, x_n]$ za neki $n \in \mathbb{N}_0$ i neka su $x_0 < \cdots < x_n$ čvorovi interpolacijskog polinoma $p_n(x)$ za funkciju $f(x)$. Tada za svaki $x \in [x_0, x_n]$ postoji $\psi \in [x_0, x_n]$ takav da vrijedi:*

$$e_n(x) = f(x) - p_n(x) = \frac{\omega(x)}{(n+1)!} f^{n+1}(\psi),$$

gdje je

$$\omega(x) = \prod_{i=0}^{n} (x - x_i).$$

Definicija 7. *Kažemo da je funkcija $g(x)$ reda $\mathcal{O}(h(x))$ za $x \to x_0$, odnosno*

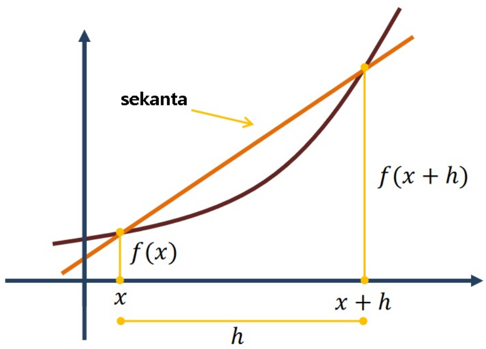
$$g(x) = \mathcal{O}(h(x)),\ x \to x_0$$

ako postoje konstante λ, C takve da

$$|x - x_0| \leq \lambda \implies |g(x)| \leq C|h(x)|.$$

2.2 Numeričko deriviranje realnih funkcija realnih varijabli

Derivacija funkcije nam daje informaciju o tome kako se vrijednost funkcije mijenja u odnosu na promjenu ulazne vrijednosti. Vrijednost derivacije funkcije u točki geometrijski se interpretira kao koeficijent smjera tangente na grafu funkcije u toj točki. Pokušajmo sada odrediti izraz koji opisuje taj koeficijent: započinjemo prvo s konstruiranjem sekante, tj. za neku funkciju f i za neku točku njene domene x napravimo da ta sekanta prolazi kroz čvorove $(x, f(x))$ i $(x+h, f(x+h))$ za neki $h > 0$, odnosno:

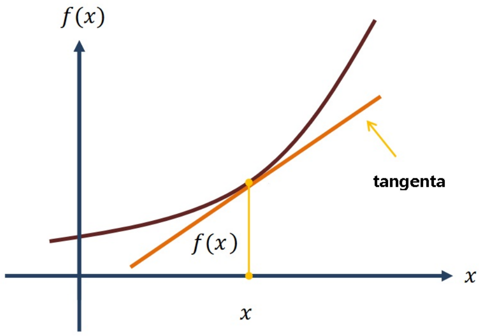


Slika 2: Sekanta. [3]

Koeficijent smjera te sekante je:

$$\frac{f(x+h) - f(x)}{h}, \tag{2}$$

a kako bi dobili koeficijent smjera tangente moramo smanjiti h (približiti $x + h$ i x proizvoljno blizu), odnosno odrediti limes izraza (2) kada h teži k nuli.



Slika 3: Tangenta. [3]

Intuitivno, izraz (2) mora biti aproksimacija egzaktne vrijednosti derivacije funkcije u točki. Numeričke metode za približno određivanje derivacije u točki su vrlo korisne jer otklanjaju potrebu za određivanjem limesa čiji postupak može biti vrlo zahtjevan.

Primjer 6. *Promotrimo funkciju $f(x) = x^2$. Njena (egzaktna) derivacija je:*

$$\lim_{h\to 0}\frac{f(x+h)-f(x)}{h} = \lim_{h\to 0}\frac{(x+h)^2-x^2}{h} = \lim_{h\to 0}\frac{x^2+2xh+h^2-x^2}{h} = \lim_{h\to 0}\frac{2xh+h^2}{h} = 2x.$$

Derivirajmo sada numerički za $h = 1$:

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

Za $h = 0.01$:

$$\frac{f(x+0.01)-f(x)}{0.01} = \frac{x^2+0.02x+0.0001-x^2}{0.01} = 2x+0.01.$$

Jasno je da uzevši manji h dobivena derivacija se približava onoj egzaktnoj, pa slutimo na osnovu ovog primjera da je ovakav pristup aproksimiranju vrijednosti derivacije u redu.

