Univerza *v Ljubljani*
Fakulteta za *matematiko in fiziko*

Naloga 1: Izračun Airyjevih funkcij

Matematično-fizikalni praktikum

Dimitrije Pešić
Vpisna številka: 28201072

Predavatelj: prof. dr. Borut Paul Krševan

Ljubljana, 10.10.2023

1 Uvod

Naloga je od nas zahtevala, da z uporabo kombinacije Maclaurinove vrste in asimptotskega razvoja poiskati čim učinkovitejši postopek za izračun vrednosti Airyjevih funkcij $Ai(x)$ in $Bi(x)$ na vsej realni osi z absolutno napako, manjšo od 10^{-10}. Enako smo želeli narediti tudi z relativno napako in ugotoviti ali je to sploh možno.

Dodatno smo imeli nalogo poiskati prvi 100 ničel Airyjevih funkcij $Ai(x)$ in $Bi(x)$.

2 Numerično računanje

Airyjevi funkciji Ai in Bi se v fiziki pojavljata predvsem v optiki in kvantni mehaniki ter sta definirani kot neodvisni rešitvi enačbe

$$y''(x) - xy(x) = 0. \tag{1}$$

S pomočjo Maclaurinove vrste za vrednosti blizu $x \sim 0$ in Asimptotske vrste za velike x, lahko napravimo zlepek teh dveh približkov ter še zmeraj dosežemo željeno natančnost.

Da bi se izoginil računanju gamma funkcije/ fakultete za vsak člen, sem vrste prepisal v rekurzivno obliko s pomočjo dveh identitet (2, 3) za gamma Γ funkcijo:

$$\frac{\Gamma(z+n)}{\Gamma(z)} = z(z+1)\cdot\ldots\cdot(z+n-1) \tag{2}$$

$$\Gamma(s+\frac{1}{2}) = \frac{\sqrt{\pi}(2s-1)!!}{2^s} \tag{3}$$

Tako dobimo formule za Maclaurinov razvoj Ai in Bi okoli $x \approx 0$:

$$\begin{aligned} Ai(x) &= \alpha f(x) - \beta g(x), \\ Bi(x) &= \sqrt{3}\left[\alpha f(x) + \beta g(x)\right], \end{aligned} \tag{4}$$

kjer sta α in β konstanti, $f(x)$ in $g(x)$ pa:

$$\begin{aligned} f(x) &= \sum_{k=0}^{\infty} \left(\frac{1}{3}\right)_k \frac{3^k x^{3k}}{(3k)!} = \sum_{k=0}^{\infty} \left(\frac{1}{3}\right)_k f_k(x) \\ g(x) &= \sum_{k=0}^{\infty} \left(\frac{2}{3}\right)_k \frac{3^k x^{3k+1}}{(3k+1)!} = \sum_{k=0}^{\infty} \left(\frac{2}{3}\right)_k g_k(x), \end{aligned} \tag{5}$$

le te, sem pa razvil v rekurzivne formule:

$$\begin{aligned} (z)_n &= \frac{\Gamma(z+n)}{\Gamma(z)} = (z+n-1)\frac{\Gamma(z+n-1)}{\Gamma(z)} = (z+n-1)(z)_{n-1}, \\ f_k(x) &= \frac{3^k x^{3k}}{(3k)!} = \frac{3x^3}{3k(3k-1)(3k-2)}\frac{3^{k-1}x^{3(k-1)}}{(3(k-1))!} = \frac{3x^3}{3k(3k-1)(3k-2)} f_{k-1}(x), \\ g_k(x) &= \frac{3^k x^{3k+1}}{(3k+1)!} = \frac{x^3}{(3k+1)k(3k-1)}\frac{3^{k-1}x^{3(k-1)}}{(3(k-1))!} = \frac{x^3}{(3k+1)k(3k-1)} g_{k-1}(x). \end{aligned} \tag{6}$$

Člene Maclaurinove vrste sem sešteval dokler ni bila razlika sosednjih členov v vrsti, po absolutni vrednosti manjša od željene natačnosti $\varepsilon = 10^{-11}$, ali pa, da je število členov preseglo določeno mejo.

