1.

Uvod

Numerička matematika bavi se izvodom, analizom i implementacijom algoritama za numeričko rješavanje matematičkih problema koji se javljaju u prirodnim i društvenim znanostima te u inženjerstvu.

Numerički algoritmi stari su gotovo kao i ljudska civilizacija. Tako su, na primjer, već drevni Egipćani poznavali metode za rješavanje nekih jednostavnih jednadžbi o čemu svjedoči Rhindov papirus[1] pronađen u Luxoru, a koji je nastao približno 1650. godine pr. n. e. Spomenimo nadalje starogrčkog matematičara Arhimeda iz Siracuse (287–212 pr. n. e.) koji se, pored ostalog, bavio aproksimacijom realnih brojeva $\sqrt{3}$ i π, te računanjem površina likova i volumena tijela. U srednjem vijeku mnogi zanimljivi rezultati pripadaju perzijskim matematičarima i astronomima.

Veliki poticaj razvitku matematičkog modeliranja i numeričke matematike dali su Isaac Newton (1643–1727) i Gottfried Wilhelm Leibniz (1646–1716) razvojem diferencijalnog računa. Newton je idejni začetnik mnogih numeričkih metoda čije generalizacije nose danas njegovo ime (Newtonova metoda za pronalaženje nultočaka nelinearne jednadžbe, Newtonova interpolacijska formula itd.). Nadalje, mnogi matematičari 18. i 19. stoljeća dali su veliki doprinos razvitku numeričkih metoda za rješavanje matematičkih problema. Spomenimo ovdje samo neke od njih: Leonhard Euler (1707–1783), Joseph Louis Lagrange (1736–1813) i Carl Friedrich Gauss (1777–1855). Konačno, strelovit razvoj računala od polovice prošlog stoljeća, razvitak programskih jezika, računalne grafike, simboličkog računa, korisnički orijentiranih sučelja[2] omogućuju danas numeričko rješavanje i interpretaciju složenih matematičkih modela iz realnog svijeta.

[1] Alexander Henry Rhind (1833–1863).

[2] engl. *Graphical User Interfaces*.

U praksi obično polazimo od realnog problema čije rješenje želimo simulirati na računalu. Najprije formuliramo pripadni matematički model koji analiziramo primjenjujući teorijska znanja iz matematičke analize. Važno je da je problem matematički dobro postavljen (postoji rješenje koje je jedinstveno, te da je stabilan: male promjene početnih uvjeta vode malim promjenama rješenja). Nakon matematičke analize dolazi numeričko rješavanje problema koje započinje konstrukcijom metode, njezinom analizom i implementacijom. Završni korak sastoji se od pravilne interpretacije dobivenih rezultata te od analize pogrešaka.

Na sljedećem vrlo jednostavnom modelnom primjeru upoznat ćemo se s matematičkim modeliranjem realnog problema, njegovim numeričkim rješavanjem, te s pogreškama koje se pritom javljaju.

Primjer 1.1. (Modelni primjer)
Kosi toranj u Pisi visok je $H = 44.12$ metara i s njegova vrha bačen je kamen. Koliko vremena treba kamenu da udari u zemlju?

Slika 1.1. *Kosi toranj u Pisi*

Radi jednostavnosti, zanemarimo prisutno trenje zraka, djelovanje Coriolisove sile, te uzmimo da kamen ima jediničnu masu. Uočimo da smo time već prilikom postavljanja problema uveli određenu **pogrešku u modelu**. Pad kamena opisat ćemo matematički kao funkciju prijeđenog puta u ovisnosti o vremenu. Označimo sa $h(t)$ visinu kamena u trenutku t. Brzina kamena dana je kao derivacija funkcije h po

vremenu:

$$v(t) = \frac{dh(t)}{dt} = h'(t).$$

Nadalje, ubrzanje kamena definira se kao derivacija brzine po vremenu:

$$a(t) = v'(t) = \frac{dv(t)}{dt} = \frac{d^2h(t)}{dt^2}.$$

Za ubrzanje sile teže uzimamo približnu vrijednost $g = 9.81\frac{\mathrm{m}}{\mathrm{s}^2}$. Na ovom mjestu uvodimo **pogrešku u podacima**. Vrijedi

$$a(t) = -g,$$

odakle možemo izračunati brzinu kamena u trenutku t. Integriranjem dobivamo

$$\begin{aligned} v(t) &= \int a(t)dt = -gt + C_1, \\ h(t) &= \int v(t)dt = \int (-gt + C_1)dt = -\frac{g}{2}t^2 + C_1 t + C_2. \end{aligned}$$