Pogledajmo sada nešto detaljniji pristup numeričkom deriviranju: pretpostavimo da imamo neku proizvoljnu realnu funkciju realne varijable f te neke točke $x_0, x_1, \ldots, x_n$, u kojima je ona diferencijabilna. Naime, uz pretpostavku da je f klase C^{n+1} na nekom segmentu $[a,b]$, funkciju f možemo zapisati kao

$$f(x) = p_n(x) + e_n(x),$$

gdje je $p_n(x)$ Newtonov interpolacijski polinom, a $e_n(x)$ greška tog interpolacijskog polinoma. Uz pretpostavku da je f klase C^{n+2} za $x = x_0$ u p_n' i e_n' dobivamo:

$$p_n'(x_0) = f[x_0,x_1] + (x_0-x_1)f[x_0,x_1,x_2] + \cdots + (x_0-x_1)\ldots(x_0-x_{n-1})f[x_0,x_1,\ldots,x_n],$$

$$e_n'(x_0) = \frac{f^{n+1}(\psi)}{(n+1)!}(x_0-x_1)\ldots(x_0-x_n).$$

Zbog svojstva derivacije, odnosno pravila o derivaciji zbroja dobivamo da je

$$f'(x_0) = p_n'(x_0) + e_n'(x_0),$$

tj. $p_n'(x_0)$ je aproksimacija derivacije funkcije f u točki x_0. Ako označimo s

$$H = \max_k |x_0 - x_k|,$$

onda je za $H \to 0$, greška $e'_n(x_0)$ reda veličine

$$e'_n(x_0) \leq \mathcal{O}(H^n).$$

Dakle, aproksimacijska formula za derivaciju može biti proizvoljno visokog reda n, no valja napomenuti da tada zbog iznimne složenosti njihova uporaba u praksi nije praktična pa se stoga uvijek fokusiramo na polinome nižeg stupnja ili pribjegavamo drugim metodama za aproksimaciju funkcije. [2]

Metoda asimetrične razlike

Interesantna poveznica podijeljenih razlika i definicije derivacije nastaje za $n = 1$, $h = x_1 - x_0$, gdje imamo:

$$p'_1(x_0) = f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_0 + h) - f(x_0)}{h}.$$

Dobivena je zapravo prva aprkosimacija koju smo koristili, što nam opet podržava slutnju da bi ona mogla biti korektna. Greška te metode (uz pretpostavku da je f klase C^3) je $e'_1(x_0) = \frac{f^2(\psi)}{2}h$, ona je reda $\mathcal{O}(h)$ za $h \to 0$, tj. linearna je u ovisnosti o h. U praksi je takva konvergencija i dalje nedovoljno precizna. Poželjno je razvijanje metoda čija je greska barem kvadratnog reda. [2]

Metoda centralne razlike

Neka je $n = 2$ i neka imamo ekvidistantne čvorove. Naziv "centralna" potječe iz činjenice da je točka (u kojoj pronalazimo vrijednost derivacije funkcije) x_0 polovište dužine određene točkama x_1 i x_2. Sada za tako odabrane x_0, x_1, x_2 postoji $h > 0$ t.d. $x_1 = x_0 + h$, $x_2 = x_0 - h$ i vrijednost:

$$p'_2(x_0) = f[x_0, x_1] + (x_0 - x_1) f[x_0, x_1, x_2].$$

Nadalje, iz sljedeće tablice iščitavamo podijeljene razlike:

x_2	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$
x_2	$f(x_2)$		
		$\frac{f(x_0)-f(x_2)}{h}$	
x_0	$f(x_0)$		$\frac{f(x_1)-2f(x_0)+f(x_2)}{2h^2}$
		$\frac{f(x_1)-f(x_0)}{h}$	
x_1	$f(x_1)$		

