Numerička integracija

Pletikos, Anton

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

NUMERIČKA INTEGRACIJA

Rijeka, srpanj 2022.

Anton Pletikos
0069080477

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

NUMERIČKA INTEGRAIJA

Mentor: doc. dr. sc. Ivan Dražić

Komentor: prof. dr. sc. Viktor Sučić

Rijeka, srpanj 2022.

Anton Pletikos
0069080477

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 5. ožujka 2021.

Zavod: **Zavod za matematiku fiziku, strane jezike i kineziologiju**
Predmet: **Inženjerska matematika ET**
Grana: **1.01.06 numerička matematika**

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Anton Pletikos (0069080477)**
Studij: Preddiplomski sveučilišni studij elektrotehnike

Zadatak: **Numerička integracija // Numerical integration**

Opis zadatka:

U radu je potrebno objasniti razliku između analitičnog i numeričkog rješavanja matematičkih problema, s posebnim naglaskom na rješavanje određenih integrala. Zatim je potrebno detaljno opisati trapeznu i Simpsonovu formulu za numeričko integriranje te objasniti ideju Gaussovih kvadraturnih formula. Potrebno je realizirati pripadne algoritme u nekom od programskih jezika te praktično usporediti svojstva navedenih metoda s obzirom na točnost, broj koraka, brzinu izvršavanja algoritma te robusnost. Opisane metode potrebno je staviti u kontekst primjene u elektrotehnici.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Anton Pletikos

Zadatak uručen pristupniku: 15. ožujka 2021.

Mentor:

Doc. dr. sc. Ivan Dražić

Prof. dr. sc. Viktor Sučić (komentor)

Predsjednik povjerenstva za završni ispit:

Prof. dr. sc. Viktor Sučić

IZJAVA

Sukladno članku 8. Pravilnika o završnom radu, završnom ispitu i završetku preddiplomskih sveučilišnih studija/stručnih studija Tehničkog fakulteta Sveučilišta u Rijeci od 1. veljače 2020., izjavljujem da sam samostalno izradio završni rad prema zadatku preuzetom dana 15. ožujka 2021.

Rijeka, 07. srpnja 2022.

Anton Pletikos

Anton Pletikos

Ovaj diplomski rad posvećujem svojim roditeljima i sestri. Hvala vam na svemu što ste mi pružili tijekom mog studija. Posebno se zahvaljujem mentoru doc. dr. sc. Ivan Dražiću za pomoć pri izradi ovog rada svojim savjetima, preporukama te na iskazanoj potpori i strpljenju.

Sadržaj

1. Uvod

Numerička matematika je grana matematike koja proučava matematičke probleme koji se javljaju u praksi, na poljima kao što su tehnika i znanost. Prilikom matematičkog opisa koristit ćemo se fizikalnom slikom. Problem nastaje kada promatramo stvarnu pojavu i želimo je matematički opisati. Fizikalna slika značajno je jednostavnija od stvarne pojave jer se pojedini faktori zanemaruju. Nelinearni se problemi lineariziraju, a stvarne fizikalne vrijednosti moramo aproksimirati. Uz sve navedene probleme standardni algebarski pristup može biti značajno teži zbog čestog korištenja računala kao alata u rješavanju problema. Kako bi se zaobišle komplikacije, koristi se numerički pristup.

Numerički pristup nikada ne daje točno rješenje već isključivo aproksimacijsko rješenje. Dobivena su rješenja aproksimacije koje imaju pogrešku zbog čega je dobro poznavati vezu između korištene numeričke metode i pogreške.

Teoremi numeričke matematike nisu dovoljni za rješavanje problema te je potrebno s razumijevanjem sagledati zadani problem. Ako ne uvažimo taj pristup, odabrane metode neće biti dovoljno precizne niti prihvatljive. Za uspješno rješavanje problema potrebno je poznavati određena svojstva problema. Mora se promotriti postoje li jedinstvena rješenja te kako se rješenja ponašaju ako se početni podaci izmjene . Odabir metode ovisit će o konvergenciji kao i o brzini konvergencije.

Konvergencija u matematici opisuje granično ponašanje skupa beskonačnih brojeva prilikom približavanja beskonačnosti. Možemo reći da je to broj kojem se približavaju elementi nekog skupa skup kada iteracijska varijabla skupa teži ka beskonačnosti. Uz konvergenciju bitna je i složenost kao i točnost odnosno broj računskih operacija koje su potrebne za račun, kao i podatak o broju značajnih znamenaka izračunanog rješenja.

Ubrzan razvoj numeričke matematike počeo je s početkom dvadesetog stoljeća. Razvoj numeričke matematike usko je vezan uz razvoj računala, operacijskih sustava i programskih jezika.
U njihovoj sinergiji razvijeni su standardni programski alati za pojedina područja koji omogućuju rutinsko modeliranje čak i zahtjevnijih problema. Tako su nastale grane numeričke matematike – numerička analiza i numerička linearna algebra.

Numerička analiza bavi se pronalaženjem i unaprijeđenjem postupaka za numeričko izračunavanje i ostalih grana vezanih uz matematičku analizu kao što su numeričko deriviranje, numeričko rješavanje diferencijalnih jednadžbi i pogotovo za ovaj rad bitno numeričko integriranje. Uz to bitno je naglasiti da se numerička analiza bavi i ocjenjivanjem pogrešaka metoda, pogrešaka samih metoda i pogreška prouzrokovanih arhitekturom računala.

Začetke numeričke matematike pronalazimo su još u starom Egiptu gdje se koristio numerički algoritam za pronalazak korijena brojeva. Navedena metoda kao i još brojne druge metode pronađene su u Rhind papirusu napravljenom 1650. godine prije nove ere. Eudoks iz Knida[1] i Arhimed[2] usavršili su metodu iscrpljenja. Metoda iscrpljenja je tehnika koju su razvili klasični Grci za dokazivanje veze između površine i volumena. Metoda se smatra prethodnicom modernoga integriranja. Zasnovana je na aksiomu koji kaže da se određena veličina može napraviti manjom veličinom ako prvu veličinu prepolovimo. Metoda se zasniva na prepolavljanju početne zadane veličine. Prepolavljanjem dobivamo broj, odnosno omjer, koji nam opisuje koliko je najmanja dobivena veličina manja od početne. Time ovaj aksiom može dokazati da je površina kruga proporcionalna kvadratu njegovog radijusa. Naziv „Metoda iscrpljenja" dobiva naziv tek u renesansnoj Europi. Naziv Metoda iscrpljenja odnosi se na primjenu rigoroznih postupaka koji su iscrpljivali područje figura uzastopnim poligonalnim aproksimacijama.

Na temeljima dotad razvijenih metoda Isaac Newton[3] i Gottfried Leibniz[4] razvili su matematičke metode za rješavanje fizikalnih modela kojima se nije moglo pronaći eksplicitno rješenje. Zbog toga jednostavnije je bilo precizno aproksimirati rješenje čime je došlo do veće potrebe za korištenjem numeričkih metoda. Newton je posebno bitan jer je razvio brojne numeričke metode za rješavanje niza raznih problema. Uz njegovo se ime vežu mnoge matematičke metode nastale kao izvedenice njegovih originalnih ideja. Newtonovi radovi, kao što su rješenja za generalne funkcije za opisivanje polinoma i teorije gravitacije bili su mogući zbog primjene numeričkih metoda.

Uz Newtonova dostignuća važno je spomenuti Johna Napiera[5]. John Napier javno je objavio metodu logaritama 1614. godine. Razvio je tablicu s pomoću koje je sada moguće jednostavno pronaći vrijednosti logaritamskih funkcija. Teoretičari koji su doprinijeli razvoju numeričke analize u 18. i 19. stoljeću su Švicarac Leonhard Euler[6], Talijan Joseph-Louis Lagrange[7] i Nijemac

[1]Grčki matematičar i astronom Eudoks iz Knida dokazao je da je volumen piramide jednak jednoj trećini volumena prizme, te volumen stošca jednake osnovice. Smatra se da je izumitelj pješčanog sata, ali i da je Zemlja sfernog oblika. Koristio je kalendar s 365,25 dana u godini.

[2]Arhimed je grčki matematičar, astronom i fizičar. Jedno od Arhimedovih većih postignuća je aproksimiramacija broja π. Aproksimirao je π tako što je opisivao i upisivao pravilne višekutnike u krug. Pretpostavio je da se rezultat nalazi u području od 3 i $1/7$ do 3 i $10/71$. Bavio se određivanjem težišta tijela i pronalaženjem njihova obujma. Također bavio se hidrodinamikom. Izumio Arhimedov vijak i Arhimedov koloturnik.

[3]Jedan od najpoznatijih engleskih fizičara, matematičara i astronoma je Isaac Newton. Najveća postignuća ostvario je na poljima matematike, statike i dinamike tijela. Osim matematike i fizike bavio se i kemijom. Prvi Newtonovi radovi bili su izvod binomnoga poučaka za cjelobrojne potencijale te kasnije za sve racionalne brojeve. Bavio se diferencijalnim i integralnim računom. Doprinio je razvitku empirijske interpolacijske formule i metode za približna rješenja algebarske jednadžbe bilo kojega stupnja.

[4]Prvi predsjednik Berlinske akademije znanosti Gottfried Wilhelm Leibniz jedan je od najpoznatijih njemački filozofa, matematičara i fizičara. Leibnizova najveća matematička dostignuća su diferencijalni, integralni račun i smislio infinitezimalni račun.

[5]John Napier škotski teolog, matematičar, fizičar, astronom i astrolog. Napisao je jedno od najbitnijih djela u matematici "Opis divnog kanona logaritama". Opisao je kako ručno rješavati logaritme i sastavio je logaritamske tablice. Djelo doprinosi ubrzanju razvoja matematike, fizike i astronomije.

[6]Leonhard Euler švicarski matematičar, fizičar i astronom. Objavio je oko 900 radova. Odbor Švicarske akademije znanosti je 50 godina objavljivao njegove radove. Razvio je teoriju redova, uveo je Eulerove integrale. Na području diferencijalne geometrije dao je prvu formulu zakrivljenosti ploha (Eulerov poučak).

[7]Talijan Joseph-Louis Lagrange bio je matematičar i astronom. Svoj je doprinos dao na poljima matematičke analize, teorije brojeva kao i klasične mehanike.

Carl Friedrich Gauss[8].

Kasnije su mnogi matematičari i fizičari pokušavali dobiti matematičke modele za fizikalne zakone. Tako su i nastali modeli za krute tvari, mehaniku fluida, kao i modeli magnetizma i elektriciteta. U 20. stoljeću metode pronalaze mjesto i u relativističkoj i kvantnoj mehanici. Jedna od najraširenijih i najkorištenijih metoda numeričke analize uključuje približavanje kontinuirane površine ili strukture konačnim brojem jednostavnih elemenata. Metoda je poznata pod nazivom metoda konačnih elemenata. Tehniku je razvio Harold Martin 1950. godine. Metoda je razvijena u tvrtki Boeing radi analize naprezanja krila tada novih mlaznih zrakoplova, no metoda je od tada pronašla široku primjenu u mnogim poljima.

Ovaj uvodni pregled numeričke matematike napravljen je prema izvoru [1].

U ovom radu bavimo se posebnom vrstom numeričkih metoda koje se koriste kod približnog rješavanja integrala, s posebnim naglaskom na Trapeznu formulu te Simpsonovu formulu. Također će biti pokazano kako se te metode koriste unutar programskog jezika Python, a biti će prikazana i njihova primjena u elektrotehnici.

[8]Jedan od najvećih matematičara svih vremena bio je njemački matematičar i astronom Carl Friedrich Gauss. Gaussove zasluge za polje matematike: Gaussova raspodjela, Gaussova krivulja i Gaussov algoritam. Također, u fizici: Gaussovi zakon električnoga polja, Gaussov zakon magnetskoga polja i Gaussov sustav jedinica.

2. Pogreške u numeričkoj matematici

Bit numeričke matematike je određivanje približnih rješenja pri čemu se po samom smislu takvog tipa rješenja pojavljuje određena pogreška, a za dobru matematičku analizu metode bitno je poznavati sve moguće izvore pogrešaka. Pogreške ne nastaju samo zbog primjene numeričke metode, već i zbog idealizacije stvarne slike prilikom računanja, a cilj je približiti se stvarnim veličinama. Dobiveni rezultati uvijek će biti aproksimacije realnih vrijednosti, a važno je znati koliko moguće pogreške utječu na rezultat. Ovo poglavlje obrađeno je prema izvorima [2, 3, 4].