Iz uvjeta $h(t) = 0$ dobivamo da će bačeni kamen nakon

$$t = \frac{C_1 \pm \sqrt{C_1^2 + 2gC_2}}{g} \tag{1.1}$$

sekundi udariti u tlo. Da bismo dobili eksplicitno rješenje, potrebno je odrediti vrijednosti konstanti C_1 i C_2. Konstantu C_1 dobivamo iz početnog uvjeta prema kojemu je kamen u trenutku $t = 0$ bio u stanju mirovanja, odnosno njegova brzina bila je jednaka nuli:

$$0 = v(0) = -g \cdot 0 + C_1 \quad \Longrightarrow \quad C_1 = 0.$$

Konstantu C_2 dobivamo iz uvjeta $h(0) = H$:

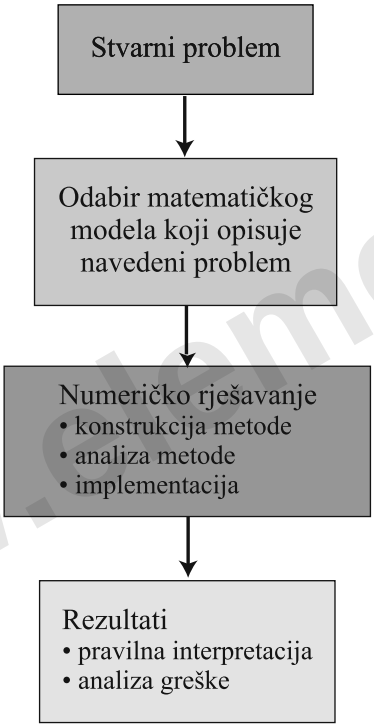
$$44.12 = h(0) = -\frac{g}{2} \cdot 0 + C_1 \cdot 0 + C_2 \quad \Longrightarrow \quad C_2 = 44.12.$$

Uvrstimo li dobivene vrijednosti u formulu (1.1), dobivamo da je rješenje problema slobodnog pada

$$t = \frac{\sqrt{44.12 \cdot 19.62}}{9.81} \text{ sekunde},$$

pri čemu smo zanemarili negativno rješenje koje nema fizikalni smisao. Džepni kalkulator daje vrijednost 2.9992 koju možemo zaokružiti na tri decimalna mjesta te dobivamo $t = 3.00$ sekunde. Primijetimo da je u rezultat ušla **pogreška kalkulatora** pri operaciji korjenovanja kao i **pogreška zaokruživanja**.

U konačan rezultat ušle su mnoge pogreške (modela, podataka, kalkulatora i zaokruživanja). Stoga se prirodno nameće pitanje je li dobiveni rezultat korektan te koliko odstupa od stvarnog rezultata. Točnost rezultata mogli bismo provjeriti tako da otputujemo u Pisu, popnemo se na vrh kosog tornja, bacimo kamen te izmjerimo vrijeme potrebno da kamen padne na tlo. Ovakva provjera rezultata svakako je moguća, no nije jako praktična. Općenito, direktna provjera rezultata u praksi je često ili preskupa ili uopće nije moguća. Stoga je u realnim problemima nužno imati dobar matematički model, kvalitetne numeričke metode za njegovo rješavanje te pouzdane ocjene pogreške. Postupak rješavanja nekog problema iz prakse shematski možemo prikazati slikom 1.2.



Slika 1.2. *Shematski prikaz rješavanja problema iz prakse*

1.1. Pogreške

Kao što smo već rekli na početku poglavlja, u ovoj knjizi upoznat ćemo se s raznim numeričkim metodama te ćemo analizirati dobivena rješenja. Na modelnom problemu (1.1) vidjeli smo da se prilikom numeričkog rješavanja javljaju različiti tipovi pogrešaka:

- pogreške modela
- pogreške u mjerenjima
- pogreške metode za numeričko rješavanje
- pogreške aritmetike računala

Mjera za pogrešku. Neka je $\hat{x}$ aproksimacija realnog broja x dobivena nekom numeričkom metodom. Pogrešku te aproksimacije označujemo sa $E(x,\hat{x})$ i definiramo formulom

$$E(x,\hat{x}) = x - \hat{x}.$$

Uočimo da ovako definirana pogreška može biti i negativna. Uobičajena mjera za točnost ili pogrešku aproksimacije realnog broja x približnom vrijednošću $\hat{x}$ jest **apsolutna pogreška**

$$E_{\text{abs}} = |E(x,\hat{x})| = |x - \hat{x}|.$$

Primjer 1.2. Neka je $x = 2.2$ egzaktna vrijednost neke veličine, dok odabrana numerička metoda daje približnu vrijednost $\hat{x} = 2.20345$. Tada apsolutna pogreška iznosi