Tablica 4: Podijeljene razlike. [2]

Dobivamo :

$$p_2'(x_0) = \frac{f(x_1) - f(x_0)}{h} - h\frac{f(x_1) - 2f(x_0) + f(x_2)}{2h^2} = \frac{f(x_1) - f(x_2)}{2h}.$$

Greška ove aproksimacije je kvadratnog reda, kao takva je znatno bolja od prije navedene metode:

$$e_2'(x_0) = \frac{f^3(\psi)}{6}(x_0 - x_1)(x_0 - x_2) = -h^2\frac{f^3(\psi)}{6}. \quad [2]$$

Greška zaokruživanja kod numeričkog deriviranja

Primjer 7. *Metodom centralne razlike aproksimirano derivaciju funkcije* $f(x) = \sin x$ *u točki* $a = 0.5$*. Uzimavši različite* $h > 0$ *dolazimo do sljedeće tablice:*

h	$\frac{f(a+h)-f(a-h)}{2h}$	apsolutna greška
10^{-7}	0.8775825372	$2.5 \cdot 10^{-8}$
10^{-9}	0.87755825622	$2.9 \cdot 10^{-10}$
10^{-11}	0.8775813409	$1.2 \cdot 10^{-6}$
10^{-16}	1.110223025	$2.3 \cdot 10^{-1}$
10^{-17}	0.000000000	$8.8 \cdot 10^{-1}$

Tablica 5: Greška zaokruživanja. [4]

Zbog katastrofalnog kraćenja apsolutna greška nakon odabira dovoljno male vrijednosti h počinje rasti. Cilj nam je odrediti vrijednost h za koju je greška minimalna. Vrijedi:

$$f'(x_0) = \frac{f(x_1) - f(x_2)}{2h} + e_2'(x_0), \quad e_2'(x_0) = -h^2\frac{f^3(\psi)}{6}.$$

Pretpostavimo da $\overline{f(x_1)} = f(x_1) + \epsilon_1$, $\overline{f(x_2)} = f(x_2) + \epsilon_2$, $|\epsilon_1|, |\epsilon_2| \leq \epsilon$, odnosno da se provodi zaokruživanje dobivenih vrijednosti. Uvrstimo li u gornji izraz, dobivamo:

$$f'(x_0) = \frac{\overline{f(x_1)} - \overline{f(x_2)}}{2h} - \frac{\epsilon_1 - \epsilon_2}{2h} + e_2'(x_0).$$

Sada uz pretpostavku da je h prikaziv na računalu, te da je greška pri računanju kvocijenta u podijeljenoj razlici zanemariva, uvodimo oznaku err_2 za ukupnu grešku:

$$err_2 = f'(x_0) - \frac{\overline{f(x_1)} - \overline{f(x_2)}}{2h} = -\frac{\epsilon_1 - \epsilon_2}{2h} + e_2'(x_0)$$

$$\implies |err_2| \leq \frac{|\epsilon_1| + |\epsilon_2|}{2h} + |e_2'(x_0)|$$

$$\implies |err_2| \leq \frac{\epsilon}{h} + \frac{M_3}{6}h^2, \quad M_3 = \max_{x\in[x_2,x_1]} |f^3(x)|.$$

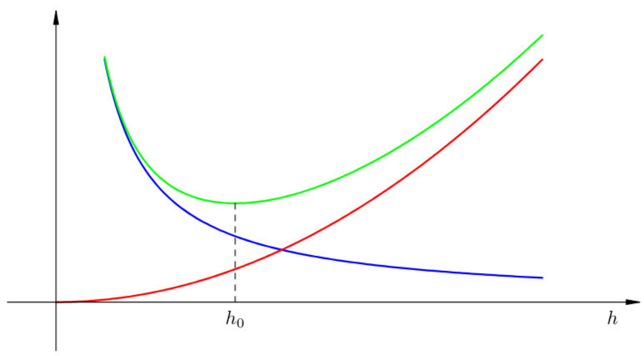
Minimum te greške možemo pronaći tako da gledamo dobiveni izraz kao funkciju po h, tj. $e(h) = \frac{\epsilon}{h} + \frac{M_3}{6}h^2$, te njenu derivaciju izjednačimo s nulom. Dobiveni h_0 će zbilja biti onaj u kojem ta funkcija postiže minimum iz razloga jer $e''(h) > 0$ za $h > 0$. Dakle,