S tem sem izognil potencialnim problemom napak prelivanja in predolgem času izvajanja na račun slabše natančnosti.

Za asimptotsko vrsto ni bilo potrebno celotne vrste izraziti rekurzivno saj smo že vnaprej vedeli da ne bomo seštevali veliko število členov, saj bi le to divergiralo stran od željenega rezultata. Tako formule ostanejo:

$$L(z) \sim \sum_{s=0}^{\infty} \frac{u_s}{z^s}, \quad P(z) \sim \sum_{s=0}^{\infty} (-1)^s \frac{u_{2s}}{z^{2s}}, \quad Q(z) \sim \sum_{s=0}^{\infty} (-1)^s \frac{u_{2s+1}}{z^{2s+1}}, \tag{7}$$

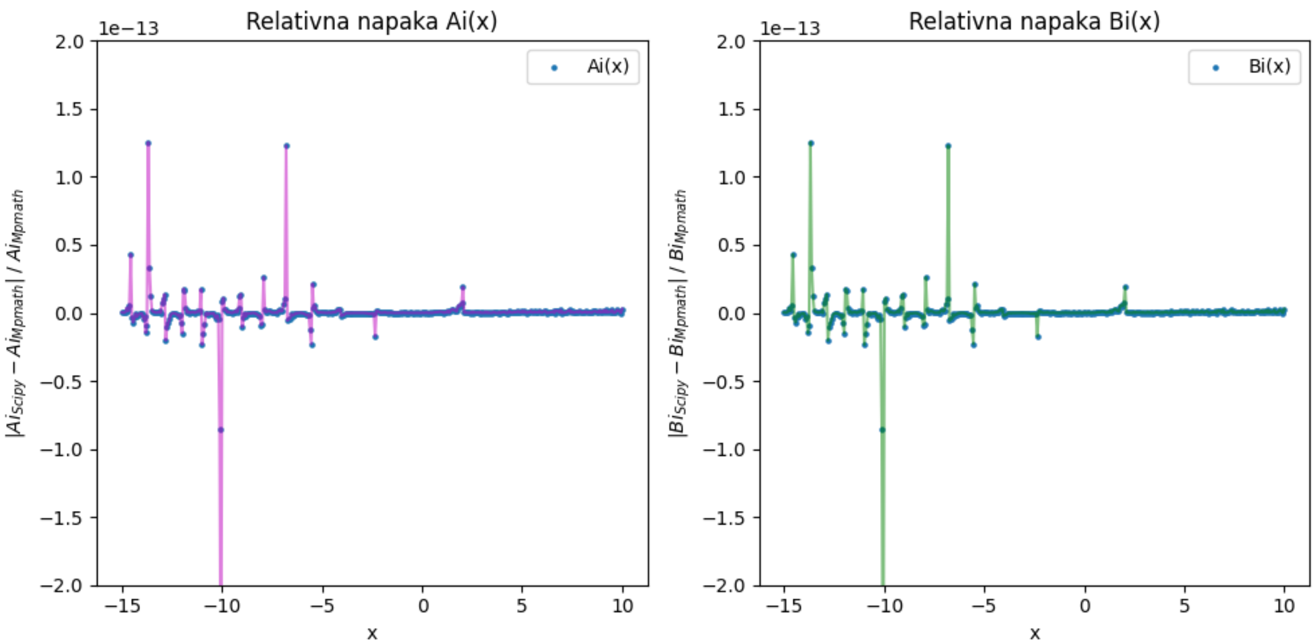
kjer sem pa člen u_s izrazil kot

$$u_s = \frac{\Gamma(3s + \frac{1}{2}}{54^s s! \Gamma s + \frac{1}{2}} = \frac{(3s - \frac{1}{2})(3s - \frac{3}{2})(3s - \frac{5}{2})}{54s(s - \frac{1}{2})} u_{s-1}; \quad u_0 = 1 \tag{8}$$

3 Preverjanje vgrajenih funkcij

Pri nalogi sem za referenco uporabljal vgrajeno implementacijo $scipy.special.airy(x)$ iz paketa *Scipy*. Čeprav so to implementirali mnogo pametnejši ljudje od mene, se je še zmeraj potrebno zavedati, da tudi referenčna funkcija ni popolnoma natančna ampak vendarle nek približek. Uporabil sem še neodvisno implementacijo Airyjevih funkcij iz paketa *Mpmath* in s tem primerjal natančnost obeh.