2.1. Vrste pogrešaka

Primjenom numeričke analize u znanosti izdvojene su sljedeće vrste pogrešaka: pogreške zaokruživanja, pogreške nastale zbog nepreciznosti ulaznih podataka, pogreške metode, modela i strojne pogreške.

Pogreške zaokruživanja nastaju kada zaokružimo broj na prvih par značajnih znamenki. To se događa zbog jednostavnosti računanja jer nije uvijek potrebno raditi s punim brojem i time svjesno radimo grešku. Ponekad nije moguće uzeti stvarni broj, kao što su π ili e.

Pogreške nastale zbog nepreciznosti ulaznih podataka često proizlaze iz vrijednosti konstanti. Vrlo često se za akceleraciju sile teže uzima $9.81\,\mathrm{m\,s^{-2}}$, iako je ta konstanta promjenjiva u ovisnosti o položaju na Zemlji.

Primjer 2.1. *Ako je tijelo mase $100kg$ na visini od $1m$ i pod utjecajem akceleracijske sile teže, imat će potencijalnu energiju.*

Formula za potencijalnu energiju glasi:

$$E_p = m \cdot h \cdot g^2, \tag{2.1}$$

gdje je E_p potencijalna energija dok m je masa, h visina i g konstanta za akceleraciju sile teže. Najčešće uzimamo za akceleraciju sile teže $9.81\ ms^{-2}$, no ako želimo koristiti standardizirani broj onda koristimo $9.80665\ ms^{-2}$. Dobivamo dva različita rješenja prvo za $9.81\ ms^{-2}$ koje glasi:

$$E_{p1} = 100 \cdot 1 \cdot (9.81)^2 = 9623\ J. \tag{2.2}$$

Za $9.80665\ ms^{-2}$ dobivamo:

$$E_{p1} = 100 \cdot 1 \cdot (9.80665)^2 = 9617.038 J. \tag{2.3}$$

Pogreške metode nastanu kada beskonačan iterativan proces zaustavimo već nakon nekoliko koraka. Dobar primjer za to je traženje lokalnih ekstrema funkcije. Iterativni proces označuje proces koji se ponavlja tako da od rezultata neke funkcije pravimo početni uvjet te iste funkcije te postupak ponavljamo zadani broj puta. Pogreška modela nastaje zbog složenosti stvarnog dijela i našeg pojednostavljivanja kako bi ga se moglo analizirati kao idealan model. Strojna pogreška nastaje prilikom zaokruživanja rezultata na fiksan broj decimala.Taj broj se može povećati ovisno o zapisu kao što je zapis s dvostrukom preciznošću koji će imati dvostruko veći broj decimala.

2.2. Apsolutna i relativna pogreška

Pogreška aproksimacije je razlika između stvarne i njezine aproksimirane vrijednosti. Ako stvarnu vrijednost nazovemo a, aproksimiranu vrijednost a^*, dobivamo izraz za izračun pogreške koji glasi:

$$a - a^*. \tag{2.4}$$

gdje je a stvarna vrijednost, a a^* njezina aproksimacija.

Koristeći se izrazom za pogrešku aproksimacije, možemo dobiti izraz za apsolutnu pogrešku aproksimacije Δ^*. Ona će biti jednaka apsolutnoj vrijednosti pogreške aproksimacije, a izražava se na sljedeći način:

$$\Delta^* = |a - a^*|. \tag{2.5}$$

Primjer 2.2. *Izrazimo $\sqrt{2}$ rekurzivnom formulom:*

$$a_{n+1} = \frac{1}{2}\left(a_n + \frac{2}{a_n}\right). \tag{2.6}$$

Prva vrijednost aproksimacije a_0 je jednaka:

$$a_0 = 1. \tag{2.7}$$

Formula koristi skup n koji sadrži parametre:

$$n = 0, 1, 2, 3, ... \tag{2.8}$$

Dobiveni će brojevi konvergirati prema $\sqrt{2}$, a vidjet će se kako pogreške aproksimacije mogu biti pozitivne i negativne. Sve se zajedno može prikazati u tablici:

Tablica 2.1. Tablica aproksimacije $\sqrt{2}$

n	a_n^*	$a - a_n^*$
0	1	$+0.414214$
1	1.5	-0.085786
2	1.416667	-0.002453
3	1.414216	-0.000002

U danoj tablici n predstavlja skup prirodnih brojeva koje koristimo u rekurzivnoj formuli za $\sqrt{2}$. a_n^* su rješenja rekurzivne formule, a pogrešku aproksimacije predstavlja $a - a_n^*$.

Omjer apsolutne pogreške i apsolutne vrijednosti nazivamo relativna pogreška, a pišemo kao:

$$\delta a^* = \frac{\Delta a^*}{|a|}. \tag{2.9}$$

Pogreške aproksimacije mogu biti pozitivni i negativni brojevi. Apsolutna pogreška aproksimacije predstavljat će razliku koja odgovara razlici aproksimacije i stvarne vrijednosti.

Stvarna vrijednost najčešće nije poznata, poznata je pogreška aproksimacije. Aproksimacija se uvijek kreće u intervalu $[-\epsilon, \epsilon]$ s time da uvijek vrijedi uvijet $\epsilon > 0$. Prema tome možemo zaključiti da će za apsolutnu pogrešku aproksimacije uvijek vrijediti:

$$-\epsilon \leq a - a^* \leq \epsilon \tag{2.10}$$

ili drugačije zapisano

$$a = a^* \pm \epsilon. \tag{2.11}$$

Realni se iznos može opisati s aproksimiranom vrijednosti uz poznatu pogrešku aproksimacije.

Primjer 2.3. *Zadana je tablica rezultata mjerenja:*

Tablica 2.2. Tablica s rezultatima mjerenja

mjerenje	1	2	3	4	5
R	50.1	49.8	49.95	50.5	50.34

Ako se za vrijednost uzima $R = 50.138$*, a* $\epsilon = 2.138$ *rezultat se obično zapisuje kao* $R = 50.138 \pm 2.138$.

U praksi se često koristi i relacija:

$$\epsilon = \frac{|a - a^*|}{1 + |a^*|}, \tag{2.12}$$

gdje je a stvarna, a a^* aproksimirana vrijednost.

Primjer 2.4. *Izračunajmo relativnu pogrešku aproksimacije za* $\sqrt{2}$ *koristeći se aproksimacijom od* $a^* = 1.41$.

$$\delta a^* = \frac{|\sqrt{2} - 1.41|}{1.41} = 2.9\%. \tag{2.13}$$

2.3. Značajne znamenke

Realni broj a u dekadskom sustavu može se zapisati kao:

$$a = b_m \cdot 10^m + b_{m-1} \cdot 10^{m-1} + ... + b_{m-n} \cdot 10^{m-n}, \tag{2.14}$$

tako da vrijedi:

$$b_m \neq 0, m \in \mathbb{R}. \tag{2.15}$$

Pri tome je b iz skupa $\{0, 1, 2, ..., 9\}$. Elementima ovog skupa je opisan broj a. Prema tome se a^* zapisuje kao:

$$a = b_m^* \cdot 10^m + b_{m-1}^* \cdot 10^{m-1} + ... + b_{m-n}^* \cdot 10^{m-n}. \tag{2.16}$$

Bitno je naglasiti da se a i a^* moraju podudarati u prvih par n znamenki, odnosno mora vrijediti:

$$b_m^* = b_m, b_{m-1}^* = b_{m-1}, ..., b_{m-n}^* = b_{m-n}. \tag{2.17}$$

Zaključujemo da će apsolutna greška Δa^* biti jednaka:

$$\Delta a^* = |a - a^*| = |b_{m-n} - b_{m-n}^*|10^{m-n} + ... \leq |b_{m-n} - b_{m-n}^*|10^{m-n} + ... \tag{2.18}$$

tako da vrijedi:

$$\Delta a^* = |a - a^*| \leq \frac{1}{2}10^{m-n+1}, \tag{2.19}$$

Prikažimo sada na primjeru postupak pronalaženja značajnih znamenki.

Primjer 2.5. *Zadamo broj* $0.000454896 \pm 3 \cdot 10^{-5}$*, a iz ovakvog se zapisa iščitava:*

$$a = 0.000454896 \pm 3 \cdot 10^{-6}, \tag{2.20}$$

pa je

$$m = -4. \tag{2.21}$$

Koristeći se prethodnim izrazom, zaključuje se:

$$\delta a^* = |a - a^*| \leq \frac{1}{2} \cdot 10^{m-n+1}. \tag{2.22}$$

Uvrštavaju se brojevi:

$$3 \cdot 10^{-6} < \frac{1}{2} \cdot 10^{-6}. \tag{2.23}$$

Dobiva se:

$$\frac{1}{2} \cdot 10^{-6} = \frac{1}{2} \cdot 10^{-4-n+1}. \tag{2.24}$$

Rezultat je:

$$n = 9, \tag{2.25}$$

Iz ovog je rješenja vidljivo da su značajne samo prve dvije znamenke broja a^**. Tako da* a^* *zapisujemo kao* $a^* = 0.000458$

Postoji još jedan način na koji možemo definirati značajni broj znamenki broja a^*. Ako definiramo da za najveći pozitivni broj n vrijedi jednadžba:

$$\Delta a^* = \frac{|a - a^*|}{|a^*|} < \frac{1}{2} \cdot 10^{-n+1}. \tag{2.26}$$

Ovom jednadžbom najveći broj za koji je moguće razlika u značajnim znamenkama je jednaka 1.

Primjer 2.6. *Ako imamo broj* 65.12338238 *i taj broj treba zaokružiti na jednu, dvije, tri, pet, sedam decimala. Rezultat bi bio:*

Tablica 2.3. Tablica zaokruživanja značajnih znamenki

N	Rezultat
1	65.1
2	65.12
3	65.124
5	65.12338
7	65.1233824

U tablici N *predstavlja redni broj značajne znamenke na koju je potrebno zaokružiti zadani broj.*

Vidljivo je da se može zaokruživati po pravilu: ako je prva znamenka poslije posljednje značajne znamenke veća od 5*, onda posljednju značajnu znamenku povećamo za* 1*. U suprotnom zapisujemo posljednju značajnu znamenku.*

3. Numeričko integriranje

U ovom poglavlju bavimo se problemom integriranja, tj. računanja određenog integrala, najprije s analitičkog, a zatim sa numeričkog aspekta. Poglavlje je obrađeno prema izvorima [2, 3, 4].

3.1. Integriranje

U kasnom sedamnaestom stoljeću došlo je do prvih ideja vezanih uz integriranje, a glavni začetnici ideje bili su Isaac Newton i Gottfried Wilhelm Leibniz. Integriranje zajedno s deriviranjem čine osnovne alate infinitezimalnog računa. Integral je ključan koncept u višoj matematici. Spaja područja infinitezimalnog računa i matematičke analize.

Integriranje je postupak preko kojega dobivamo površinu ispod zadane krivulje. Funkcija je obično zadana u nekom intervalu i može uključivati beskonačnost. Ako želimo integrirati funkciju f, na intervalu od a do b koristit ćemo se ovom formulom:

$$I(f) = \int_a^b f(x)\,dx, \tag{3.1}$$

gdje je $I(f)$ rješenje integrala funkcije $f(x)$ u intervalu $[a, b]$, $f(x)$ integrirana funkcija, a i b granice integriranja.

Vrlo mali skup funkcija možemo jednostavno integrirati. Zbog toga najčešće ne možemo koristiti osnovni teorem integralnog računanja već ga moramo proširiti s Newton–Leibnitzovom formulom za računanje $I(f)$.

Newton-Leibnitizovom formulom izražava se tvrdnja da je integral funkcije u zatvorenom intervalu jednak razlici vrijednosti njezine primitivne funkcije F u krajnjim točkama intervala. U Newton-Leibnitizovoj formuli $F(a)$ i $F(b)$ primitivne su funkcije od $f(x)$ u krajnjim točkama intervala a i b. Newton-Leibnitizovu formulu zapisujemo kao:

$$I(f) = \int_a^b f(x)\,dx = F(b) - F(a). \tag{3.2}$$

Primitivne funkcije su one funkcije čije će derivacije biti jednake originalnoj funkciji. Algebarski to možemo izraziti kao

$$F'(x) = f(x). \tag{3.3}$$

Kada nije moguće izračunati integral uz pomoć Newton-Leibnitizove formule, onda je potrebno koristi se numeričkom matematikom kako bi se dobila aproksimacija rješenja. Numerička integracija sastoji se od puno raznih algoritama s ciljem izračunavanja brojčane vrijednosti pojedinog integrala. Razlozi korištenja aproksimacije umjesto analitičkih rješenja su sljedeći:

- Ako je funkcija koja se integrira $f(x)$ poznata samo na nekim mjestima, tada nije moguće integriranje Newton-Leibnitizovom formulom. Takvi su slučajevi najčešće prisutni u računalstvu i informatici prilikom strojne obrade podataka.
- Ako znamo funkciju koju želimo integrirati, no nije moguće pronaći njenu primitivnu funkciju. Dobar primjer bio bi:
$$f(x) = e^{-x^2}. \tag{3.4}$$
- Ako je moguće pronaći primitivnu funkciju, ali je puno lakše aproksimirati krajnje rješenje.

