Besselova funkcija ploščina hipotrohoide

Aleš Kert, 63170010

15.5.2023

1 Besselova funkcija

Pri implementaciji Besselove funkcije smo primerjali dve metodi - adaptivno trapezno pravilo in adaptivno Simpsonovo pravilo. Na Sliki 2 je prikazana primerjava trapeznega pravila s Simpsonovim. V našem primeru je Simpsonovo pravilo veliko bolj primerno za aproksimacijo površine pod sinusoido.

Pri velikih vrednostih x, se "nagibanost" funkcije znatno poveča, kar naredi izračun vrednosti s podano toleranco veliko zahtevnejši, kar je vidno na Sliki 3, kjer je viden čas izvajanja adaptivnega Simpsonovega pravila. Način, kako bi to eksponentno rast časovne zahtevnosti omejili, je uvedba asimptotične oblike funckije, ki se izračuna pri visokih vrednostih x, vendar pa se je v naših eksperimentih izkazalo, da je za zahtevano natančnost potrebna zelo visoka meja, ter se zdi uporabnost takšne rešitve vprašljiva. Uporabljena asimptotična funkcija je prikazana v Enačbi 1.

Pridobljena krivulja Besselove funkcije za razpon vrednosti $x \in [0, 50]$ je vidna na Sliki 1.

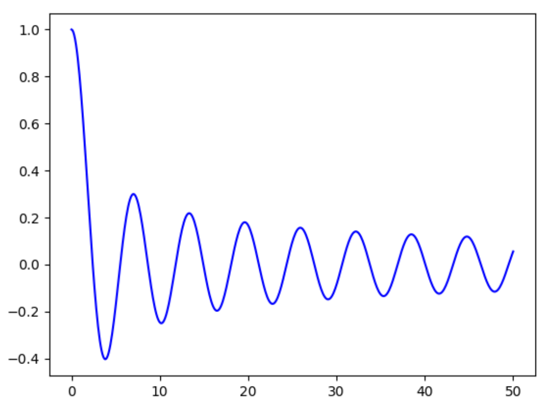

Figure 1: Prikaz Besselove funkcije.

$$J_0 \approx \sqrt{\frac{2}{x\pi}} cos(x - \frac{\pi}{4}) \tag{1}$$

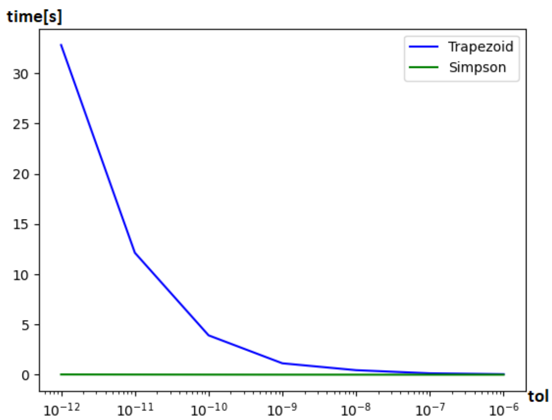

Figure 2: Primerjava adaptivnega trapeznega pravila in adaptivnega Simpsonovega pravila. Manjša kot je toleranca, večji je potreben čas, kar je bolj opazno pri trapeznem pravilu. Opomba : toleranca pada od desne proti levi.

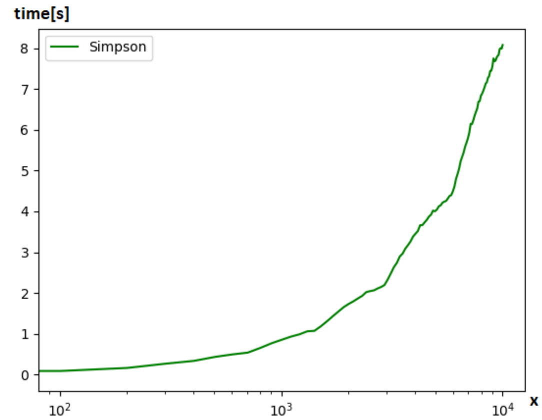

Figure 3: figure
Primerjava časovne zahtevnosti adaptivnega Simpsonovega pravila za velike vrednosti x. Ker je funkcija pri visokih vrednostih x-a zelo valovita, se časovna zahtevnost hitro poveča.

2 Ploščina hipotrohoide

Izračuna ploščine hipotrohoide se lahko lotimo z izračunom izseka hipotrohoide, v našem primeru štirinajstinke, in množenjem te ploščine s 14. Vemo, da se izsek, označen na Sliki 4, začne z 0, konča pa s prvim samopresečiščem krivulje. S sledenjem krivulji vemo, da se to presečišče nahaja nekje na intervalu $t \in [0.25\pi, 0.5\pi]$, kar lahko vzamemo za izhodišče. Načeloma bi lahko pregledovali celoten interval od 0.5π naprej, vendar vemo, da je drugi parameter pri presečišču na intervalu $s \in [5.9\pi, 6.1\pi]$, kar skrajša iskanje.