$$e'(h) = -\frac{\epsilon}{h^2} + \frac{M_3}{3}h = 0,$$

$$\implies h_0 = \left(\frac{2\epsilon}{M_3}\right)^{1/3}.$$

Finalno, uvrštavanjem dobivamo iznos za najmanju vrijednost greške:

$$e(h_0) = \frac{3}{2}\left(\frac{M_3}{3}\right)^{1/3}\epsilon^{2/3}. \text{ [2]}$$



Slika 4: Graf funkcije greške $e(h)$ (zeleno), greške zaokruživanja (crveno) i funkcije greške bez greške zaokruživanja $\frac{M_3h^2}{6}$ (plavo). [2]

Primjer 8. *Zadani su čvorovi koji odgovaraju jednadžbi funkcije $y = x^3$: $(x_0, y_0) = (1, 1)$, $(x_1, y_1) = (2, 8)$, $(x_2, y_2) = (3, 27)$, $(x_3, y_3) = (4, 64)$, $(x_4, y_4) = (5, 125)$. Zadatak je aproksimirati vrijednosti derivacije u tim čvorovima uporabom metode numeričkog deriviranja. Koristimo formulu za numeričko deriviranje drugog reda, odnosno $p_2'(x_i) = f[x_i, x_{i+1}] + (x_i - x_{i+1})f[x_i, x_{i+1}, x_{i+2}]$. Riječ je o ekvidistantnim čvorovima pa računamo konačne razlike i primjenjujemo Lemu 1:*

$$\begin{array}{llll}
x_0 = 1 & f(x_0) = 1 & & \\
 & & \Delta f(x_0) = 7 & \\
x_1 = 2 & f(x_1) = 8 & & \Delta^2 f(x_0) = 12 \\
 & & \Delta f(x_1) = 19 & \\
x_2 = 3 & f(x_2) = 27 & & \Delta^2 f(x_1) = 18 \\
 & & \Delta f(x_2) = 37 & \\
x_3 = 4 & f(x_3) = 64 & & \Delta^2 f(x_2) = 24 \\
 & & \Delta f(x_3) = 61 & \\
x_4 = 5 & f(x_4) = 125 & &
\end{array}$$

Tablica 6: Konačne razlike [1]

$$f[x_0, x_1] = \frac{\Delta^1 f(x_0)}{1!1^1} = 7, \quad f[x_1, x_2] = \frac{\Delta^1 f(x_1)}{1!1^1} = 19, \quad f[x_2, x_3] = \frac{\Delta^1 f(x_2)}{1!1^1} = 37,$$
$$f[x_3, x_4] = \frac{\Delta^1 f(x_3)}{1!1^1} = 61, \quad f[x_0, x_1, x_2] = \frac{\Delta^2 f(x_0)}{2!1^2} = 6, \quad f[x_1, x_2, x_3] = \frac{\Delta^2 f(x_1)}{2!1^2} = 9,$$
$$f[x_2, x_3, x_4] = \frac{\Delta^2 f(x_2)}{2!1^2} = 12.$$