Razmotrimo sada problem traženja integrala:

$$I(f) = \int_a^b f(x)\,dx. \tag{3.5}$$

Kao što je već rečeno, nekad nije moguće izračunati integral funkcije $f(x)$ jer nije moguće dobiti primitivnu funkciju $F(x)$ ili je podintegralna funkcija poznata samo u par točaka. Tada se koristimo metodama numeričke integracije kako bi se dobila aproksimacija rješenja.

Kako bi se izračunala aproksimacija rješenja integrala $I(f)$, interpolirat ćemo podintegralnu funkciju f s funkcijom ρ. Aproksimacija glasi:

$$I^* = \int_a^b \rho(x)\,dx. \tag{3.6}$$

Postoji uvjet da funkcija ρ mora zadovoljavati uvjet da je apsolutna vrijednost pogreške aproksimacije ΔI^* manja od pogreške aproksimacije ϵ. Apsolutna vrijednost pogreške aproksimacije ΔI^* je razlika između stvarne vrijednosti I i aproksimirane vrijednosti I^* .

$$\Delta I^* = |I - I^*| < \epsilon, \tag{3.7}$$

pri čemu je zadana točnost $\epsilon > 0$.

3.2. Trapezno pravilo

Trapezno pravilo najjednostavnije je pravilo za numeričko integriranje. Koristi se za određivanje površine ispod krivulje, tako da se krivulja podijeli u dovoljan broj jednakih dijelova. Točnost aproksimacije ovisi o količini podijeljenih dijelova. Što je više dijelova, to je aproksimacija točnija. Dijelovi su trapezi. Aproksimacija je suma površina više trapeza s jednakim osnovicama.

Trapezna formula se dobiva na slijeći način – funkciju $f(x)$ je potrebno interpolirati s interpolacijskim polinomom prvog stupnja. Čvorovi interpolacij su:

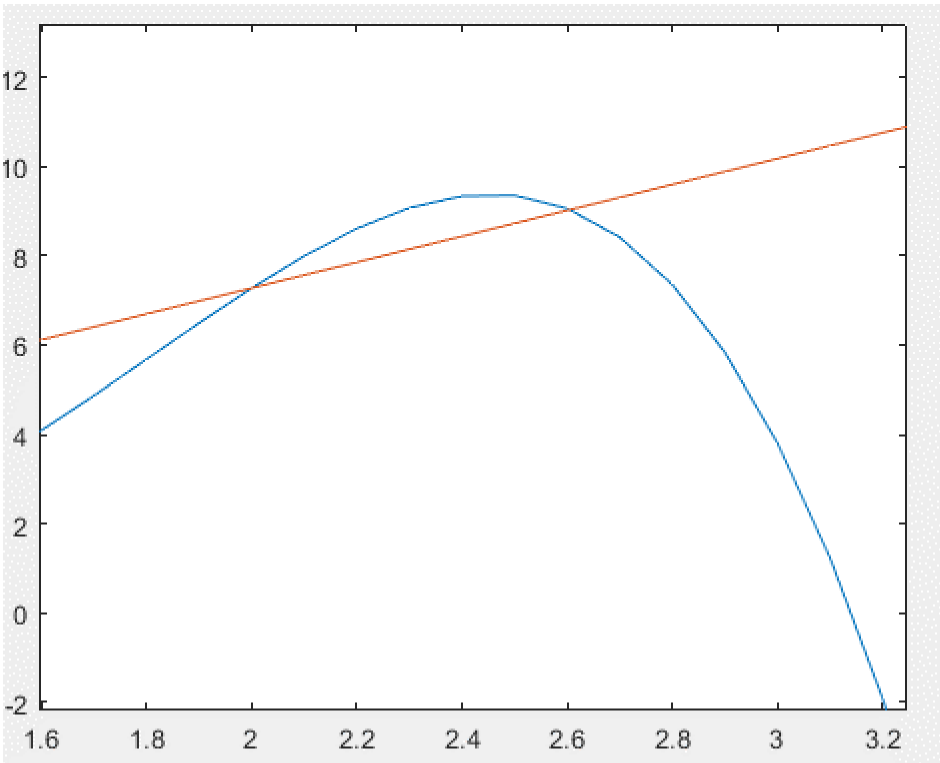
$$x_0 = a, \tag{3.8}$$

$$x_1 = b. \tag{3.9}$$

Dobiven pravac $P_1(x)$ koji siječe funkciju $f(x)$ u točkama $T_0(a, f(a)), T_1(b, f(b))$, a glasi:

$$P_1(x) = f(a) + \frac{f(b) - f(a)}{b - a}(x - a). \tag{3.10}$$

Pravac $P_1(x)$ i možemo prikazati grafički. Pretpostavimo da je $f(x) = x^3 \sin(x)$, zadana na intervalu $[2, 3]$. Iz toga kažemo da je $a = 2$ i $b = 3$. Kako bi se aproksimiralo integral, koristi se jednadžba za pravac $P_1(x)$ da bi aproksimirali područje ispod funkcije $f(x)$ u interval $[a, b]$. Pravac $P_1(x)$ prolazi kroz točke $T_0(a, f(a)), T_1(b, f(b))$ koje se nalaze na sjecištima rubnih intervala funkcije $f(x)$. Prostor omeđen pravcem $P_1(x)$ i rubnim intervalom $[a, b]$ je tražena aproksimacija.



Slika 3.1. Graf aproksimacije trapeznim pravilom

Slika prikazuje funkciju $f(x)$ i pravac $P_1(x)$. Pravcem $P_1(x)$ opisuje se gornja osnovica trapeza kojim se aproksimira funkciju $f(x)$. Plava boja označuje funkciju $f(x)$, a crvenom bojom označen je pravac $P_1(x)$.

Dalje vrijedi:

$$I^* = \int_a^b P_1(x)\,dx = \frac{b - a}{2}(f(a) + f(b)), \tag{3.11}$$

gdje I^* predstavlja površinu trapeza ispod krivulje. Trapez je obrubljen s točkama $f(a)$ i $f(b)$, njegova visina je $h = b - a$.

Točno rješenje integrala sastoji se od aproksimacije i pogreške aproksimacije. Kada je f dovoljno glatka funkcija, tada za neko $c \in (a, b)$ vrijedi izraz za točno rješenje integrala:

$$I^* = \int_a^b f(x)\,dx = \frac{b-a}{2}(f(a) + f(b)) - \frac{(b-a)^3}{12} f''(c). \tag{3.12}$$

Veličina greške E_n ovisi o veličini intervala integracije $[a, b]$. Pogreška se može smanjiti tako da se interval $[a, b]$ podijeli na više pod intervala, a trapezno pravilo primjenjuje se na svaki od pod intervala.

Pretpostavimo da funkciju $f(x)$ poznajemo u $n + 1$ točaka $x_0, ..., x_n \in [a, b]$ te prema tome vrijedi:

$$x_1 - x_0 = ... = x_n - x_{n-1} = h, \tag{3.13}$$

gdje su:

$$x_0 = a, x_n = b. \tag{3.14}$$

Znamo od prije i može se zaključiti da je visina

$$h = \frac{b-a}{n}, \tag{3.15}$$

kao i da se interval $[a, b]$ dijeli točkama $x_0, ..., x_n$ na n jednakih pod intervala.

Za svaki interval primijenimo posebno trapezno pravilo, dok vrijedi:

$$y_i = f(x_i), i = 0, ..., n. \tag{3.16}$$

Na interval $[x_{i-1}, x_i]$ jednadžba će glasiti:

$$\int_{x_{i-1}}^{x_i} f(x)\,dx = \frac{h}{2}(y_{i-1} + y_i) - \frac{h^3}{12} f''(c_i), \tag{3.17}$$

pri čemu je

$$c_i \in (x_{i-1}, x_i). \tag{3.18}$$

Ako je f'' neprekidna, na intervalu $[a, b]$ postoji $c \in (a, b)$ takav da vrijedi:

$$\frac{1}{n} \sum_{i=1}^{n} f''(c_i) = f''(c). \tag{3.19}$$

Opće trapezno pravilo sada glasi:

$$I = I^* + E_n, \tag{3.20}$$

gdje je I točno rješenje, I^* je aproksimacija rješenja, a E_n pogreška aproksimacije.

Aproksimaciju rješenja $I*$ zapisujemo kao:

$$I^* = \frac{h}{2}(y_0 + 2y_1 + ... + 2y_{n-1} + y_n). \tag{3.21}$$

Pogrešku aproksimacije E_n zapisujemo:

$$E_n = -\frac{b-a}{12}h^2 f"(x). \tag{3.22}$$

Primjer 3.1. *Izračunajmo integral funkcije* $f(x) = x^3 - x^2 + x$ *na intervalu* $[5, 9]$.

Pomoći Newton–Leibnitzove formule dobivamo:

$$I = \int_5^9 (x^3 - x^2 + x)\, dx = \frac{x^2(3x^2 - 4x + 6)}{12}\Big|_5^9. \tag{3.23}$$

Uvrštavanjem slijedi:

$$I = 1437.75 - 127.083 = 1310.667 \tag{3.24}$$

Odredimo sada približno rješenje koristeći trapeznu formulu ako je $n = 4$. *Uočimo da je:*

$$\Delta x = \frac{b-a}{x} = \frac{9-5}{4} = 1 \tag{3.25}$$

Raspisujemo formulu:

$$T_n = \frac{\Delta x}{2}[f(5) + 2f(6) + 2f(7) + 2f(8) + f(9)] \tag{3.26}$$

Rješenje je:

$$T_4 = \frac{1}{2}[105 + 372 + 602 + 912 + 657] = 1324 \tag{3.27}$$

Razlika aproksimacije i pravog rješenja je 13.333 *za* $n = 4$.

Ponovimo sada postupak za $n = 5$.

$$\Delta x = \frac{b-a}{x} = \frac{9-5}{5} = 0.8, \tag{3.28}$$

$$T_n = \frac{\Delta x}{2}[f(5) + 2f(5.8) + 2f(6.6) + 2f(7.4) + 2f(8.2) + f(9)]. \tag{3.29}$$

Rješenje je:

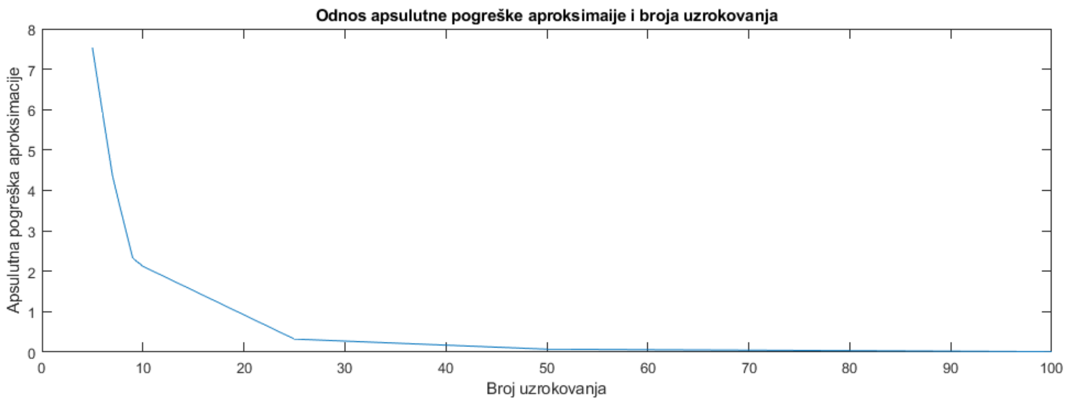
$$T_5 = \frac{0.8}{2}[105 + 2 \cdot 250.536 + 2 \cdot 357.864 + 2 \cdot 492.326 + 657] = 1319.2 \tag{3.30}$$

Iz aproksimacije za broj uzrokovanja $n = 5$ *vidljivo je da je pogreška aproksimacije manja za* 4.8 *u odnosu na broj uzrokovanja* $n = 4$. *Apsolutna greška sada je* 8.533. *Pretpostavka je da* *će se povećanjem broja uzrokovanja apsolutna pogreška smanjiti i time će aproksimacija postati preciznija. Rezultati su zapisani u tablici:*

Tablica 3.1. Tablica aproksimacije funkcije $f(x) = x^3 - x^2 + x$ trapeznim pravilom

n	A^*	ΔA
5	1318.2	7.533
6	1316.59	5.923
7	1315.02	4.343
8	1314	3.323
9	1313	2.323
10	1312.8	2.123
25	1311	0.323
50	1310.75	0.073
100	1310.688	0.011

U tablici n označava broj uzrokovanja, A^* je dobivena aproksimacija, a ΔA apsolutna pogreška aproksimacije. Iz tablice vidi se da je apsolutna pogreška manja što je veći broj uzrokovanja.