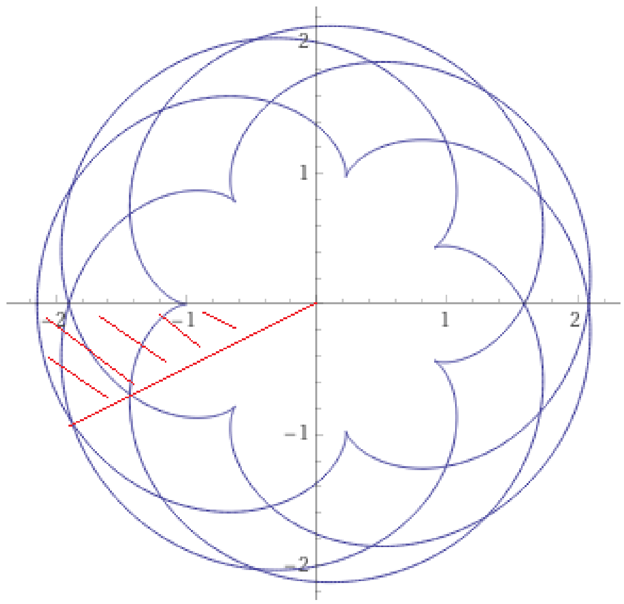

Figure 4: Prikaz hipotrohoide in štirinajstinke ploščine, ki nas zanima za izračun ploščine celotne hipotrohodine.

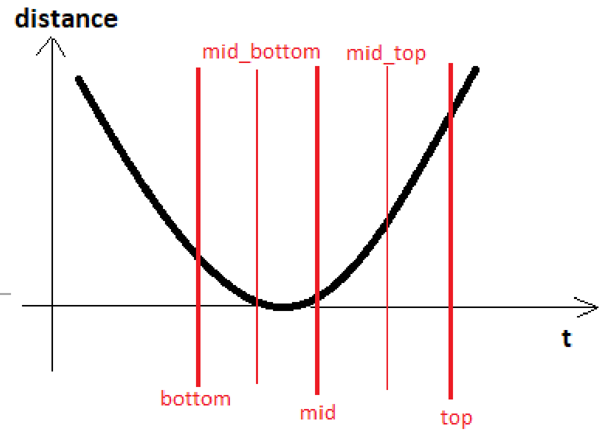

Figure 5: figure
Prikaz predpostavke razdalje točke na prvem intervalu od drugega intervala. Primer pokaže potrebo po vmesnih točkah *mid bottom* in *mid top*. Iz podatka, da je razdalja pri *mid* manjša od razdalje pri *bottom* in *top*, ne moremo izvesti skrčenja intervala, brez da bi tvegali preskok čez ničlo. *Mid* je še vedno lahko na levi ali na desni strani presečišča.

Ker je razdalja nenegativna vrednost, se tega ne moremo lotiti z navadno bisekcijo, zato uvedemo prirejeno bisekcijo, ki uporablja dve vmesni vrednosti, kot je vidno na Sliki 5, za krčenje intervala iskanja. Vse vrednosti so srednje vrednosti med *top* in *bottom*. Premikanje mej *top* in *bottom* se izvaja po naslednjem algoritmu, kjer je xy točka na prvem intervalu, ostale točke pa so na drugem. Vsako iteracijo se razdalje in vmesne vrednosti izračunajo na novo.

$mid\_distance \leftarrow distance(xy, xy\_mid)$
$mid\_top\_distance \leftarrow distance(xy, xy\_mid\_top)$
$mid\_bottom\_distance \leftarrow distance(xy, xy\_mid\_bottom)$
if $mid\_top\_distance \geq mid\_distance$ & $mid\_bottom\_distance \geq mid\_distance$ **then**
  $top \leftarrow mid\_top$
  $bottom \leftarrow mid\_bottom$
else if $mid\_top\_distance \geq mid\_distance$ & $mid\_bottom\_distance \leq mid\_distance$ **then**
  $top \leftarrow mid\_top$
else if $mid\_top\_distance \leq mid\_distance$ & $mid\_bottom\_distance \geq mid\_distance$ **then**
  $bottom \leftarrow mid\_bottom$
end if

Takšen algoritem deluje, če predpostavimo, da je celoten interval znotraj "doline" nekega presečišča, oziroma razdalja na intervalu $[bottom, x]$ vedno pada, na intervalu $[x, top]$ pa vedno raste. Pri zelo majhni toleranci za napake se lahko pojavi problem, kjer zaradi računske napake ta predpostavka propade, vendar znotraj zaželjene natančnosti tega pojava ni. Podrobnosti implementacije pa so razloženi v komentarjih kode. V algoritmu se uporablja prirejena bisekcija za ocenitev vrednosti, ki jo uporablja druga prirejena bisekcija.

Izračunano presečišče x nato uporabimo v naslednji formuli, da pridobimo ploščino izseka. Za izračun integrala je bilo uporabljeno adaptivno Simpsonovo pravilo.

$$p = \frac{1}{2}\int_0^x \dot{x}y - x\dot{y}. \tag{2}$$

Končni rezultat nam vrne ploščino 14.716887324753069.

References