$$|x - \hat{x}| = 0.00345 = 3.45 \cdot 10^{-3}.$$

Ovdje je apsolutna pogreška reda veličine 10^{-3}, a rješenje je točno do na treću decimalu. Bili bismo sretniji s metodom koja bi davala pogrešku reda veličine 10^{-6}, dok bi nas, s druge strane, pogreška reda veličine 10^{6} užasnula. Uočimo da je u fizikalnim primjerima jako važno u kojim mjernim jedinicama računamo pogrešku. Uzmimo da su gore navedeno egzaktno rješenje x, kao i rješenje $\hat{x}$ dobiveno numeričkom metodom izraženi u metrima. Ako pak rješenja, kao i apsolutnu pogrešku izrazimo u nanometrima, dobivamo da je

$$|x - \hat{x}| = 3.45 \cdot 10^{6}\,\text{nm}.$$

Dobili smo ogromnu apsolutnu pogrešku, dok rješenje, kao i prije, ima tri točne znamenke. S druge strane, mjerimo li rezultate u kilometrima, imamo

$$|x - \hat{x}| = 3.45 \cdot 10^{-6}\,\text{km}.$$

U ovom slučaju pogreška djeluje jako mala, no i dalje imamo rješenje s tri točne znamenke.

U gornjem primjeru problemi su nastali zbog lošeg skaliranja. Jedan od načina da se poteškoće izbjegnu je promatranje **relativne pogreške** koja je za $x \neq 0$ definirana na sljedeći način:

$$E_{\text{rel}} = \frac{E_{\text{abs}}}{|x|} = \frac{|x - \hat{x}|}{|x|}.$$

Vidimo da je ovdje veličina pogreške skalirana obzirom na veličinu vrijednosti koja se računa. Relativna pogreška za podatak iz primjera 1.2 iznosi

$$\frac{|2.2 - 2.20345|}{|2.2|} = 1.57 \cdot 10^{-3}$$

i ne ovisi o tome u kojim je mjernim jedinicama podatak zadan.

Pogreške modela nastaju obično zanemarivanjem utjecaja određenih sila poput otpora zraka ili trenja, odnosno raznim pojednostavnjenjima realnih situacija. Promotrimo kako nastaju pogreške modela na jednom realnom primjeru iz područja matematičkog modeliranja poluvodiča.

Primjer 1.3. (Matematičko modeliranje poluvodiča)
Svakodnevni život danas je nezamisliv bez poluvodičkih uređaja. Široku primjenu poluvodičkih uređaja od 50-ih godina prošlog stoljeća do danas omogućilo je rapidno smanjivanje njihove dimenzije:

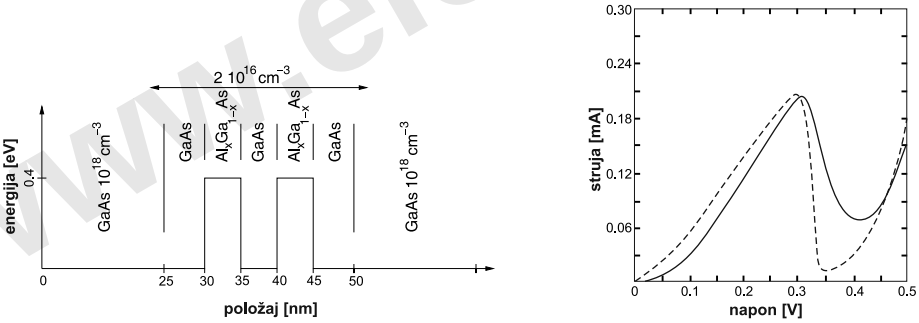
Godina	1971	1982	1993	2003	2004	2006	2010	2012
Dimenzija [nm]	10 000	1500	350	130	90	70	45	22

Podaci prikazani u gornjoj tablici preuzeti su s *International Roadmap for Semiconductors* (`http://www.itrs.net/`). Prvi Intelov procesor 4004 proizveden 1971. sastojao se od 2250 tranzistora od kojih je svaki imao karakterističnu duljinu 10 μm. Dimenzija tranzistora u Pentiumu 4 koji se na tržištu pojavio 2004. smanjena je na 90 nm, dok su moderni poluvodički uređaji svojom veličinom dosegli nanoskalu. Na slici 1.3 prikazana je fotografija[3] prvog tranzistora koji su u Bellovim laboratorijima konstruirali Bardeen, Brittain i Shockley 1947. godine. Spomenimo ovdje da su za svoja istraživanja na području fizike poluvodiča nagrađeni 1956. godine Nobelovom nagradom za fiziku.