Slika 3.2. Odnos apslutne pogreške aproksimaije i broja uzrokovanja

Prikazan je graf koji opisuje odnos apsolutne pogreške i uzrokovanja.

Primjer 3.2. *Zadana je funkcija $f(x) = \sin(x)$ i traži se iznos integrala $I(f)$ na u intervalu $[3.5\pi, 5\pi]$.*

Prvo izračunajmo analitičku vrijednost integrala Newton–Leibnitzovom formulom za računanje $I(f)$. Rješenje glasi:

$$I = \int_{3.5\pi}^{5\pi} \sin(x)\, dx = -\cos(x) \mid_{3\pi}^{5\pi} = 0 - (-1) = 1. \tag{3.31}$$

Sada ćemo ga riješiti koristeći se trapeznim pravilom, uzevši da je $n = 3$.

$$\Delta x = \frac{b-a}{x} = \frac{5\pi - 2\pi}{3} = 0.5\pi. \tag{3.32}$$

Ispisujemo jednadžbu za trapezno pravilo:

$$T_n = \frac{\Delta x}{2}[f(3.5\pi) + 2f(4\pi) + 2f(4.5\pi) + f(5\pi)]. \tag{3.33}$$

Uvrštavamo brojeve u trapeznu jednadžbu:

$$T_3 = \frac{\pi}{4}[0 + 2 + 0 - 2]. \tag{3.34}$$

Dobivamo rezultat:

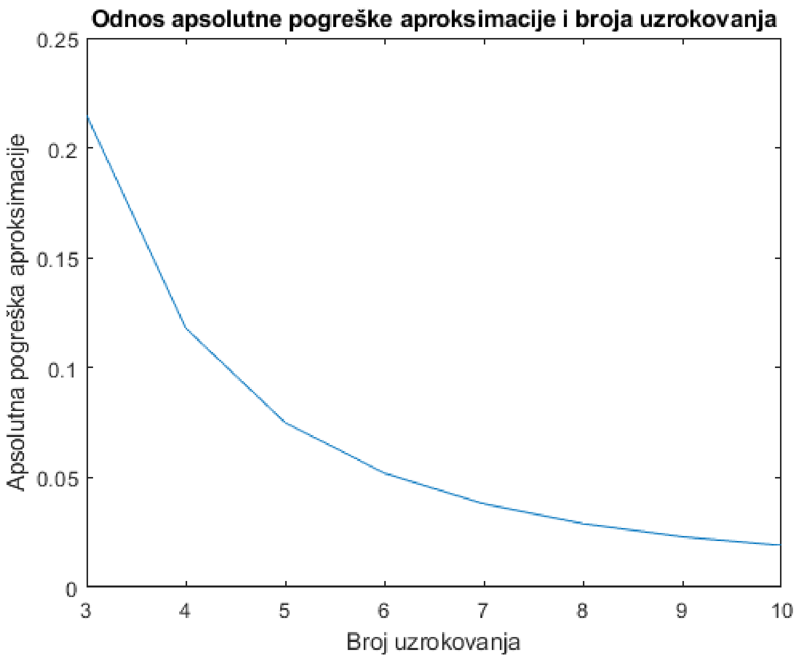
$$T_3 = \frac{\pi}{4}1 = 0.785. \tag{3.35}$$

Ponovno je u pitanju broj uzrokovanja. Raspisujemo tablicu gdje je n broj uzrokovanja A^* aproksimacija rezultata i ΔA apsolutna pogreška aproksimacije.

Slika 3.3. Tablica aproksimacije funkcije $f(x) = \sin(x)$ trapeznim pravilom

n	A^*	ΔA
3	0.785	0.215
4	0.882	0.118
5	0.925	0.075
6	0.948	0.052
7	0.962	0.038
8	0.971	0.029
9	0.977	0.023
10	0.981	0.019

U tablicu n je broj uzrokovanja, A^* dobivena aproksimacija i ΔA apsolutna pogreška aproksimacije.



Slika 3.4. Odnos apslutne pogreške aproksimaije i broja uzrokovanja

Graf prikazuje odnos apsolutne pogreške aproksimacije i broja uzrokovanja. Na apscisi y *nalazi se apsolutna pogreška aproksimacije, a na ordinati* x*, broj uzorkovanja. Vidi se da se apsolutna pogreška smanjuje s brojem uzorkovanja. Drugačije rečeno s većim brojem uzrokovanja imamo točniju aproksimaciju.*

Primjer 3.3. *Zadani su podaci o pumpanju vode. Tablicom su zadani s obzirom na to koliko pumpa može ispumpati litara vode u minuti. Tablica glasi:*

Slika 3.5. Tablica podataka pumpe vode

min	l/min
0	3.8
10	4.5
20	6.2
30	7.0
40	7.5
50	6.9
60	6.2

Tablica prikazuje očitani protok vode koji je pumpa stvorila u trenutku mjerenja. Cilj nam je izračunati ukupan protok vode u promatranom intervalu, što će biti integral dane funkcije prikazane tabelarno.

Tablica ima 7 *redaka koji tvore* 1 *interval i u taj jedan pod interval je podijeljen u* 6*. Što znači da je* $n = 6$*. Granice intervala su* 0 *i* 60*.*

$$\Delta x = \frac{b-a}{x} = \frac{60-0}{6} = 1.0 \tag{3.36}$$

Ispisujemo jednadžbu za trapezno pravilo:

$$I^* = \frac{\Delta x}{2}[f(0) + 2f(10) + 2f(20) + 2f(30) + 2f(40) + 2f(50) + f(60)]. \tag{3.37}$$

Uvrštavamo brojeve u trapeznu jednadžbu:

$$I^* = \frac{10}{2}[3.8 + 2 \cdot 4.5 + 2 \cdot 6.2 + 2 \cdot 7.2 + 2 \cdot 7.5 + 2 \cdot 6.9 + 6.2]. \tag{3.38}$$

Dobivamo rezultat:

$$I^* = 371. \tag{3.39}$$

Primjer 3.4. *Zadamo jednadžbu* $f(x) = x\cot(x)$ *i tražimo iznos integral* $I(f)$ *na u intervalu* $[0, 2]$ *sa zadanim* $n = 4$*.*
Dužina jednog pod intervala za broj pod intervala n=4 je:

$$\Delta x = \frac{b-a}{x} = \frac{2-0}{4} = 0.5. \tag{3.40}$$

Trapezno pravilo glasi:

$$I^* = \frac{\Delta x}{2}[f(0) + 2f(0.5) + 2f(1) + 2f(1.5) + f(2)]. \tag{3.41}$$

Dolazi do problema jer se nezna koliko iznosi $0 \cdot \cot(0)$. *Problem će se riješiti s L'Hospitalovim pravilom.*

$$f(0) = \lim_{x \to 0} x \cot(x). \tag{3.42}$$

Kotangens se može raspisati kao:

$$f(0) = \lim_{x \to 0} x \frac{\cos(x)}{\sin(x)}, \tag{3.43}$$

Kako bi se pronašao limens primjenjuje se L'hospitalovo pravilo:

$$f(0) = \lim_{x \to 0} \frac{x \cos(x)}{\sin(x)} /^{L.H.} \tag{3.44}$$

Primjenom L'hospitalovog pravila dobiva se:

$$f(0) = \lim_{x \to 0} \frac{1 \cos(x) + x(-\sin(x))}{\cos(x)}. \tag{3.45}$$

Uvrštavaju se brojevi:

$$f(0) = \frac{\cos(0) + 0}{\cos(0)}. \tag{3.46}$$

Rezultat glasi:

$$f(0) = 1. \tag{3.47}$$

Dokazano je da $x \cdot \cot(x)$ *konvergira prema* 1 *u točki* $0 \cdot \cot(0)$. *Uvrštavamo brojeve u trapeznu jednadžbu:*

$$I^* = \frac{0.5}{2}[1 + 1.755 + 1.0806 + 0.1447 + 0.915], \tag{3.48}$$

Dobiven je rezultat:

$$I^* = 0.857. \tag{3.49}$$

3.3. Newton-Cotesova formula

Postupak opisan kod izvoda trapeznog pravila može se generalizirati kako bi se postigla bolja aproksimacija integrala. Potrebno je interval $[a, b]$ podijeliti na veći broj podintervala jednale širine. Podintervali bit će podijeljeni s točkama koje nazivamo čvorovima. Potom, kako bi došli do točnije aproksimacije funkciju $f(x)$ aproksimiramo interpolacijskim polinomom n-tog reda u Lagrangeovom obliku:

$$P_n(x) = \sum_{k=0}^{n} f(x_k) p_k(x), \tag{3.50}$$

$$p_k(x) = \prod_{t=0,\ t \neq k}^{n} \frac{x - x_1}{x_k - x_i}, \tag{3.51}$$

pri čemu x_i označava čvor.

Aproksimaciju integrala I^* dobivamo kao:

$$I^*_{(n)} = \int_a^b P_n(x)dx = \sum_{k=0}^{n} f(x_k) \int_a^b p_k(x)dx = (b-a) \sum_{k=0}^{n} \omega_k f(x_k), \tag{3.52}$$

gdje je

$$\omega_k = \frac{1}{b-a} \int_a^b p_k(x)dx, \tag{3.53}$$

a te brojeve nazivamo težinama.

U formulu za I^* aproksimaciju integrala uvodimo supstituciju $x = a + (b-a)t$, te dobivamo:

$$\omega_k = \frac{1}{b-a} \int_a^b p_k(x)dx, \tag{3.54}$$

$$\omega_k = \frac{1}{b-a} \int_a^b \prod_{t=0,\ t\neq k}^{n} \frac{x - x_1}{x_k - x_i} dx, \tag{3.55}$$

$$\omega_k = \int_0^1 \prod_{t=0,\ t\neq k}^{n} \frac{nt - i}{k - i} dt. \tag{3.56}$$

Tako dobivamo sljedeći oblik Newton-Cotesove formule:

$$P_n(x) = (b-a) \sum_{k=0}^{n} \omega_k f(x_k), \tag{3.57}$$

$$\omega_k = \int_0^1 \prod_{t=0\, t\neq k}^{n} \frac{nt - i}{k - i} dt. \tag{3.58}$$

3.4. Simpsonova formula

U 18. stoljeću razvilo se novo pravilo za numeričko izračunavanje određenog integrala. Novo pravilo se naziva Simpsonovo pravilo prema matematičaru Thomasu Simpsonu, u nekim kulturama je pozato i kao "pravilo bačve". Drugi naziv pravilo je dobilo od Johannesa Keplera koji ga je izveo 1615. godine nakon što je vidio kako se pravilo koristi prilikom punjena vinskih bačva.

Kada koristimo kvadratne funkcije za aproksimaciju zadanih funkcija, dobivamo Simpsonovu formulu. Formulu izvodimo tako da uvrstimo $n = 2$ u Newton-Cotesovu formulu.