Natančnost bi lahko stestirali tudi z uporabo numerične integracije z veliko natančnostjo s paketom Decimal, ali pa morda s C++ knjižnico ALGLIB.



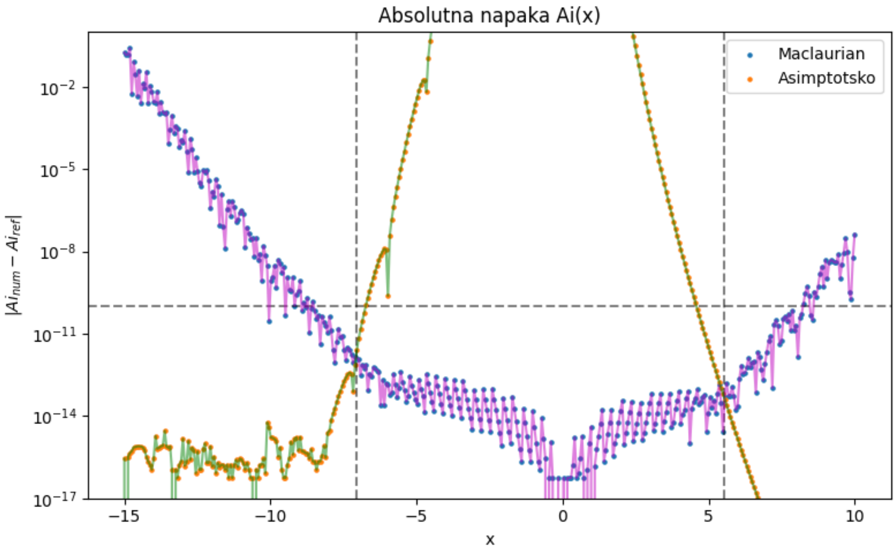
Slika 1: Graf relativne napake vgrajenih funkcij Ai in Bi

Iz grafa je razvidno, da je relativna napaka pod 10^{-13} kar je za naše potrebe dovolj natančno.

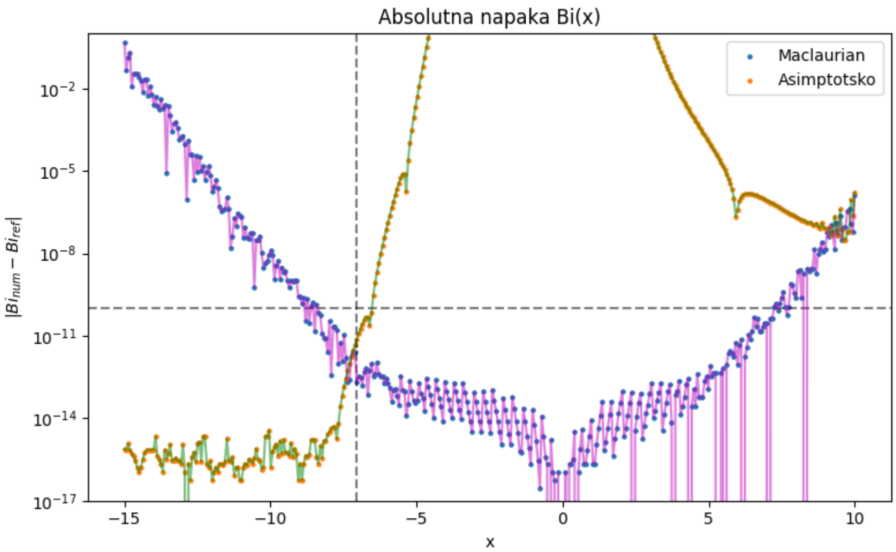
4 Rezultati

Program sem napisal v PYTHON jeziku in uporabil pakete *Numpy*, *Matplotlib* ter *Scipy*. Za konstanti $\alpha = Ai(0) = Bi(0)/\sqrt{3}$ in $\beta = -Ai'(0) = Bi'(0)/\sqrt{3}$ sem namesto vrednosti iz navodil uporabil vrednosti iz paketa *Scipy*.