[3]Fotografija je preuzeta s `http://www.porticus.org/bell/belllabs_transistor.html`.

Slika 1.3. *Prvi tranzistor konstruiran 1947.*

Matematičko modeliranje poluvodiča je multidisciplinarno područje kojim se bave inženjeri, fizičari i matematičari. U svrhu reduciranja troškova proizvodnje poluvodičkih uređaja, potrebno je izvesti odgovarajuće matematičke modele koji će se zatim upotrebljavati u numeričkim simulacijama. Zbog složenosti realne situacije prilikom modeliranja nužna su pojednostavnjenja. Poluvodički uređaj obično se nalazi u nekom električnom krugu. Da bi se proizvela struja kroz kontakt, na poluvodički uređaj primjenjuje se odgovarajući napon. Odnos između primijenjenog napona i dobivene struje opisuje strujno-naponska karakteristika (CV, engl. *Current-Voltage characteristics*). Na slici 1.4 vidimo pojednostavnjenu skicu rezonantno-tunelirajuće diode (RTD, engl. *Resonant-Tunneling Diode*) [22] te njezinu eksperimentalnu strujno-naponsku karakteristiku [24].



Slika 1.4. *RTD (lijevo). Eksperimentalna CV-karakteristika (desno)*

Rezonantno-tunelirajuća dioda sadržava dvije potencijalne barijere između kojih se nalazi tzv. kvantna jama (engl. *quantum well*). Zbog malog promjera potencijalnih barijera (između 5 i 10 nm) elektroni mogu tunelirati kroz njih, a dobiveni kvantni efekt može se uočiti na eksperimentalnoj strujno-naponskoj karakteristici. Preciznije, uočimo da dobivena struja nije monotono rastuća funkcija primijenjenog napona. Strujno-naponska karakteristika prikazana na slici 1.4 (desno) ima dio u kojem dobivena struja pada iako se napon povećava, što nije u skladu s Ohmovim zakonom. Dobiveni efekt je kvantne prirode i naziva se NDR-efekt (engl. *Negative Differential Resistence Effect*), a nastaje zbog tuneliranja elektrona kroz potencijalne barijere RTD-diode.

Iz osnovnih fizikalnih principa željeli bismo izvesti matematički model (sustav parcijalnih diferencijalnih jednadžbi) koji bi dovoljno dobro opisivao realnu situaciju. Pitanja koja se pritom prirodno nameću su:

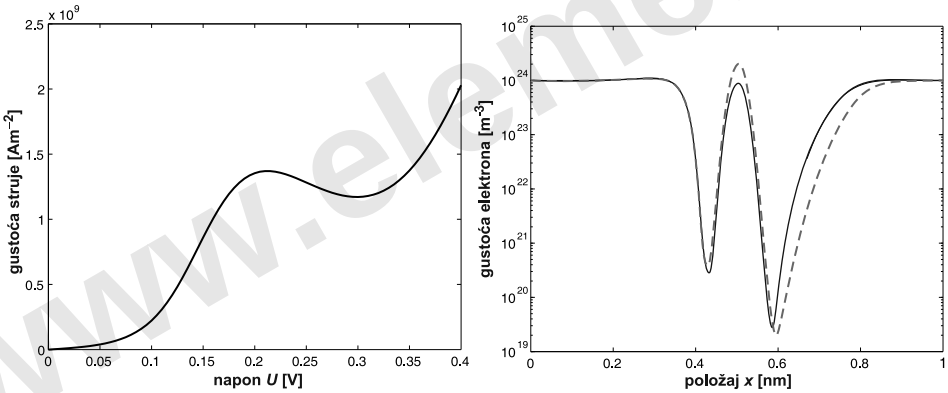
- Kako modelirati gibanje mnoštva čestica? Promatramo li gibanje elektrona i šupljina kao fluidno gibanje (makroskopski) ili kao gibanje rijetkog plina (mikroskopski)?
- Na koji način uključiti geometriju uređaja (utjecaj različitih efektivnih masa, potencijalne barijere, doping)?
- Kako kristalna rešetka utječe na gibanje elektrona?
- Kako modelirati međusobne kolizije elektrona te kolizije između elektrona i kristalne rešetke?
- Koji su fizikalno opravdani rubni uvjeti?
- Na koji način opisati kvantne efekte?