Prva bitna točka glasi:

$$x_0 = a. \tag{3.59}$$

Druga bitna točka je:

$$x_1 = \frac{a+b}{2}. \tag{3.60}$$

Treća točka je:

$$x_2 = b. \tag{3.61}$$

Težina prve točke prema tome glasi:

$$\omega_0 = \frac{1}{2}\int\limits_0^1 (2t-1)(2t-2)dt = \frac{1}{6}. \tag{3.62}$$

Težina druge točke je:

$$\omega_1 = -\int\limits_0^1 2t(2t-2)dt = \frac{2}{3}. \tag{3.63}$$

Težina treće točke iznosi:

$$\omega_2 = \frac{1}{2}\int\limits_0^1 2t(2t-1)dt = \frac{1}{6}. \tag{3.64}$$

Dobivamo Simpsonovu formulu:

$$I^* = \int\limits_b^a P_2(x)dx = \frac{b-a}{6}\left((f(a) + 4f(\frac{a+b}{2}) + f(b)\right). \tag{3.65}$$

Simpsonova formula zapisanu u ovome obliku aproksimira funkciju $f(x)$ na intervalu $[a, b]$ s polinomom 2. stupnja. Pogreška aproksimacije izražava se preko formule:

$$E = I - I^* = -\frac{(b-a)^{(5)}}{90} f^{(4)}(c), \tag{3.66}$$

tako da vrijedi:

$$c \in (a, b). \tag{3.67}$$

E je pogreška aproksimacije, I točna vrijednost, a $I*$ aproksimirana vrijednost. Pogreška E povećava se s intervalom integracije $[a, b]$. Kako bi smanjili pogrešku, nužno je podijeliti interval $[a, b]$ na paran broj podintervala, $n = 2m$. Njihova se duljina izračunava preko formule, $h = \frac{b-a}{2}$. Podinterval će prema tome biti definiran s relacijom $x_i = a + ih$, $i = 0, 1, .., n$. Kada znamo E pogrešku aproksimaciju, možemo raspisati generalizirano Simpsonovo pravilo:

$$I = I^* + E_n. \tag{3.68}$$

Simpsonovo pravilo govori o tome kako je točno rješenje I zbroj $I*$ aproksimacije rješenja i pogreške aproksimacije E_n. Konačno se dobiva:

$$I^* = \frac{h}{3}((y_0 + y_{2m}) + 4(y_1 + ... + y_{2m-1}) + 2(y_2 + y_{2m-2})). \tag{3.69}$$

Primjer 3.5. *Zadana je funkcija* $f(x) = x^4$ *i traži se iznos integrala* $I(f)$ *pomoću Simpsonove formule na u intervalu* $[2, 10]$.

Dužina jednog podintervala za $n = 4$ *je:*

$$\Delta x = \frac{b-a}{x} = \frac{10-2}{4} = 2. \tag{3.70}$$

Simpsonovo pravilo za $f(x)$ *tada glasi:*

$$I^* = \frac{\Delta x}{3}[f(2) + 4f(4) + 2f(6) + 4f(8) + f(10)]. \tag{3.71}$$

Uvrštava se brojeve u jednadžbu:

$$I^* = \frac{2}{3}[16 + 1024 + 2592 + 16384 + 10000]. \tag{3.72}$$

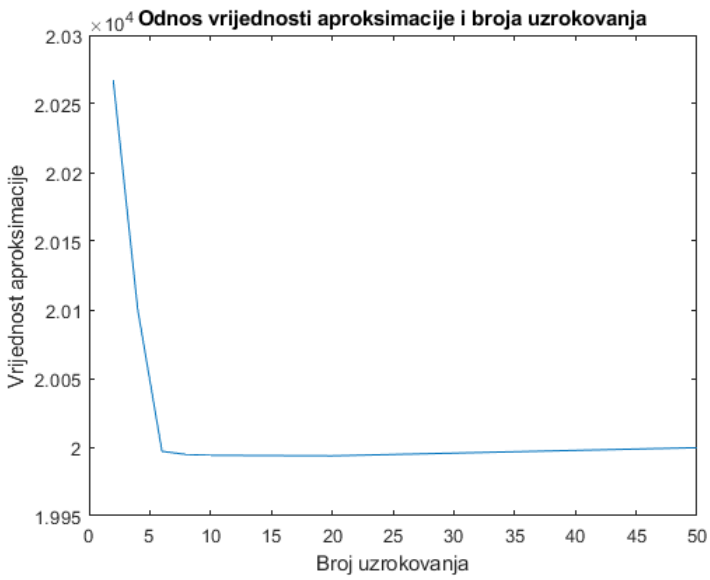
Dobiven je rezultat:

$$I^* = 20100.666, \tag{3.73}$$

dok je točna vrijednost integrala 19993.6.

Tablica 3.2. Tablica aproksimacije funkcije $f(x) = x^4$ *Simpsonovim pravilom*

n	I^*	ΔA
2	20266.66	273.06
4	20100.666	107.066
6	19996.972	3.372
8	19994.66	1.06
10	19994.036	0.436
20	19993.67	0.07
50	19993.601	0.01



Slika 3.6. Odnos vrijednosti aproksimacije i broja uzrokovanja

U tablici n označava broj uzrokovanja, A^* je dobivena aproksimacija, ΔA apsolutna pogreška aproksimacije.

Graf prikazuje aproksimiranu vrijednost integrala I^* s obzirom na broj uzrokovanja ili bolje rečeno broj pod intervala n. Na apscisi se nalazi broj uzrokovanja n, a na ordinati aproksimirana vrijednost integrala I^*.

Može se uočiti da je rezultat već zadovoljavajući nakon 6 uzrokovanja. Pogreška iznosi 3.372 za 4 uzrokovanja dobivamo pogrešku od 107.066. Pogreška je 0.168% ukupnog rezultata. Može se reći da već na 6 uzrokovanja imamo dobar rezultat. Može se vidjeti da se povećava točnost, a time i smanjuje pogreška s brojem uzrokovanja proporcionalno s N^4.

Primjer 3.6. *Zadana je tablica s podacima brzine automobila. Tablica glasi:*

Tablica 3.3. Tablica brzine auta

min	m/s
0	25
5	28
10	32
15	30
20	29
25	26
30	23

Tablica govori o izmjerenim brzinama automobila u danim trenucima, a potrebno je izračunati ukupni put što se svodi na računanje integrala.

Tablica ima 7 redaka koji tvore 1 interval i u taj jedan pod interval je podijeljen u 6 što znači da je broj uzrokovanja n jednak 6. Granice intervala su 0 i 30. Iz tih podataka može se dobiti da vrijeme jednog podintervala je:

$$\Delta x = \frac{b-a}{x} = \frac{30-0}{6} = 5min. \tag{3.74}$$

Dobiveno vrijeme potrebno pretvoriti iz minuta u sekunde. Tako da nova dužina podintevala glasi:

$$\Delta x = \frac{5min}{1}\frac{60s}{1min} = 500s. \tag{3.75}$$

Ispisujemo jednadžbu za Simpsonovo pravilo:

$$I^* = \frac{\Delta x}{2}[f(0) + 4f(5) + 2f(10) + 4f(15) + 2f(20) + 4f(25) + f(30)]. \tag{3.76}$$

Uvrštavamo brojeve u jednadžbu:

$$I^* = \frac{300}{2}[25 + 4 \cdot 2.8 + 2 \cdot 32 + 4 \cdot 30 + 2 \cdot 29 + 4 \cdot 26 + 23]. \tag{3.77}$$

Dobivamo rezultat:

$$I^* = 50600m = 50.6km. \tag{3.78}$$

Primjer 3.7. *Zadamo jednadžbu* $f(x) = \frac{\sin(x)}{x}$ *i tražimo iznos integral* $I(f)$ *na u intervalu* $[0, 1]$ *sa zadanim* $n = 2, 4, 10$. *Dužina jednog pod intervala za broj podintervala n=2 je:*

$$\Delta x = \frac{b-a}{x} = \frac{1-0}{2} = 0.5. \tag{3.79}$$

Trapezno pravilo glasi:

$$I^* = \frac{\Delta x}{2}[f(0) + 4f(0.5) + f(1)]. \tag{3.80}$$

Potrebno je naći u koju točku konvergira $f(x) = \frac{\sin(x)}{x}$ *kada se prilazi* 0. *Postupak glasi:*

$$f(0) = \lim_{x \to 0} \frac{\sin(x)}{x}. \tag{3.81}$$

Koristse L'Hospitalovo pravilo:

$$f(0) = \lim_{x \to 0} \frac{\sin(x)}{x} /^{L.H.}, \tag{3.82}$$

dobiva se:

$$f(0) = \lim_{x \to 0} \cos(x) \tag{3.83}$$

Uvrštavaju se brojevi:

$$f(0) = \cos(0), \tag{3.84}$$

i dobiva se rezultat:

$$f(0) = 1. \tag{3.85}$$

Dokazano je da funkcija $f(x) = \frac{\sin(x)}{x}$ *konvergira prema* 1 *u točki* $\frac{\sin(0)}{0}$. *Uvrštavamo brojeve u trapeznu jednadžbu:*

$$I^* = \frac{0.5}{2}[1 + 3.8354 + 0.84147]. \tag{3.86}$$

Dobiven je rezultat:

$$I^* = 1.4192. \tag{3.87}$$

Dužina jednog pod intervala za broj pod intervala n=4 je:

$$\Delta x = \frac{b-a}{x} = \frac{1-0}{2} = 0.25. \tag{3.88}$$

Trapezno pravilo glasi:

$$I^* = \frac{\Delta x}{2}[f(0) + 4f(0.25) + 2f(0.5) + 4f(0.75) + f(1)]. \tag{3.89}$$

Uvrštavamo brojeve u trapeznu jednadžbu:

$$I^* = \frac{0.25}{2}[1 + 3.9584 + 1.9177 + 3.6354 + 0.84147]. \tag{3.90}$$

Dobiven je rezultat:

$$I^* = 1.4191. \tag{3.91}$$

Dužina jednog pod intervala za broj pod intervala n=10 je:

$$\Delta x = \frac{b-a}{x} = \frac{1-0}{10} = 0.1. \tag{3.92}$$

Trapezno pravilo glasi:

$$\begin{aligned} I^* = \frac{\Delta x}{2}[&f(0) + 4f(0.1) + 2f(0.2) + 4f(0.3) + 2f(0.4) + 4f(0.5) \\ &+2f(0.6) + 4f(0.7) + 2f(0.8) + 4f(0.9) + f(1)]. \end{aligned} \tag{3.93}$$

Uvrštavamo brojeve u Simpsonovu jednadžbu:

$$\begin{aligned} I^* = \frac{0.1}{2}[&1 + 3.999 + 1.9867 + 3.94 + 1.9471 + 3.8354 \\ &+1.8821 + 3.681 + 1.7934 + 3.481 + 0.84147]. \end{aligned} \tag{3.94}$$

Dobiven je rezultat:

$$I^* = 1.4164. \tag{3.95}$$

3.5. Usporedba trapeznog i Simpsonovog pravila

U sljedećem primjeru razmotrit ćemo razliku između numeričkih metoda kako bi se lakše mogla utvrditi njihova moguća primjena.

Primjer 3.8. *Potrebno je odrediti apsolutnu pogrešku aproksimacije trapezne i Simpsonove formule za:*

$$\int\limits_{-2}^{2} e^{0.3x} \sin(10x)\, dx. \tag{3.96}$$

Prvo je potrebno izračunati točnu vrijednost kako bi je kasnije usporedili s aproksimacijama. Integracijom početnog izraza dobivamo:

$$\left. \frac{10e^{0.3x}(3\sin(10x) - 100\cos(10x))}{1009} \right|_{-2}^{2}. \tag{3.97}$$

Uvrštavamo brojeve. Dobivamo:

$$\frac{e^{-\frac{3}{5}}(30\sin(20) + 1000\cos(20))}{1009} + \frac{e^{\frac{3}{5}}(30\sin(20) - 1000\cos(20))}{1009}. \tag{3.98}$$

Daljnjim računom slijedi:

$$\frac{e^{-\frac{3}{5}}((30e^{-\frac{6}{5}} + 30)\sin(20) + (1000 - 1000e^{-\frac{6}{5}}\cos(20))}{1009}, \tag{3.99}$$

te je dobiven rezultat:

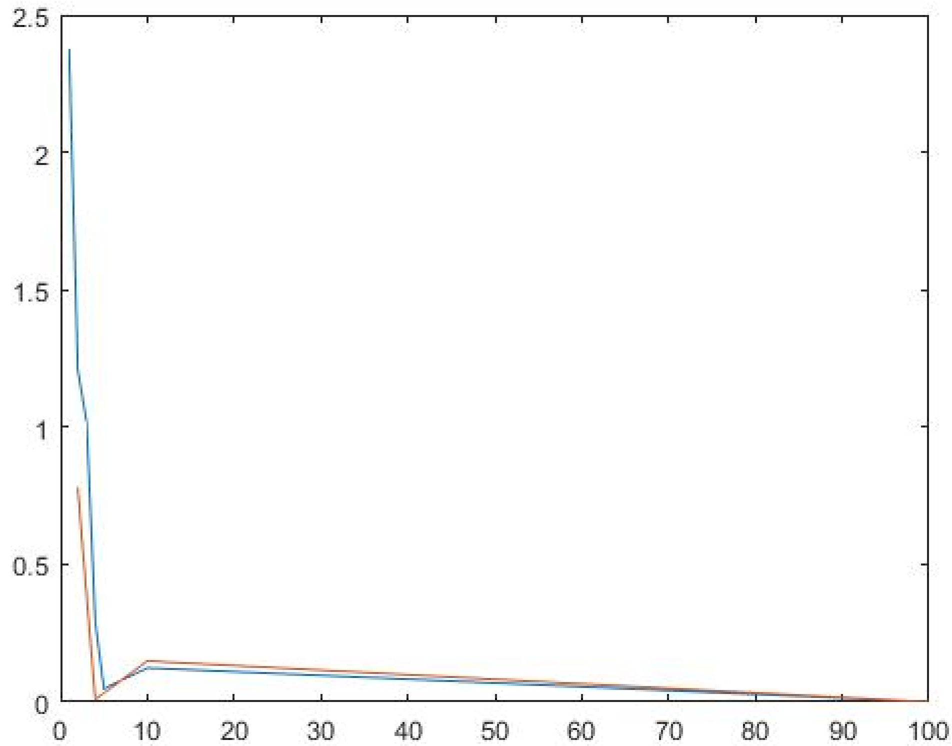
$$-0.045426908. \tag{3.100}$$

Kada je riješen integral. Ispisuje se tablica s rješenjima aproksimacije integracije.