Funkcije iz prejšnjega poglavja sem implementiral in poračunal vrednosti absolutnih napak za obe implementaciji, Maclaurin in asimptotsko. Rezultata tega sta prikazana na grafih 2 in 3. S tem smo lahko določili optimalno točko na kateri moramo zamenjati iz asimptotskega razvoja v Maclaurinov, da bi obdržali željeno natančnost. Opazimo lahko, da se pri absolutni napaki funkcije $Bi(x)$ napaka hitro veča, ker členi hitro naraščajo. S tem Python programu zmanjka prostora pri zapisu števila saj je omejen na 16 signifikantnih mest.

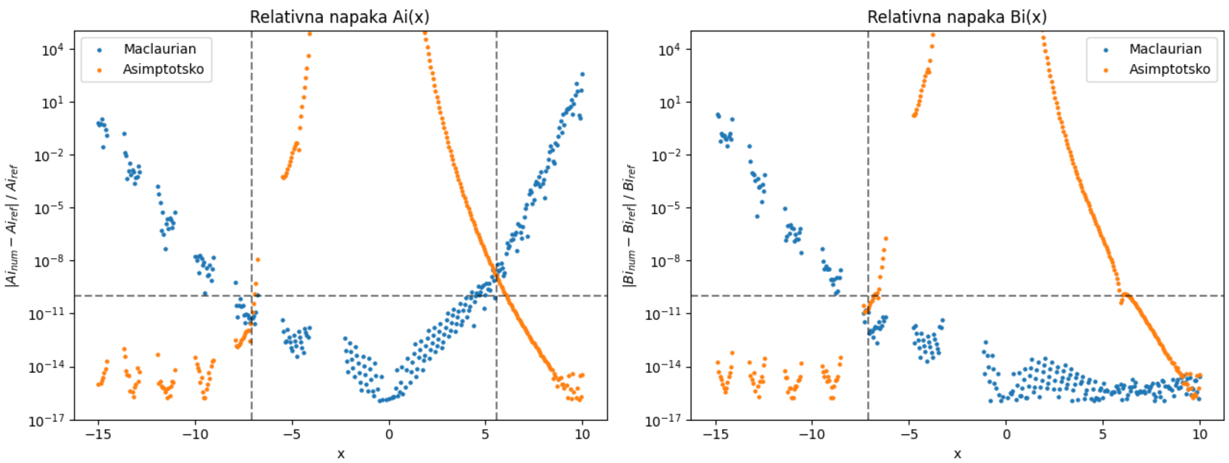


Slika 2: Graf Absolutne napake za Ai(x)



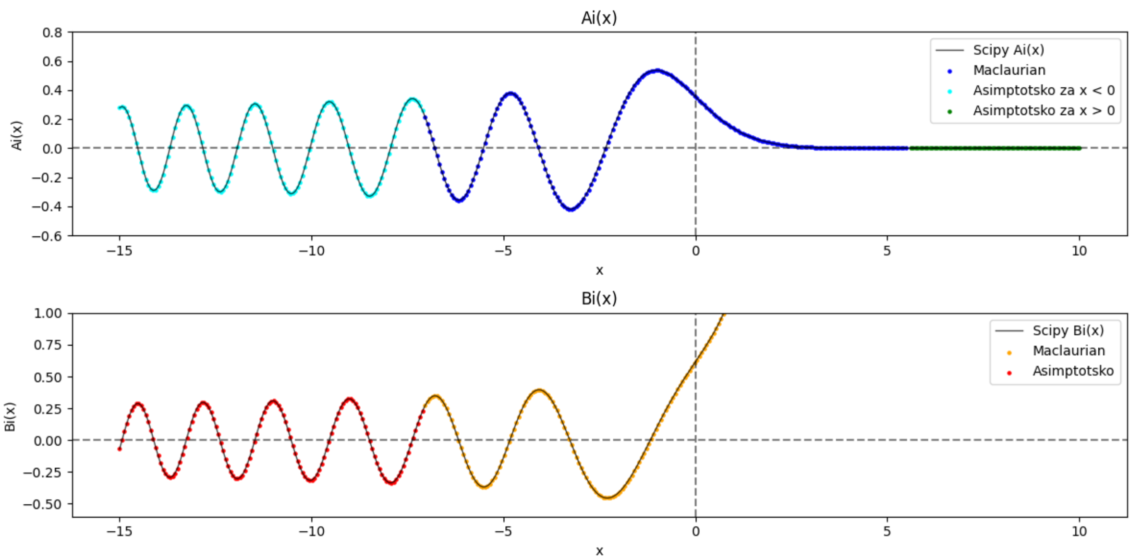
Slika 3: Graf Absolutne napake za Bi(x)