Jedan od modela kojim se može dobiti strujno-naponska karakteristika RTD je (skalirani) kvantni drift-difuzijski model (QDD, engl. *Quantum Drift-Diffusion Model*):

$$\partial_t n - \operatorname{div} J = 0, \quad J = -\frac{\varepsilon^2}{6} n \nabla \left(\frac{\Delta \sqrt{n}}{\sqrt{n}} \right) + T \nabla n - n \nabla V, \quad x \in \Omega \subset \mathbb{R}^3,\ t > 0,$$

pri čemu je n gustoća elektrona, J gustoća struje, T temperatura kristalne rešetke, V potencijal, ε skalirana Planckova konstanta. Na slici 1.5 vidimo rezultate simulacija koje su dobivene numeričkim rješavanjem QDD-modela. Možemo uočiti da strujno-naponska karakteristika pokazuje NDR-efekt. S druge strane, na slici 1.5 (desno) prikazana je gustoća elektrona izračunana u prvom lokalnom maksimumu te sljedećem lokalnom minimumu strujno-naponske karakteristike. Primijetimo

da gustoća elektrona drastično oscilira. Naime, zbog tuneliranja elektroni se ne zadržavaju unutar potencijalnih barijera gdje n doseže vrijednosti reda veličine 10^{19} m^{-3}. U području između barijera gustoća elektrona znatno raste i iznosi približno 10^{24} m^{-3}.



Slika 1.5. *CV-karakteristika RTD (lijevo). Gustoća elektrona n (desno)*

Napomenimo ovdje da je gornja strujno-naponska karakteristika RTD dobivena primjenom Scharfetter–Gummelove metode. Spomenuta metoda primjenjuje se u numeričkom rješavanju problema koji sadržavaju konvekciju i difuziju. Više o metodi može se pogledati npr. u [28, 32]. Jedan od glavnih nedostataka QDD-modela je pretpostavka da je temperatura u poluvodiču konstantna, što ne odgovara realnoj situaciji. Unatoč spomenutoj nepreciznosti modela koja je posljedica pojednostavnjivanja realne situacije, prema [28] QDD-model se još od 1994. primjenjuje u industrijskim simulacijama.

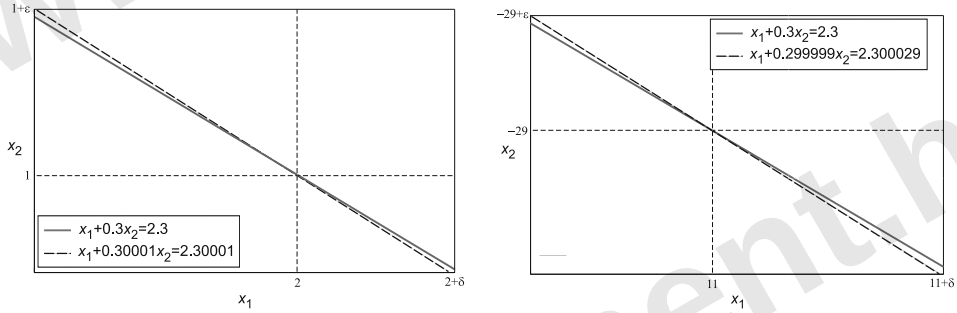
Pogreške u mjerenjima, odnosno pogreške u ulaznim podacima javljaju se uglavnom zbog nemogućnosti točnih mjerenja. Ovdje je važno naglasiti da ako je problem nestabilan, tada male pogreške u ulaznim podacima mogu dovesti do velike pogreške u rezultatu. Navedimo dva jednostavna primjera.

Primjer 1.4. Zadana su dva linearna sustava

$$\begin{aligned} x_1 + 0.3x_2 &= 2.3 \\ x_1 + 0.30001x_2 &= 2.30001, \end{aligned} \tag{1.2a}$$

$$\begin{aligned} x_1 + 0.3x_2 &= 2.3 \\ x_1 + 0.299999x_2 &= 2.300029. \end{aligned} \tag{1.2b}$$

Primijetimo da drugu jednadžbu sustava (1.2b) dobivamo malom perturbacijom koeficijenata druge jednadžbe sustava (1.2a). Uzmimo da su koeficijenti prvog sustava egzaktni te da je perturbacija u koeficijentima nastala zbog pogreške u mjerenju. Lako se vidi da rješenje sustava (1.2a) iznosi $x_1 = 2$, $x_2 = 1$. Kako je perturbacija koeficijenata reda veličine 10^{-5}, očekujemo da će rješenje sustava (1.2b) biti *blisko* rješenju sustava (1.2a). No, rješenje sustava (1.2b) iznosi $x_1 = 11$, $x_2 = -29$. Dakle, *mala* promjena koeficijenata dovela je do *velike* promjene rješenja. Za takve probleme kažemo da su **loše uvjetovani**. Radi boljeg razumijevanja dobivenih rezultata pogledajmo grafičku interpretaciju zadanih linearnih sustava. Svaka jednadžba sustava (1.2a), odnosno (1.2b) predstavlja jedan pravac u $\mathbb{R}^2$. Riješiti navedene sustave znači naći sjecište pripadnih pravaca u $\mathbb{R}^2$. Slika 1.6 grafički prikazuje rješenje promatranih sustava. Vidimo da je riječ o gotovo identičnim pravcima. Dakle, nije neočekivano da mala perturbacija koeficijenata znatno pomiče njihovo sjecište.