Tablica 3.4. Tablica usporedbe aproksimacije trapezne i Simpsonove metode

n	Trapezna	ΔA_{Trap}	Simpsonova	ΔA_{Simp}
1	2.3249	2.370326	/	/
2	1.16245	1.207876	0.7749	0.7749
3	0.97585	1.021276	/	/
4	0.24989	0.295316	-0.05428	$8.854 \cdot 10^{-3}$
5	0.00354	0.0485966	/	/
10	0.0787	0.12413	0.1037628	0.14918
100	-0.044644	$7.82 \cdot 10^{-4}$	-0.045437	$1.1 \cdot 10^{-5}$

U tablici su navedeni podaci o rezultatima aproksimacije integracije i njihovim apsolutnim pogreškama. Koristeći se trapeznim i Simpsonovim pravilom dobili su se rezultati aproksimacije koje smo nazvali "Trapezna" i " Simpsonova". Dobiveni rezultati oduzeti su od prave vrijednosti integrala te tako je dobivena apsolutna greška aproksimacije za obje metode. Apsolutne pogreške nazvane su ΔA_{Trap} i ΔA_{Simp}.



Slika 3.7. Graf usporedbe aproksimacije trapezne i Simpsonove metode

Prikazan je graf usporedbe aproksimacije trapezne i Simpsonove metode gdje se na apscisi nalazi broj uzrokovanja a, na ordinati apsolutne vrijednosti pogrešaka aproksimacija. Plavom *bojom prikazani su rezultati dobiveni trapeznom metodom, dok crvenom su prikazani rezultati dobiveni Simpsonovom metodom.*

Simpsonovo pravilo moguće je samo primijeniti na parnim uzrokovanjima n. *Promatrajući graf, vidi se da u većini slučajeva Simpsonovo pravilo točnije opisuje graf od trapeznog pravila. Razlog je da trapezno pravilo za isti broj uzrokovanja* n *koristi veći broj točaka. Kako bi se pod interval opisao polinomom, potrebne su tri točke. Pod interval se dijeli na dva dijela i dobiva novu točku. Trapezno pravilo ima prednost jer zahtjeva manje koraka za izračun.*

4. Implementacija algoritama za numeričku integraciju u programu Python

Python je programski jezik opće namjene visoke razine. Dizajniran je da bude što jednostavniji za korištenje. S pomoću Pythonove jezične konstrukcije i objektno orijentiranog pristupa jasno i lagano može se napisati novi algoritam. Python je 1980-ih osmislio Nizozemac Guido van Rossum kao poboljšanju verziju programskog jezika ABC. Oba su programska jezika bili projekti koje je radio u svoje slobodno vrijeme. Programi su bili inspirirani programom SETL koji je služio za rukovanje operativnim sustavom Amoeba. Programi su postali javno dostupni tek 1989. godine. Ovo je poglavlje obrađeno prema [5].

4.1. Naivna numerička integracija u programskome jeziku Python

Za izračun aproksimacije određenog integrala funkcije $f(x)$ možemo koristiti sumu površina pravokutnika ispod funkcije. Takav je pravokutnik omeđen točkama:

$$(x_{i-1}, 0), (x_i, 0), (x_{i-1}, f(x_{i-1})), (x_i, f(x_i)). \tag{4.1}$$

Pravokutnici koji se koriste za određivanje mogu imati jednake podintervale koji se određuju preko formule za duljinu podintervala $\Delta x = (b - a)/N$. Jedan podinterval možemo zapisati i kao $[x_{i-1}, x_i]$. Površinu pravokutnika opisujemo s formulom:

$$(x_i - x_{i-1})f(x_{i-1}) + \frac{1}{2}(x_i - x_{i-1})(f(x_i) - f(x_{i-1})), \tag{4.2}$$

odnosno.

$$\frac{1}{2}(x_i + x_{i-1})(x_i - x_{i-1}). \tag{4.3}$$

Formula predstavlja sumu pravokutnika u intervalu aproksimacije integrala $[a, b]$. Drugim riječima, dobili smo izraz koji možemo koristiti za aproksimaciju integrala:

$$T_N(f) = \frac{\Delta x}{2} \sum_{i=1}^{N} f((x_i) + f(x_{i-1})), \tag{4.4}$$

pri čemu vrijedi:

$$x_i = a + i\Delta x. \tag{4.5}$$

Ovaj izraz jednostavno bi se mogao kodirati unutar programskog jezika Python, no takav pristup je kompleksan, a i sama aproksimacija pravokutnicima može voditi na veću pogrešku. Stoga u nastavku predstavljamo ugrađene funkcije Pythona za približni izračun integrala.

4.2. Implementacija trapeznog pravila u programskome jeziku Python

Kako bi mogli izračunati integral neke funkcije unutar programskog jezika Python, najprije ju trebamo definirati, što radimo na sljedeći način:

```
def f(x):
return 1/(1 + x**2)
```

U definiciji smo definirali funkciju $f(x)$ će se integrirati. Funkcija glasi:

$$f(x) = \frac{1}{1+x^2}. \tag{4.6}$$

Sada ćemo definirati funkciju *trapezoidal* čija uloga biti implementacija trapezne formule. Zaglavlje funkcije definirano je s:

```
def trapezoidal(x0,xn,n):
```

Definirana funkcija će u sebi sadržavati funkcije i petlje pomoću kojih će se dobiti algoritam koji će izvršiti trapezno pravilo. Funkcija „trapezoidal“ sadrži parametarske varijable. Varijabla $x0$ je početak intervala, xn je kraj intervala i n je broj pod intervala ili broj uzrokovanja. Treba još reći da n mora biti prirodni broj dok $x0$ i xn mogu biti realni brojevi.

Sljedeće naredbe nalaze se unutar definicije trapeznog pravila.

Prvi korak u izračunavanju aproksimacije određivanje je širine jednog podintervala. Širinu podintervala nazovimo h. Taj izraz se u Pythonu zadaje s:

```
h = (xn - x0) / n
```

Dobiveni broj h uvijek će biti float ili drugim riječima realni broj.

Kao poseban element uvest ćemo sumu prvog i zadnjeg elementa funkcije $f(x)$ u intervalu od $[x0, xn]$. Novu varijablu kojoj smo pridružili sumu, nazvali smo integration. To radimo sljedećim naredbama:

```
integration = f(x0) + f(xn)
```

Sljedeći dio koda sadrži brojač kojemu je cilj $n-1$ puta ponoviti kod. Postiže se tako da se varijabla i postavlja na vrijednost jedan, potom svaki „krug“ funkcije povećava za jedan dok ne prestane vrijediti uvjet i>n. Uvjet dolazi iz funkcije „for i in range()“. NIz naredbi koji implementira opisano, zadan je s:

```
for i in range(1,n):
k = x0 + i*h
integration = integration + 2 * f(k)
```

Unutar petlje prvo za svaki novi i, računamo varijablu k, koja predstavlja korak unutar računa. Dobivenu varijablu k uvodimo kao element u funkciju $f(x)$ i time izračunavamo iznos $f(k)$. Element $f(k)$ se množi s 2 i zbraja s varijablom integration. Ako je prvi „krug“ petlje, onda je iznos varijable integration jednak sumi vrijednosti funkcije na početku i kraju intervala $f(x0) + f(xn)$. Za sve će ostale „krugove“ integration imati pridruženu vrijednost $2 * f(k)$ i vrijednost varijable integration će se mijenjati svaki „krug“ petlje.

Kako bi se dobila konačna aproksimacija, potrebno je varijablu integration pomnožiti s dužinom pod intervala h te je podijeliti s 2. To vidimo u sljedećem dijelu koda:

```
integration = integration * h/2
```

Na kraju se samo vraća iznos funkcije nakon što je pozovemo.

```
return integration
```

Pri pozivu funkcije moramo odrediti granice i odrediti broj podintervala ili drugim riječima koraka. To radimo sljedećim nizom naredbi:

```
lowerlimit = float(input("Unesite donju granicu integracije: "))
upperlimit = float(input("Unesite gornju granicu integracije: "))
sublnterval = int(input("Unesite željeni broj podintervala "))
```

Sada možemo pozvati funkciju za računanje integrala trapeznom formulom.

```
result = trapezoidal(lowerlimit, upperlimit, subinterval)
print(result)
```

Postoji još jedan način rješavanja problema s trapeznom formulom, a to je korištenje gotove funkcije iz knjižnice. Knjižnice su programski paketi koji sadrže već gotove funkcije. U paketu SciPy postoji niz funkcija za aproksimaciju određenih integrala i numeričko rješavanje diferencijalnih jednadžbi. Iz knjižnice je potrebna funkcija scipy.integrate.trapz koja izračunava aproksimaciju određenog po pravilu trapeza. Sintaksa je vrlo jednostavna i glasi:

```
scipy.integrate.trapz(y, x, dx, axis)
```

U funkciju se "scipy.integrate.trapz(y, x, dx, axis)" uvrštavaju varijable. Jedna od tih varijabli je y koja predstavlja funkciju koju će se integrirati. Varijabla x je niz ili interval. Nije obavezno kao ulaz. Ako nije zadan, onda se koristi dx. Prema zadanim postavkama je 0. dx je razmak između točaka, skalarna vrijednost koja je također izborni ulaz. Koristi se kada ne koristimo x i prema zadanim postavkama je 1. $axis$ je cjelobrojna vrijednost koja određuje os integracije.

Ako samo koristimo y:

```
y = [3,4,5,6]
print(np.trapz(y))
```

Izlaz je: 13.5.

Koristimo li x i y:

```
y = [3,4,5,6]
x = [1,2,3,4]
print(np.trapz(y,x))
```

Izlaz je 13.5, a račun je potpuno jednak kao kada koristimo samo y jer su dužine pod intervala bile jednake.

Koristimo li y i dx:

```
y = [3,4,5,6]
print (np.trapz(y,dx = 2))
```

Izlaz je 27.

Račun je drugačiji i pogrešan. Razlog toga je što dx je dužina pod intervala, a y je niz brojeva na kojima bi se trebala vršiti integracija. U normalnom računu oni su ovisni jedan o drugome, ali ovdje su fiksni. Sugerirali smo da dužina pod intervala y je 1, a s dx da je 2. Odnosno:

$$\frac{2}{2}(3 + 4 \cdot 2 + 5 \cdot 2 + 6) = 27. \tag{4.7}$$

4.3. Implementacija Simpsonove formule u programskome jeziku Python

Simpsonovo pravilo koristi kvadratni polinom za svaki podinterval radi aproksimacije funkcije $f(x)$ i za izračunavanje aproksimacije određenog integrala $I*$. Podsjetimo se da Simpsonova formula za jedan podinterval glasi:

$$I^* = \int_b^a P(x)dx = \frac{b-a}{6}((f(a) + 4f(\frac{a+b}{2}) + f(b)), \tag{4.8}$$

gdje su a i b rubovi zadanog intervalan. Zbog preciznosti podijelimo interval na više pod intervala s povećanjem broja uzrokovanja N. Simpsonova formula koju ćemo koristi u algoritmu tada glasi:

$$S_N(f) = \frac{\Delta x}{3} \sum_{i=1}^{N/2} (f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})). \tag{4.9}$$

,

Ovdje je N broj podintervala u intervalu $[a, b]$. Dužina podintervala Δx se određuje preko formule

$$\Delta x = (b - a)/N. \tag{4.10}$$

Udaljenost između čvorova x_i je određena preko formule $x_i = a + i\Delta x$.

Pokažimo sada kako bi se Simpsonova formula implementirala u programski jezik Python.

Postupamo slično kao i kod trapeznog pravila. Najprije definiramo funkciju:

```
def f(x):
return 1/(1 + x**2)
```

Sada definiramo proceduru za korištenje Simpsonove formule, sa sljedećim zaglavljem:

```
def simpson13(x0,xn,n):
```

Kodom "def simpson13(x0,xn,n):" izrađujemo novu funkciju. Unutar funkcije bit će pohranjen algoritam potreban za aproksimaciju integrala Simpsonovom formulom. Prilikom definiranja funkcije Simpsonove formule potrebno je naglasiti tri varijable. Početak intervala integracije $x0$, kraj intervala aproksimacije integracije xn i broj pod intervala ili uzrokovanja n.