Kot pa je razvidno iz grafa 4 tega problema pri relativni napaki za Bi seveda ni.



Slika 4: Graf relativne napake Maclaurin in asimptotskega približka.

Zdaj poznamo optimalen način da zlepimo približka skupaj. Za funkcijo $Ai(x)$ na negativni polosi zamenjamo iz asimptotskega v Maclaurinov razvoj pri $x = -7.074$, ter na pozitivni osi ravno obratno pri $x = 5.5478$. Za $Bi(x)$ pa samo zamenjamo le na negativni polosi. Zlepljen graf je prikazan spodaj 5.



Slika 5: Zlepljena grafa Airyjevih funkcij Ai in Bi.

5 Iskanje ničel

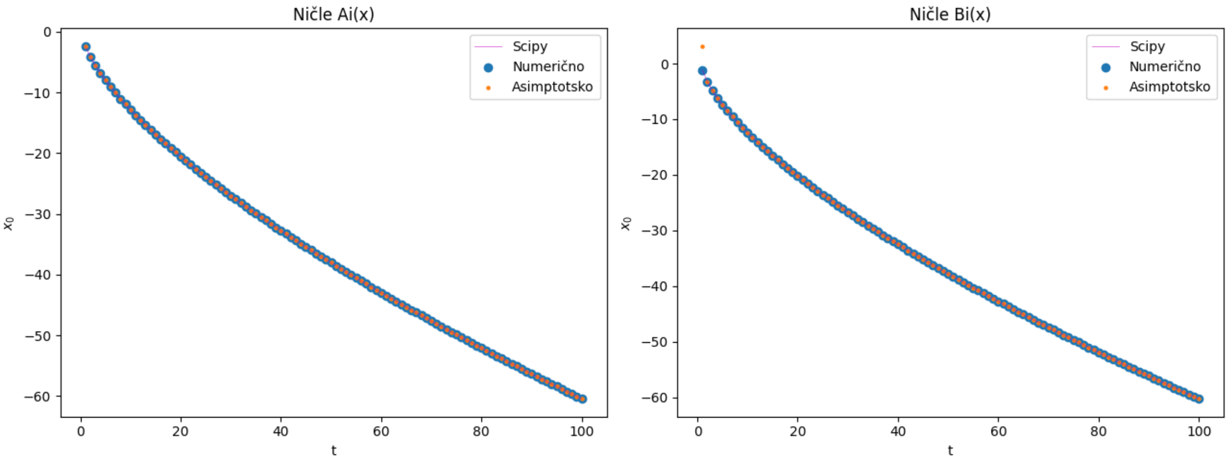
Najprej sem implementiral referenčne vrednosti za ničle Airyjevih funkcij z uporabo *scipy.special.ai_zeros* in *scipy.special.bi_zeros*. Nato sem implementiral formule podane v nalogi za izračun prvih stotih ničel $\{a_s\}_{s=1}^{100}$ in $\{b_s\}_{s=1}^{100}$:

$$a_s = -f\left(\frac{3\pi(4s-1)}{8}\right), \qquad b_s = -f\left(\frac{3\pi(4s-3)}{8}\right), \qquad s = 1, 2, \dots ,$$