Slika 1.6. *Približni grafički prikaz rješenja sustava* (1.2a) *(lijevo) i* (1.2b) *(desno), pri čemu su ε i δ malene vrijednosti*

Primjer 1.5. Promotrimo polinom

$$\begin{aligned} p(x) &= (x-1)(x-2)(x-3)(x-4)(x-5)(x-6)(x-7) \\ &= x^7 - 28x^6 + 322x^5 - 1960x^4 + 6769x^3 - 13132x^2 + 13068x - 5040. \end{aligned}$$

Korijeni polinoma $p(x)$ očito su realni i iznose $1,2,3,4,5,6,7$. Odaberimo sada koeficijent 28 polinoma $p(x)$ koji se nalazi uz član x^6 te ga promijenimo za 0.001. Dobivamo polinom

$$p_1(x) = x^7 - 28.001x^6 + 322x^5 - 1960x^4 + 6769x^3 - 13132x^2 + 13068x - 5040.$$

Zanima nas kakve posljedice na nultočke ima učinjena, naizgled beznačajna, promjena koeficijenta. Za računanje nultočaka polinoma $p_1(x)$ koristimo MATLAB, odnosno naredbu `roots`:

```
>> koef = [1 -28.001 322 -1960 6769 -13132 13068 -5040];
>> roots(koef)
ans =
   7.1336
   5.4757 + 0.2314i
   5.4757 - 0.2314i
   3.9008
   3.0158
   1.9995
   1.0000
```

Uočimo da je jedino vrijednost 1 nultočka oba polinoma p i p_1. Nadalje, vrijednosti $2,3,4$ i 7 su se promijenile za najviše za 0.1336, dok su preostala dva korijena polinoma p_1 kompleksno-konjugirana. Vidimo da je relativno mala promjena jednog koeficijenta polaznog polinoma dovela do bitne promjene vrijednosti pripadnih nultočaka. U tom smislu možemo reći da je problem pronalaženja nultočaka zadanog polinoma $p(x)$ nestabilan, obzirom na njegove koeficijente.

Pogreške metoda za numeričko rješavanje problema često nastaju kad se beskonačni procesi moraju zamijeniti konačnima. To uključuje sve postupke koji se definiraju pomoću limesa, poput deriviranja i integriranja, te postupke u kojima krajnje rješenje dobivamo konvergencijom niza približnih rješenja. Takve pogreške uobičajeno dijelimo na:

- **pogreške diskretizacije** (engl. *discretization errors*)
- **pogreške odbacivanja** (engl. *truncation errors*)
- **pogreške zaokruživanja** (engl. *rounding errors*)

Primjer 1.6. (Konačne diferencije)
Pogreške diskretizacije javljaju se u formulama konačnih diferencija. Aproksimirajmo vrijednosti prve i druge derivacije realne proizvoljno glatke funkcije f u danoj točki. U tu svrhu promotrimo Taylorov razvoj funkcije f oko točke x:

$$f(x+h) = f(x) + \frac{f'(x)}{1!}h + \frac{f''(x)}{2!}h^2 + \frac{f'''(x)}{3!}h^3 + \dots \tag{1.3a}$$

$$f(x-h) = f(x) - \frac{f'(x)}{1!}h + \frac{f''(x)}{2!}h^2 - \frac{f'''(x)}{3!}h^3 + \dots \tag{1.3b}$$

Izrazimo li $f'(x)$ iz razvoja (1.3a), imamo

$$\begin{aligned} f'(x) &= \frac{f(x+h)-f(x)}{h} - \frac{f''(x)}{2!}h - \frac{f'''(x)}{3!}h^2 - \cdots \\ &= \frac{f(x+h)-f(x)}{h} + \mathrm{O}(h). \end{aligned}$$

Čitaoce koji nisu upoznati sa značenjem simbola O upućujemo na poglavlje *Dodatak*.