Nakon što smo definirali Simpsonovu funkciju, moramo implementirati novu varijablu za dužinu pod intervala. Dužinu pod intervala nazvat ćemo h i njoj ćemo dodijeliti iznos razlike vrijednosti rubova intervala $[x0, xn]$ podijeljenih s brojem pod intervala n.

```
h = (xn - x0) / n
```

Kako bi se pojednostavio algoritam, prvo implementiramo novu varijablu integration. Varijabli integration dodajemo vrijednosti zbroja iznosa funkcije $f(x)$ u točkama $f(x0)$ i $f(xn)$.

```
integration = f(x0) + f(xn)
```

Pomoću koda "for i in range(1,n):" postavljamo jednostavnu petlju koja se ponavlja n puta. Dok se petlja ponavlja, sljedeći se dio koda izvršava.

```
for i in range(1,n):
```

Novoj varijabli k za svaki ciklus dodijelimo zbroj počeka intervala i „udaljenosti od početka intervala“. „Udaljenosti od početka intervala“ dobili smo tako što smo širinu pod intervala pomnožili s brojem i. i se uvećava za jedan kako se izvrši ciklus.

```
k = x0 + i*h
```

Iznos funkcije $f(x)$ u točki k svaki se ciklus mijenja i kako bi se skratio kod zbraja odmah s varijablom $integration$, utvrdi se kojim se brojem množi. U kodu $integration = integration + 2 * f(k)$ varijabla integration zbraja se s $2 * f(k)$ te pridjeljuje se sama sebi, odnosno varijabli integration. Ovo je moguće jer imamo zadan iznos varijable integration i varijable integration pamti svoju vrijednost iz svakog ciklusa. Time će integration na zadnjem ciklusu iznositi zbroj svih pod intervala i granica intervala.

```
if i%2 == 0:
integration = integration + 2 * f(k)
else:
integration = integration + 4 * f(k)
```

Za razliku od opće trapezne formule u općoj Simpsonovoj formuli nije „srednji dio“ formule pomnožen samo s 2, već i sa 4 i to naizmjenično. Prvi iznos granica pod interval poslije početka intervala bit će pomnožen s brojem 4. Iznos sljedeće granice pod intervala s brojem 2 i tako dok se ne stigne do kraja intervala. Kako bi se proces opisao algoritmom, postavljamo uvijete s funkcijom "if", "else" te svakom pod intervalu dodjeljujemo njegov redni broj n. Iz danog vidi se da svi su pod intervali, koji su višekratnici broja 2, pomnoženi s 2, a ostali s 4. Najlakši način provjere je li broj višekratnik broja je provjera ostatka. Ostatak, da bi zadovoljio uvjet, mora biti 0.

Konačna vrijednost aproksimacije integracije jednaka je vrijednosti integration pomnoženoj s dužinom pod intervala h koji smo prethodno podijelili s 3. Kako bi program radio pomoću funkcije "return", program vraća iznos varijable integration kao vrijednost funkcije "simpson13(x0,xn,n)".

```
integration = integration * h/3
return integration
```

Korisnik zapisuje ulazne parametre:

```
x0 = float(input("Donji limit integracije: "))
xn = float(input("Gornji limit integracije: "))
n = int(input("Broj pod intervala: "))
```

Ispisuje se rezultat:

```
rezultat = simpson13(x0,xn,n)
print("Rezultat: ",rezultat)
```

Drugi način korištenja Simsonove formule u Phytonu je uz paket SciPy. Potrebna funkcija "scipy.integrate.simps" koja izračunava aproksimaciju određene funkcije po Simpsonovoj formuli. Sintaksa glasi:

```
scipy.integrate.simps (y, x, dx, axis, even)
```

Unutar funkcije imamo varijable $y, x, dx, axis, even$. Varijabla y predstavlja funkciju koju se želi integrirati. x je brojevni niz koji predstavlja interval integracije. Nije obavezno kao ulaz, a ako nije zadan koristi se dx. Prema zadanim postavkama je 0. dx je razmak između točaka. Skalarna je vrijednost koja je također izborni ulaz. Prema zadanim postavkama je 1. $axis$ je cjelobrojna vrijednost koja određuje os integracije. $even$ je opcionalan. Postoje 3 moda rada. Mod avg koristi trapezoidno pravilo na prvom i zadnjem pod intervalu. "Mod first" koristi trapezoidno pravilo na zadnjem interval, te "mod last" koje koristi trapezno pravilo na prvom intervalu.

Implementacija bi izgledala ovako:

```
import numpy as np
from scipy import integrate

x = np.arange(0, 10)
y = np.arange(0, 10)
gfg = integrate.simps(y, x)
print(gfg)
```

Rješenje je: 40.5.

```
import numpy as np
from scipy import integrate

x = np.arange(0, 10)
y = np.sqrt(x)
gfg = integrate.simps(y, x)
print(gfg)
```

Rješenje je: 17.875.

5. Primjena numeričkih metoda u elektrotehnici

Primjer za numeričku integraciju moguće je pronaći u regulacijskoj tehnici. U regulacijskoj tehnici za regulaciju koriste se povratne veze koje rade pomoću regulatora. Regulator je mehanizam upravljačke petlje koji koristi povratnu informaciju kako bi kontinuirano modulirano upravljanje električnog stroja ili nekog drugog električnog ulaza, a signal koji regulator dobiva od mjernih senzora na električnom stroju nazivamo povratnom vezom. Među brojnim regulatorima imamo proporcionalno integralni regulator (PI) i proporcionalno integralno derivacijski regulator (PID). Oba regulatora koriste integral po vremenu u svojim logičkim procesima.

Pokažimo opisanu implementaciju numeričke integracije na jednom primjeru.

Primjer 5.1. *Pretpostavlja se da je na PID regulator doveden signal koji je opisan funkcijom*

$$f(x) = 2.3sin(x)^{-0.7x}. \tag{5.1}$$

Potrebna je provjera svakog dijela regulatora, tako i integralnog dijela. Nepredviđena greška na bilo kojemu dijelu u PID regulatora bi izazvala grešku koja bi poremetila krajnji rezultat. Kako se integriranje u PID regulatorima vrši preko aproksimacije trapeznim pravilom, najjednostavnija provjera je pomoću trapeznog pravila.

Integral funkcije signala glasi:

$$\int_0^{2\pi} 2.3\sin(x)e^{-0.7x}\,dx. \tag{5.2}$$

Koristi se trapezno pravilo prilikom čega broj uzrokovanja je $n = 2$.

$$I_2 = \frac{\Delta x}{2}[f(0) + 2f(\pi) + f(2\pi)]. \tag{5.3}$$

Uvrštavaju se brojevi u trapeznu jednadžbu:

$$I_2 = \pi[0 + 0 + 0]. \tag{5.4}$$

Dobiva se rezultat:

$$I_2 = 0. \tag{5.5}$$

Rješenje nije dobro, što možemo provjeriti i direktnom analitičkom integracijom. Potreban je veći broj uzrokovanja n. Uzimamo za novi primjer da je $n = 6$. Tada je:

$$I_6 = \frac{\Delta x}{2}[f(0) + 2f(\frac{\pi}{3}) + 2f(\frac{2\pi}{3}) + 2f(\pi) + 2f(\frac{4\pi}{3}) + 2f(\frac{5\pi}{3}) + f(2\pi)]. \tag{5.6}$$

Nakon što brojeve uvrstimo u jednadžbu dobivamo:

$$I_6 = \frac{\pi}{6}[0 + 1.91396 + 0.91955 + 0 - 0.21226 - 0.10198 + 0]. \tag{5.7}$$

Dobiveni rezultat je:

$$I_6 = 1.31909. \tag{5.8}$$

Za provjeru rezultata potrebno je izračunati explicitno rješenje funkcije.

$$\int_0^{2\pi} 2.3\sin(x)e^{-0.7x}\,dx. \tag{5.9}$$

$$I = 1.52463. \tag{5.10}$$

Kako znamo aproksimaciju i točno rješenje, možemo naći pogrešku aproksimacije ΔA. Ona će iznositi:

$$\Delta A = Tocno\ rjesenje - Aproksimacija\ rjesenja, \tag{5.11}$$

$$\Delta A = 1.52463 - 1.31909, \tag{5.12}$$

$$\Delta A = 0.20554. \tag{5.13}$$

Poznavajući pogrešku aproksimacije ΔA, možemo izračunati relativnu pogrešku aproksimacije $|\epsilon|$:

$$|\epsilon| = |\frac{Pogreska\ aproksimacije}{Tocno\ rjesenje}| \cdot 100\%, \tag{5.14}$$

Uvrštavaju se brojevi, te se dobiva:

$$|\epsilon| = |\frac{0.20554}{1.52463}| \cdot 100\%, \tag{5.15}$$

Relativna pogreška aproksimacije $|\epsilon|$ je:

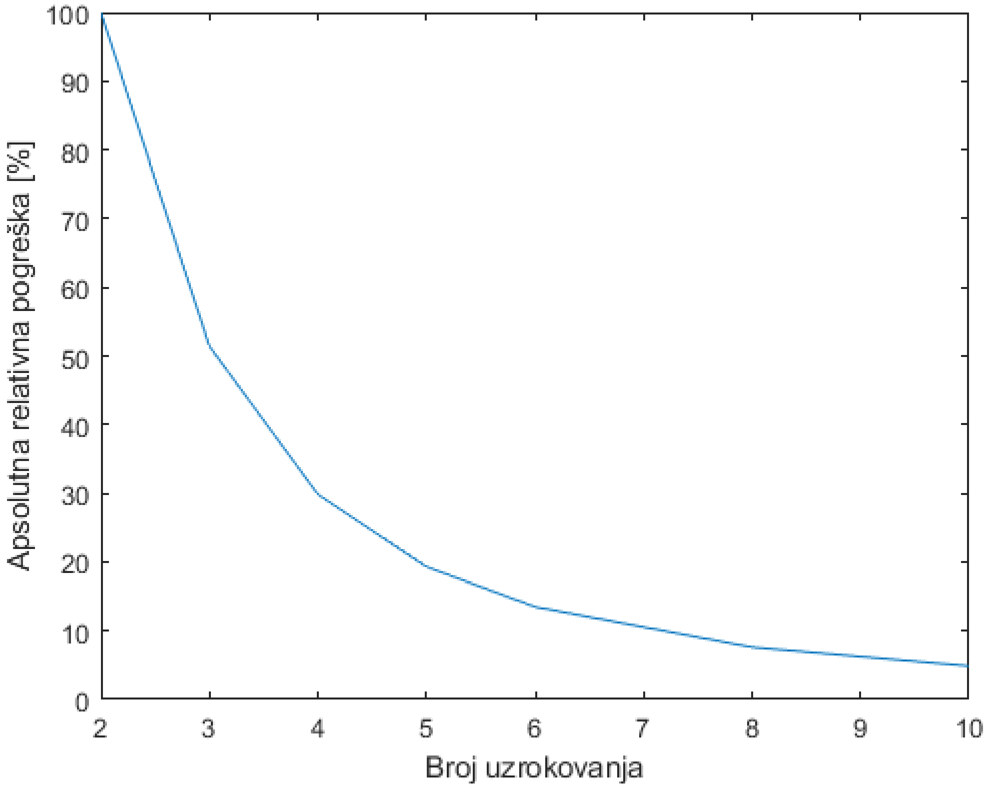
$$|\epsilon| = 13.481\%. \tag{5.16}$$

Tablica 5.1. Tablica aproksimacije PID regulatora

n	I^*	ΔA	$\|\epsilon\|$
2	0	1.52463	100%
3	0.74068	0.78395	51.419%
4	1.06979	0.45484	29.833%
5	1.2302	0.29443	19.312%
6	1.31909	0.20554	13.481
8	1.40848	0.11615	7.618
10	1.45014	0.07449	4.886

U tablici n predstavlja broj uzrokovanja, I^* dobivenu aproksimaciju, ΔA apsolutnu pogrešku aproksimacije i $|\epsilon|$ je relativna pogreška aproksimacije.

U tablici su zapisani dobiveni rezultati I^*, apsolutne greške ΔA i relativne apsolutne greške $|\epsilon|$. Vidi se da se povećanjem broja uzrokovanja n smanjuje relativna apsolutna greška $|\epsilon|$ i dobiva se bolja aproksimacija.