Uvrstimo sve određene vrijednosti u formulu:

$$p_2'(x_0) = f[x_0, x_1] + (x_0 - x_1)f[x_0, x_1, x_2] = 7 + (1 - 2)6 = 1,$$

$$p_2'(x_1) = f[x_1, x_2] + (x_1 - x_2)f[x_1, x_2, x_3] = 19 + (2 - 3)9 = 10,$$

$$p_2'(x_2) = f[x_2, x_3] + (x_2 - x_3)f[x_2, x_3, x_4] = 37 + (3 - 4)12 = 25,$$

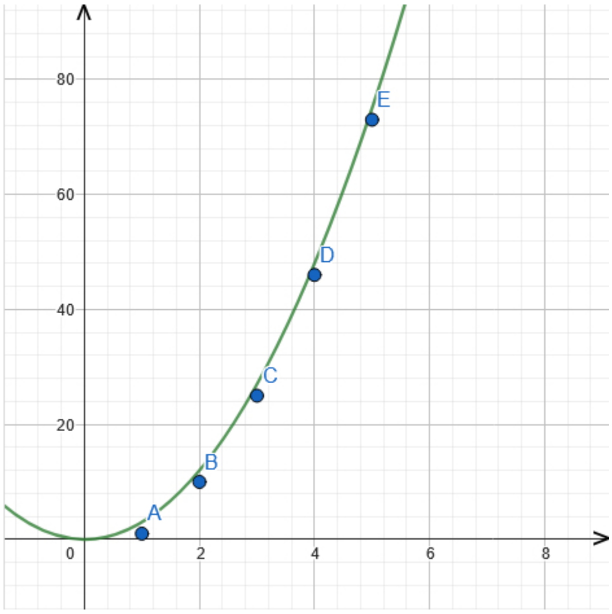
Primijetimo da kod računanja vrijednosti $p_2'(x_3)$ i $p_2'(x_4)$ nemamo mogućnost primijene formule $p_2'(x_i) = f[x_i, x_{i+1}] + (x_i - x_{i+1})f[x_i, x_{i+1}, x_{i+2}]$ jer ta formula zahtjeva poznavanje vrijednosti x_5 i x_6. Stoga za zadnja dva čvora primjenjujemo formulu $p_2'(x_i) = f[x_i, x_{i-1}] + (x_i - x_{i-1})f[x_i, x_{i-1}, x_{i-2}]$, odnosno koristimo obrnuti redoslijed čvorova. Primijetimo:

$$f[x_3, x_2] = \frac{f(x_2) - f(x_3)}{x_2 - x_3} = \frac{(-1)(f(x_3) - f(x_2))}{(-1)(x_3 - x_2)} = f[x_2, x_3] = 37.$$

Analogno, vrijedi: $f[x_4, x_3] = f[x_3, x_4]$, $f[x_4, x_3, x_2] = f[x_2, x_3, x_4]$, $f[x_3, x_2, x_1] = f[x_1, x_2, x_3]$. *Odnosno imamo:*

$$p_2'(x_3) = f[x_3, x_2] + (x_3 - x_2)f[x_3, x_2, x_1] = 37 + (4-3)9 = 46,$$

$$p_2'(x_4) = f[x_4, x_3] + (x_4 - x_3)f[x_4, x_3, x_2] = 61 + (5-4)12 = 73.$$



Slika 5: Točke $A(1,1)$, $B(2,10)$, $C(3,25)$, $D(4,46)$, $E(5,73)$ i graf funkcije $y = 3x^2$.

Dobili smo čvorove $(1,1)$, $(2,10)$, $(3,25)$, $(4,37)$ *i* $(5,46)$ *koji odgovaraju jednadžbi funkcije* $y_d = 3x^2 - 2$. *Iz funkcije greške za numeričko deriviranje drugog reda dobivamo:*

$$e_2'(x_0) = (x_0 - x_1)(x_0 - x_2)\frac{y^3(\psi)}{6} = (-1)(-2)\frac{6}{6} = 2.$$

Sličnim postupkom dobivamo potpuno iste vrijednosti greške i za ostale čvorove, odnosno $e_2'(x_i) = 2$, $i = 0, 1, 2, 3, 4$. *Vrijedi:*

$$y'(x_i) = 3x_i^2 = 3x_i^2 - 2 + 2 = (3x_i^2 - 2) + (2) = y_d(x_i) + e'(x_i), \quad i = 0, 1, 2, 3, 4.$$

3 Implementacija u Pythonu

Python je jedan od najpoznatijih programskih jezika današnjice. Zbog svoje jednostavnosti i pristupačnsosti privukao je veliku i aktivnu zajednicu programera, koja je razvojem velikog broja paketa doprinijela njegovoj širokoj primjeni. Primjenu Pythona moguće je pronaći u raznim industrijskim i znanstvenim granama. U ovom dijelu rada uvodimo primjere programskih kodova realiziranih u Pythonu, a odnose se na numeričko deriviranje.