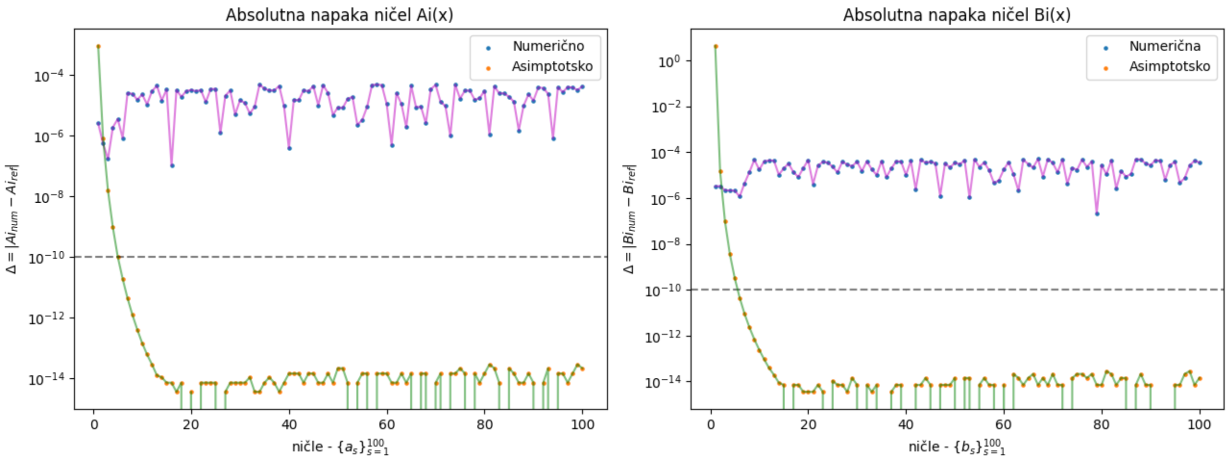
kjer ima funkcija f asimptotski razvoj [4]

$$f(z) \sim z^{2/3}\left(1 + \frac{5}{48}\,z^{-2} - \frac{5}{36}\,z^{-4} + \frac{77125}{82944}\,z^{-6} - \frac{108056875}{6967296}\,z^{-8} + \dots\right).$$

Te približke sem uporabil kot začetne ugibe za ničle zlepljeno funkcijo iz prejšnjega poglavja. Za iskanje ničel sem uporabil funkcijo *mpmath.findroot*.



Slika 6: Graf prvih 100 ničel Airyjevih funkcij Ai in Bi.



Slika 7: Absolutna napaka prvih 100 ničel.

Tukaj se beseda 'numerično' sklicuje na zlepljeno funkcijo. Ničle izračunane pri numeričnem postopku imajo dosti večjo absolutno napako kot pa z uporabljeno formulo. Bilo bi vredno poskusiti vpeljati še bolj natančen izračun, vendar mi je za to žal zmanjkalo časa.

6 Zaključek

Cilj te naloge je bil, da se na brihten način izognemu teškemu računanju z $'raw computing power'$, ki bi nam poleg veliko časa še vzelo veliko natančnosti. Zaradi tega sem prepisal približke v rekurzivni zapis in pazil na natančnost zapisa. Zanašal sem se na Pythonov 16 mestni zapis števil, kar je zadostovalo za obseg te naloge. Za večjo natančnost bi seveda lahko vpeljal še knjižnici *Decimal* ali pa *Mpmath*. Dodatno izboljšanje bi lahko bil test časovne zahtevnosti, in optimizacija po le tej. Z malo več razpoložljivega časa bi morda bilo vredno zapisati kodo v $FORTRAN$ ali pa $C++$ jeziku.

Literatura

[1] O. Vallée, M. Soares, *Airy functions and applications to physics*, Imperial College Press, London 2004.

[2] G. Szegö, *Orthogonal polynomials*, AMS, Providence 1939.

[3] L. D. Landau, E. M. Lifshitz, *Course in theoretical physics, Vol. 3: Quantum mechanics*, 3rd edition, Pergamon Press, Oxford 1991.

[4] M. Abramowitz, I. A. Stegun, *Handbook of mathematical functions*, 10th edition, Dover Publications, Mineola 1972.