Slika 5.1. Graf aproksimacije PID regulatora

Graf prikazuje odnos apsolutne pogreške aproksimacije i broja uzrokovanja. Na ordinati y nalazi se apsolutna pogreška aproksimacije, a na apscisi x, broj uzorkovanja.

Vidi se da apsolutna pogreška smanjuje s brojem uzorkovanja. Drugačije rečeno, s većim brojem uzrokovanja imamo točniju aproksimaciju.

Sve električne komponente ne odgovaraju svojoj nominalnoj vrijednost. Varijacije u materijalima i proizvodnji kao i radni uvjeti mogu utjecati na njihovu vrijednost. Najčešće se pretpostavlja se da je krug dizajniran tako da zahtijeva određenu vrijednost komponente. Pokažimo to na sljedećem primjeru.

Primjer 5.2. *Potrebno je odrediti sigurnost u vrijednost oscilatora. Za rješavanje ovog problema potrebna je funkcija gustoće vjerojatnosti integrirana za određivanje intervala povjerenja da bi oscilator imao svoju frekvenciju unutar 5% od $1kHz$. Tu vjerojatnost nalazimo u ukupnoj površini* *ispod normalne distribucije. Prema toj definiciji formula za vjerojatnost glasi:*

$$(1-\alpha) \approx \int_{-2.15}^{2.9} \frac{1}{\sqrt{\pi}} e^{-\frac{x^2}{2}}\, dx. \tag{5.17}$$

U izrazu α tražena je vrijednost koja izražava vjerojatnost sigurnosti, ostatak izraza je formula za vjerojatnost sigurnost zadanog oscilatora. Budući da integral nema eksplicitno rješenje, potrebno je koristi se numeričkom integracijom za rješavanje problema. Koristi se Simpsonovo pravilo i dobiva se izraz:

$$(1-\alpha) \approx \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)]. \tag{5.18}$$

Izraz predstavlja Simpsonovo pravilo za dani integral ako je korišten $n = 2$ broj uzrokovanja. Dalje u dobiveni izraz uvrštavamo parametre $a = -2.15$ i $b = 2.9$ te dobivamo:

$$(1-\alpha) \approx \frac{2.9+2.15}{6}[f(-2.15) + 4f(0.375) + f(2.9)], \tag{5.19}$$

Uvrštavaju se brojevi:

$$(1-\alpha) \approx \frac{5.06}{6}[0.39550 + 4(0.37186) + 0.0059525]. \tag{5.20}$$

Rješenje aproksimacije je:

$$(1-\alpha) \approx 1.2902. \tag{5.21}$$

Kako bi se provjerila pogreška, potrebno je eksplicitno rješenje, no nije ga moguće dobiti. Iz tog razloga koristimo Matlab i njegovim ćemo se rješenjem koristiti kao pravim.

$$(1-\alpha) = \int_{-2.15}^{2.9} \frac{1}{\sqrt{\pi}} e^{-\frac{x^2}{2}}\, dx, \tag{5.22}$$

$$(1-\alpha) = 0.98236. \tag{5.23}$$

Sada kada se zna aproksimacija i točno rješenje, može se naći pogreška aproksimacije ΔA. Ona će iznositi:

$$\Delta A = Tono\ rjeenje - Aproksimacija\ rjeenja, \tag{5.24}$$

$$\Delta A = 0.98236 - 1.2902, \tag{5.25}$$

Dobivena pogreška aproksimacije ΔA je:

$$\Delta A = -0.30785. \tag{5.26}$$

Nadalje možemo pronaći relativnu pogrešku aproksimacije $|\epsilon|$:

$$|\epsilon| = |\frac{Pogreska\ aproksimacije}{Tocno\ rjesenje}| \cdot 100\%, \tag{5.27}$$

$$|\epsilon| = |\frac{-0.30785}{0.98236}| \cdot 100\%, \tag{5.28}$$

Na kraju dobivamo da je relativna pogreška aproksimacije $|\epsilon|$:

$$|\epsilon| = 31.338\%. \tag{5.29}$$

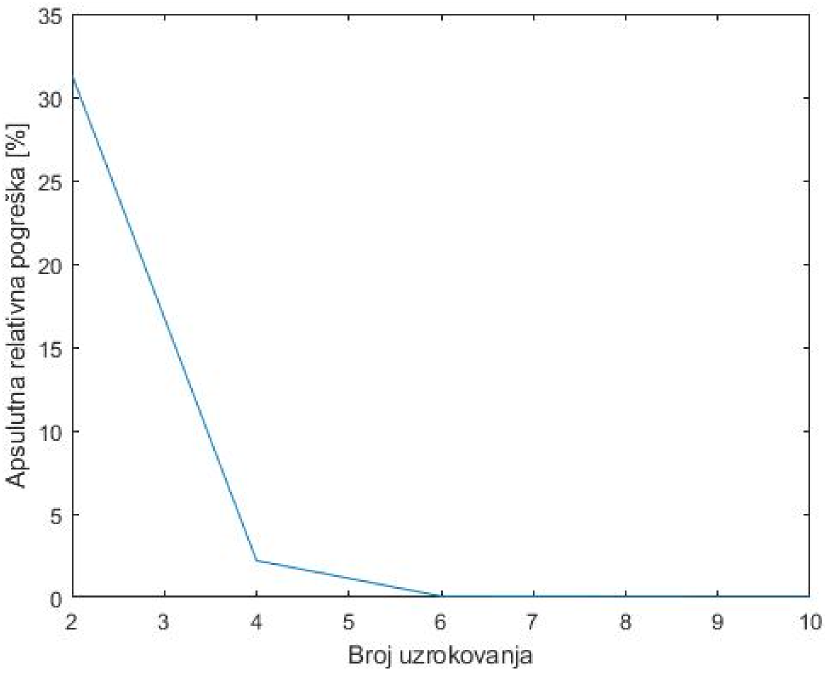
Ako bi se postupak ponovio za neko drugo uzrokovanje osim za $n = 2$, dobilo bi se novo rješenje. Nova bi dobivena rješenja bila uvijek točnija od ovoga jer je broj uzrokovanja $n = 2$ najmanja moguća vrijednost za koju je moguće primijeniti Simpsonovo pravilo. Broj uzrokovanja u Simpsonovu pravilu uvijek mora biti paran i znajući to stvaramo tablicu.

Tablica 5.2. Tablica aproksimacije sigurnosti oscilatora

n	I^*	ΔA	$\|\epsilon\|$
2	1.2902	0.30785	31.338%
4	0.96079	$2.1568 \cdot 10^{-2}$	2.1955%
6	0.98168	$6.8166 \cdot 10^{-4}$	$6.9391 \cdot 10^{-2}\%$
8	0.98212	$2.3561 \cdot 10^{-4}$	$2.3984 \cdot 10^{-2}\%$
10	0.982266	$9.2440 \cdot 10^{-5}$	$9.4101 \cdot 10^{-3}\%$

U tablicu n predstavlja broj uzrokovanja, I^* dobivenu aproksimaciju, ΔA apsolutnu pogrešku aproksimacije i $|\epsilon|$ je relativna pogreška aproksimacije.

Dobivena tablica opisuje odnos između broja uzrokovanja n, dobivenih aproksimacija I^*, pogrešaka aproksimacija ΔA i apsolutnih relativnih pogrešaka aproksimacija $|\epsilon|$. Kao što je napomenuto prije, s povećanjem uzrokovanja n dobiva se bolja aproksimacija čime se smanjuje pogreška aproksimacije ΔA te apsolutna relativna pogreška $|\epsilon|$.



Slika 5.2. Graf aproksimacije sigurnosti oscilatora

Na ordinati se nalazi apsolutna relativna pogreška $|\epsilon|$, *a na apcisi broj uzrokovanja* n.

Ovime grafom vizualizirana je prethodna teza da se s povećanjem uzrokovanja n *dobiva bolja aproksimacija te da se s povećanjem uzrokovanja* n *smanjuje apsolutna relativna pogreška* $|\epsilon|$.

6. Zaključak

Numerička integracija grana je numeričke matematike kojom se rješavaju matematički problemi koji se javljaju u praksi, a povezani su s rješavanjem određenog integrala. U radu se govorilo o dvije numeričke metode integriranja: trapeznoj i Simpsonovoj metodi. Obije metode proizlaze iz Newton-Cotesove formule. Trapeznom i Simpsonovom metodom ne dobivamo pravo rješenje već njegovu aproksimaciju.

U trapeznom pravilu aproksimiramo integral krivulje s trapezom. Ako povećamo broj trapeza kojim aproksimiramo krivulju, dobivamo opće trapezno pravilo. Broj trapeza za opće trapezno pravilo je proizvoljan. Povećanjem broja trapeza, kojima se aproksimira krivulja, povećava se točnost rezultata, a time i smanjuje pogreška. Povećanjem broja trapeza, kojima se aproksimira krivulja, povećava se kompleksnost zapisa.

Simpsonova formula numerička je metoda u kojoj se krivulja aproksimira s polinomom drugoga stupnja. Krivulja se predstavlja kroz tri čvora, odnosno dvije granice intervala aproksimacije integrala. Povećanjem broja točaka dobiva se opća Simpsonova formula. S većim brojem točaka dobiva se točnije rješenje i povećava se kompleksnost zapisa.

Razlika između trapezne i Simpsonove metode najuočljivija je u pogrešci kod aproksimacije. Za isti će broj točaka aproksimacije u istom intervalu Simpsonova formula imati manju pogrešku. Kod trapezne formule pogreška se smanjuje kvadratno s obzirom na broj točaka, dok se kod Simpsonove formule pogreške smanjuju s četvrtom potencijom u odnosu na broj točaka.

U računalstvu primjenjuje se većinom kombinacija Simpsonove i trapezne formule ili čista Simpsonova formula. Simpsonova formula ima bolju aproksimaciju za iste korištene resurse, no ako želimo ubrzati ili smanjiti kompleksnost, a zadržati preciznost, potrebno je Simpsonovu formulu kombinirati s trapeznom formulom.

U elektrotehnici obje se metode pronalaze u praksi. U radu su opisani primjeri vezani uz provjeru kvalitete proizvoda te u sustavima regulacije. U oba slučaja funkcije koje opisuju zadane probleme, mogu biti nerješive s algebarskim metodama. Iz tog razloga češće se koriste numeričke metode kako bi se došlo do rješenja na što brži i jednostavniji način.

Literatura

[1] Atkinson, K.E.: "Povjest numeričke analize", s Interneta, https://www.britannica.com/science/numerical-analysis/Historical-background, 01.04.2022.

[2] Scitovski, R.: "Numerička matematika drugo izdanje", s Interneta, https://www.mathos.unios.hr/pim/Materijali/Num.pdf, Osijek, 2004., 01.04.2022.

[3] Drmač, Z.; Hari, V.; Marušić, M.; i dr.: "Numerička Analiza", s Interneta, https://web.math.pmf.unizg.hr/~rogina/2001096/num_anal.pdf, Zagreb 2003., 01.04.2022.

[4] Došlić, T.: Numerička matematika, Gradevinski fakultet Sveučilište u Zagrebu", s Interneta, https://www.grad.unizg.hr/_download/repository/Doslic_T.%3B_Numericka_matematika.pdf, 01.04.2022.

[5] Walls, P.: "Mathematical Python", s Interneta, https://personal.math.ubc.ca/~pwalls/math-python/, 01.04.2022.

Sažetak i ključne riječi

U ovom radu opisane su metode za numeričko integriranje koje se baziraju na Newton-Cotesovim formulama, a to su trapezna i Simpsonova formula. Objašnjen je povijesni aspekt numeričke matematike, grane u koju spada numeričko integriranje. Analizirane su vrte numeričkih grešaka te njihovi uzroci. Trapezna i Simpsonova formula matematički su izvedene, te je demonstrirano njihovo korištenje s teorijskog i praktičnog aspekta. Pokazana je i njihova implementacija u programskom jeziku Python. Napravljeno je nekoliko numeričkih primjera te dva primjera povezana s elektrotehnikom.

Ključne riječi: Numeričko integriranje, Trapezno pravilo, Simpsonovo pravilo, Python

Summary and key words

This paper describes methods of numerical integration based on the Newton-Cotes formulas, namely the trapezoidal and Simpson formulas. The historical aspect of numerical mathematics, of which numerical integration is a part, is explained. The types of numerical errors and their causes were analyzed. The trapezoidal and Simpson formulas are mathematically derived and their application is demonstrated from a theoretical and practical point of view. Their implementation in the Python programming language is also shown. Several numerical examples and two examples from the field of electrical engineering are given.

Keywords: Numerical Integration, Trapezoidal Rule, Simpson Rule, Python Rule