Implementacija simetričnih i centralnih razlika

Prije svega uvozimo dva dobro poznata paketa:

```python
import numpy as np
import matplotlib.pyplot as plt
```

U nastavku programa definiramo Python funkciju čije su ulazne vrijednosti dvije liste: lista X u kojoj se nalaze ekvidistantne x-koordinate čvorova, te lista Y čiji su elementi pripadajuće y-koordinate tih čvorova. Funkcija vraća listu u kojoj su pohranjene pripadajuće aproksimacije vrijednosti derivacije:

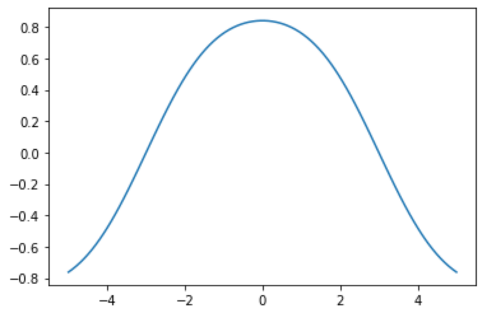
```python
def deriv(X,Y):
    h=X[1]-X[0]
    L=[]
    for i in range(len(X)):
        if i==0:
            L.append((Y[1]-Y[0])/h)
        elif i<len(X)-1:
            L.append((Y[i+1]-Y[i-1])/(2*h))
        else:
            L.append((Y[len(X)-1]-Y[len(X)-2])/h)
    return L
```

U ovom programu definirali smo konstantnu razliku (korak) h između x-koordinata zadanih čvorova i praznu listu L. U listu L pohranjujemo vrijednosti pomoću for petlje koja prolazi svim indeksima od 0 do indeksa jednakog duljini liste X. Prva i zadnja pohranjena vrijednost određena je pomoću formule izvedene iz metode asimetrične razlike, dok za ostale koristimo formulu metode centralne razlike.

Uzmimo 100 točaka u intervalu $[-5, 5]$ i definirajmo listu $Y = \sin(\cos(\frac{x\pi}{6}))$. Naredbom *plt.plot*$(X, Y)$ dobivamo grafički prikaz interpolacijske funkcije koja prolazi kroz čvorove čije su koordinate određene listama X i Y. Označimo tu interpolacijsku funkciju s f_Y. Naredbom *plt.plot*$(X, deriv(X, Y))$ dobivamo grafički prikaz interpolacijske funkcije koja prolazi čvorovima s koordinatama zadanima listom X i listom dobivenom primjenom funkcije *deriv*() na liste X i Y. Dobiveni graf je graf aproksimacije derivacije funkcije f_Y.

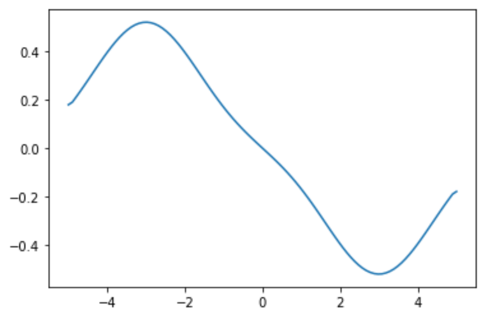
```
X=np.linspace(-5,5,100)
Y=np.sin(np.cos(X*np.pi/6))

plt.figure(1)
plt.plot(X,Y)

plt.figure(2)
plt.plot(X,deriv(X,Y))
```

Graf funkcije f_Y je:



Slika 6: Graf funkcije f_Y.