Na taj način dobili smo formulu za aproksimaciju derivacije funkcije f u točki x:

$$f'(x) \approx \frac{f(x+h)-f(x)}{h}.$$

Dobivena formula naziva se **formula konačne diferencije unaprijed**. Kako je formula dobivena odbacivanjem članova reda h, kažemo da je prvog reda točnosti. Slično, izrazimo li $f'(x)$ iz razvoja (1.3b), imamo

$$\begin{aligned} f'(x) &= \frac{f(x)-f(x-h)}{h} + \frac{f''(x)}{2!}h - \frac{f'''(x)}{3!}h^2 + \cdots \\ &= \frac{f(x)-f(x-h)}{h} + \mathrm{O}(h). \end{aligned}$$

Na taj način dobili smo još jednu formulu za aproksimaciju derivacije funkcije f u točki x koja glasi

$$f'(x) \approx \frac{f(x)-f(x-h)}{h}.$$

Ova formula naziva se **formula konačne diferencije unatrag** i također je prvog reda točnosti. Konačno, oduzmemo li razvoje (1.3a) i (1.3b), imamo

$$\begin{aligned} f'(x) &= \frac{f(x+h)-f(x-h)}{2h} - \frac{f'''(x)}{3!}h^2 - \frac{f^{(v)}(x)}{5!}h^4 - \ldots \\ &= \frac{f(x+h)-f(x-h)}{2h} + \mathrm{O}(h^2), \end{aligned}$$

čime dobivamo formulu **centralne diferencije**

$$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$$

za aproksimaciju derivacije funkcije f u točki x koja je drugog reda točnosti. Nadalje, drugu derivaciju funkcije f u točki x možemo aproksimirati formulom drugog reda točnosti koja glasi

$$f''(x) \approx \frac{f(x+h)-2f(x)+f(x-h)}{h^2}.$$

Naime, zbrajanjem razvoja danih u (1.3) pokazuje se da vrijedi

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathrm{O}(h^2).$$

Pogreške odbacivanja nastaju kada *rezanjem* beskonačnog niza ili reda na konačan broj članova odbacujemo ostatak. Pritom nam je posebno važno da možemo, na neki način, kontrolirati pogrešku. Sljedeći primjer ilustrira opisanu situaciju. Napomenimo da se u njemu spominju pojmovi poput strojnog epsilona i fp-aritmetike (engl. *floating point arithmetics*) za koje pretpostavljamo da su čitaocima poznati otprije, a kojima se sustavnije bavimo u potpoglavlju 1.2.

Primjer 1.7. Vrijednost funkcije $\sin x$ u zadanoj točki treba aproksimirati Taylorovim polinomom određenog stupnja tako da pogreška aproksimacije bude jednaka zadanoj toleranciji $\varepsilon > 0$ ili manja od nje.

Podsjetimo se: Taylorov razvoj dovoljno glatke funkcije f u okolini nule možemo zapisati kao sumu Taylorova polinoma stupnja n i pripadnog ostatka:

$$f(x) = T_n(x) + R_{n+1}(x),$$

pri čemu je

$$T_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(0)}{k!} x^k, \quad R_{n+1}(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} x^{n+1},$$

za neki ξ_x realni broj između 0 i x. Razvoj funkcije $\sin x$ u Taylorov red oko nule glasi

$$\sin x = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}. \tag{1.4}$$

Znamo da Taylorov red (1.4) konvergira za proizvoljno $x \in \mathbb{R}$. Odaberimo $n \in \mathbb{N}$ i aproksimirajmo funkciju $\sin x$ odgovarajućim Taylorovim polinomom

$$T_{2n+1}(x) = \sum_{k=0}^{n} \frac{(-1)^k x^{2k+1}}{(2k+1)!},$$

a pripadni ostatak označimo sa $R_{2n+3}(x)$. Pripadnu pogrešku aproksimacije definiramo standardno izrazom

$$|E(x)| = |\sin x - T_{2n+1}(x)| = |R_{2n+3}(x)|.$$

Pogledajmo sada koliko članova Taylorova reda trebamo uzeti da bismo postigli da vrijedi

$$|E(x)| \leq \varepsilon,$$

gdje je $\varepsilon > 0$ zadana tolerancija. Radi jednostavnosti uzet ćemo $x > 0$. Očito je da vrijedi

$$|R_{2n+3}(x)| \leq \frac{x^{2n+3}}{(2n+3)!}, \tag{1.5}$$

pri čemu smo iskoristili činjenicu da formula za n-tu derivaciju funkcije $\sin x$ glasi

$$(\sin x)^{(n)} = \sin\left(x + \frac{n\pi}{2}\right), \quad \forall n \in \mathbb{N},$$

te da za svaki $x \in \mathbb{R}$ vrijedi da je $|\sin x| \leq 1$. Iz izraza (1.5) dobivamo da ako za aproksimaciju funkcije $\sin x$ uzmemo onoliko članova pripadnog Taylorova reda (1.4), tako da apsolutna vrijednost prvog odbačenog člana padne ispod zadane tolerancije ε, napravit ćemo aproksimaciju čija pogreška $|E(x)|$ je jednaka ili manja od ε.

Promotrimo funkciju `AprSin(x)` koja računa vrijednosti $\sin x$ pomoću Taylorova razvoja.

```
function s = AprSin(x)
% AprSin Taylorov red za sin
% AprSin(x) racuna aproksimativno sin(x)
s = 0;
t = x;
n = 1;
while s+t ~= s
s = s+t;
t = -x^2/((n+1)*(n+2))*t;
n = n+2;
end
```

Uočimo da `AprSin(x)` uzima u obzir članove Taylorova reda funkcije $\sin x$ sve dok apsolutna vrijednost prvog sljedećeg člana u razvoju ne padne ispod vrijednosti strojnog epsilona. No, ipak zbog konačne fp-aritmetike i pogrešaka zaokruživanja, funkcija `AprSin(x)` ne daje zadovoljavajuće rezultate za proizvoljne vrijednosti varijable x. Uvjerimo se u to tako da pomoću funkcije `AprSin(x)` aproksimiramo vrijednosti od $\sin x$ u točkama $\pi/2, 11\pi/2, 21\pi/2, 31\pi/2$. Ispišimo koliko je članova reda bilo uzeto u obzir prilikom dobivanja rezultata, te odredimo po apsolutnoj vrijednosti najveći član reda uzet u obzir prilikom računanja. Dobivamo sljedeće rezultate:

- `AprSin(pi/2) = 1.000000000000000`
  12 članova, max. član po aps. vrijednosti `1.570796326794897`

- `AprSin(11*pi/2) = -1.000000000212873`
  38 članova, max. član po aps. vrijednosti `3.066514637383812e+06`

- `AprSin(21*pi/2) = 0.999866764041849`
  61 član, max. član po aps. vrijednosti `1.467259672825497e+13`

- `AprSin(31*pi/2) = -5.822018527024010e+03`
  79 članova, max. član po aps. vrijednosti `7.988994169819993e+19`

Kako je $\sin(31\pi/2) = -1$, vidimo da rezultat naredbe `AprSin(31pi/2)` daje potpuno pogrešan rezultat! Poznato je da Taylorov razvoj oko neke točke najbolje aproksimira danu funkciju u okolini te točke. Želimo li korištenjem razvoja (1.4) izračunati vrijednost funkcije $\sin x$ dalje od nule, potrebno je uzeti veći broj članova reda. Zaista, lako se vidi da se za računanje vrijednosti `AprSin(31pi/2)` uzima Taylorov polinom stupnja 157. No, zbog konačne fp-aritmetike i pogrešaka zaokruživanja aproksimacija polinomom tako visokog stupnja u ovom slučaju nije zadovoljavajuća. Naime, ovdje dolazi do zbrajanja (oduzimanja) brojeva različitog reda veličine što rezultira gubitkom točnosti rješenja.

Ipak, funkciju `AprSin(x)` možemo primjenjivati u aproksimaciji vrijednosti $\sin(31\pi/2)$ ako uzmemo u obzir činjenicu da je $\sin(31\pi/2) = \sin(-\pi/2)$. Općenito, kod periodičnih funkcija prirodno je iskoristiti njihovu periodičnost kako bi se izbjegli veliki argumenti, odnosno kako bi se računanje svelo na interval u kojem je algoritam poput ovog iz funkcije `AprSin(x)` dovoljno efikasan.

Da bismo mogli podrobnije govoriti o **pogreškama aritmetike računala**, podsjetimo se najprije osnovnih pojmova povezanih s reprezentacijom realnih brojeva u računalu.

1.2. Prikaz realnih brojeva u računalu

Realni brojevi implementiraju se u računalu u unaprijed zadanom formatu koji propisuje koliko se binarnih znamenaka (bitova) upotrebljava za prikaz broja, kako se one interpretiraju te na koji se način s njima računa. Dugo je postojala potreba za definiranjem standarda koji bi bio neovisan o pojedinom računalu. Godine 1985. IEEE Standards Board i American National Standards Institute usvojili su ANSI/IEEE Standard 754–1985 za binarnu aritmetiku pomičnog zareza (engl. *binary floating-point arithmetic*).