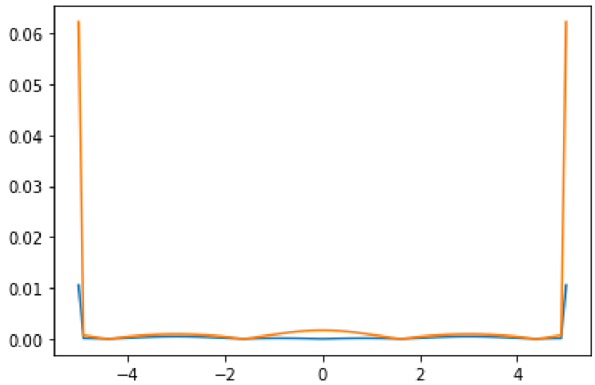
Graf aproksimacije derivacije funkcije f_Y je:



Slika 7: Graf aproksimacije derivacije funkcije Y.

```
Yder=-np.pi*np.sin(X*np.pi/6)*np.cos(np.cos(np.pi*X/6))/6
plt.figure(3)
plt.plot(X,abs(deriv(X,Y)-Yder))
plt.plot(X,abs(deriv(X,Y)-Yder)/abs(Yder))
```

Definirali smo listu egzaktnih vrijednosti derivacije funkcije f_Y s oznakom $Yder$, zatim smo grafički prikazali funkciju relativne i apsolutne greške aproksimacije njene derivacije, uzimajući za egzaktne vrijednosti elemente liste $Yder$.



Slika 8: Funkcije apsolutne (narančasto) i relativne (plavo) greške aproksimacije.

Odredimo sada maksimalnu relativnu i apsolutnu grešku:

```
L=deriv(X,Y)
APS=[]
REL=[]

for i in range(len(L)):
    APS.append(abs(L[i]-Yder[i]))
    REL.append(abs(L[i]-Yder[i])/abs(Yder[i]))

print(max(APS))
print(max(REL))
```

U listu APS pohranjujemo sve vrijednosti apsolutnih grešaka, nakon čega prikazujemo maksimalnu vrijednost te liste. Dobivena vrijednost iznosi 0.010546313078798303, što odgovara Slici 7. Na sličan način, u listu REL pohranjujemo sve vrijednosti relativnih grešaka i dobivamo da je maksimalna vrijednost te liste 0.06218008416383216.

Slučaj sa neekvidistantnim čvorovima

Definiramo nešto zahtjevniju funkciju koja ne ovisi o udaljenosti pojedinih čvorova. Prvo definiramo pomoćnu funkciju koja određuje vrijednosti podijeljenih razlika:

```
def newton(x,y,a,b):
    if a==b:
        return y[a]
    else:
        return (newton(x,y,a+1,b)-newton(x,y,a,b-1))/(x[b]-x[a])
```

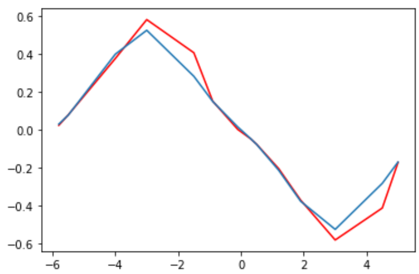
Ulazne vrijednosti funkcije *newton*() su liste x i y u kojima su pohranjene vrijednosti koordinata zadanih čvorova, te vrijednosti a i b, koje označavaju početni i završni indeks podijeljene razlike. Na primjer, za podijeljenu razliku $f[x_3, x_4, x_5]$, vrijedi $a = 3$ i $b = 5$. *newton*() je rekurzivna funkcija koja primjenjuje formulu iz definicije podijeljenih razlika i pretpostavlja da su indeksi čvorova u rastućem poretku, na primjer, $f[x_2.x_3, x_4]$. Sada definiramo funkciju koja aproksimira vrijednosti derivacija:

```
def deriv2(X,Y):
    L=[]
    for i in range(len(X)):
        if i<2:
            L.append(newton(X,Y,i,i+1)+(X[i]-X[i+1])*newton(X,Y,i,i+2))
        else:
            L.append(newton(X,Y,i-1,i)+(X[i]-X[i-1])*newton(X,Y,i-2,i))
    return L
```

Za razliku od funkcije *deriv*(), sada nemamo konstantan korak h; umjesto toga, samo definiramo praznu listu L. *For* petljom nadopunjujemo listu L, pri čemu za svaki čvor koristimo formulu za numeričko deriviranje drugog reda. Slično kao i u Primjeru 8, za prvih nekoliko čvorova koristimo formulu $p_2^{'}(x_i) = f[x_i, x_{i+1}] + (x_i - x_{i+1})f[x_i, x_{i+1}, x_{i+2}]$, dok nakon određenog čvora (u slučaju funkcije *deriv*2(), to je drugi čvor), koristimo formulu $p_2^{'}(x_i) = f[x_i, x_{i-1}] + (x_i - x_{i-1})f[x_i, x_{i-1}, x_{i-2}]$.

```
x=np.array([-5.8,-5.5,-4,-3,-1.5,-0.9,-0.1,0.3,0.5,1.2,1.9,3,4.5,5])
y=np.sin(np.cos(x*np.pi/6))
z=x**3

Yder=-np.pi*np.sin(x*np.pi/6)*np.cos(np.cos(np.pi*x/6))/6

plt.figure(1)
plt.plot(x,deriv2(x,y), color='red')
plt.plot(x,Yder)

plt.figure(2)
plt.plot(x,deriv2(x,z), color='red')
plt.plot(x,3*x**2)
```

Zadali smo čvorove koji nisu ekvidistantni i dvije liste $z = x^3$ i $y = \sin(\cos(\frac{x\pi}{6}))$. Definirali smo egzaktne derivacije interpolacijskih funkcija koje prolaze kroz čvorove zadane tim listama te smo ih grafički usporedili s aproksimacijama dobivenima pomoću funkcije $deriv2()$.



Slika 9: Graf derivacija prve interpolacijske funkcije (egzaktna je označena plavom bojom).

Slika 10: Graf derivacija druge interpolacijske funkcije (egzaktna je označena plavom bojom).

4 Zaključak

Deriviranje funkcija često je prisutno u raznim znantsvenim i inžinjerskim disciplinama. Često je potrebno brzo i efikasno odrediti vrijednost derivacije funkcije u točki gdje egzaktna vrijednost nije potrebna. Stoga je u ovom radu predstavljen pristup koji zadovoljava te zahtjeve. Korištenjem računalnih sustava olakšan je postupak određivanja derivacija funkcija i izbjegnuta pojava računskih grešaka izazvanih ljudskom nepažnjom. Osim toga, u radu je opisana efikasnost aproksimiranih vrijednosti u odnosu na egzaktne, pri čemu se ističe da ona ovisi o primjenjenoj numeričkoj metodi, rasporedu točaka ili odabiru funkcije na kojoj se provodi postupak kao i grešci zaokruživanja uzrokovanoj ograničenim resursima računalnih sustava. Cilj je uvijek odabrati metodu koja je dovoljno precizna, ali istovremeno jednostavna za primjenu. Rad detaljnije razmatra numeričke metode deriviranja kroz primjere zadatka, analizu greške i primjere programskog koda realiziranog u Pythonu.

Popis slika

Popis tablica

Bibliografija

[1] Bujačić Babić, S., Numerička matematika, materijali s predavanja - skripta

[2] Drmač Z., Marušić M., Singer S., Hari V., Rogina M., Singer S., Numerička analiza, skripta, Prirodoslovni fakultet - Matematički odjel, Sveučilište u Zagrebu.

[3] Halvorsen H.-P., - Numerical Differentiation in Python. URL: `https://www.halvorsen.blog/documents/programming/python/resources/powerpoints/Numerical%20Differentiation%20in%20Python.pdf`
(pristupio sadržaju 10. svibnja 2024.)

[4] Mørken, K., Numerical Algorithms and Digital Representation - Department of Informatics, University of Oslo. URL: `https://www.uio.no/studier/emner/matnat/math/MAT-INF1100/h10/kompendiet/komp.pdf`
(pristupio sadržaju 25. travnja 2024.)