Iskanje ničel v 1D in numerična integracija funkcij

23. 2. 2022

Reševanje enačb v 1D

Numerično reševanje enačb z eno neodvisno spremenljivko x oziroma iskanje ničel funkcij

$$\boxed{f(x) = 0}$$

<u>Opomba</u>: Iskanje večdimenzionalnih ničel $\vec{x}$ vektorskih funkcij $\vec{f}(\vec{x}) = \vec{0}$ je precej bolj zapleteno. S tem se ne bomo ukvarjali.

<u>Komentar</u>:

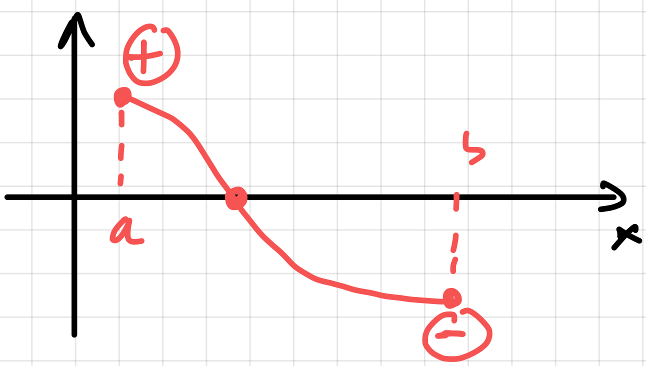
1. Ne obstaja noben numerični algoritem, ki bi nam poiskal vse ničle funkcije
2. Obstajajo algoritmi, ki v okviru numerične natančnosti najdejo ničlo, za katero vemo, da se nahaja znotraj intervala $[a, b]$.

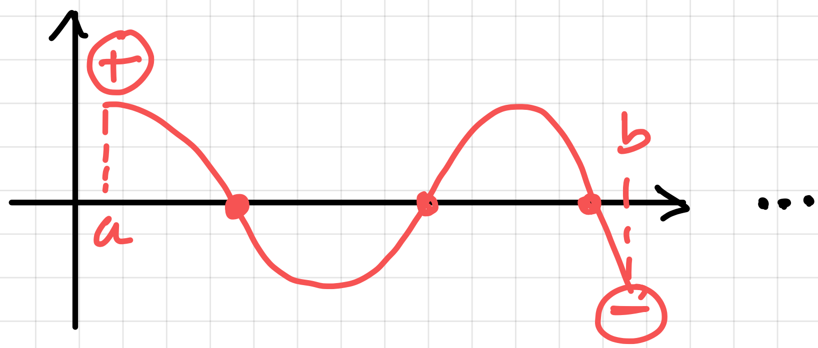
$\Rightarrow$ Ničla je lokalizirana znotraj izbranega/določenega intervala.

$$\boxed{\text{Kako določiti intervale } [a, b]?}$$

Izrek: Če je f realna zvezna funkcija na $[a,b]$ in velja, da je $f(a)\cdot f(b) < 0$, potem obstaja tak $\xi \in [a,b]$, da je $f(\xi)=0$.

Torej: Če najdemo interval $[a,b]$, da velja, da je $f(a)\cdot f(b) < 0$, potem lahko na intervalu pričakujemo liho število ničel: $1, 3, 5, \ldots$

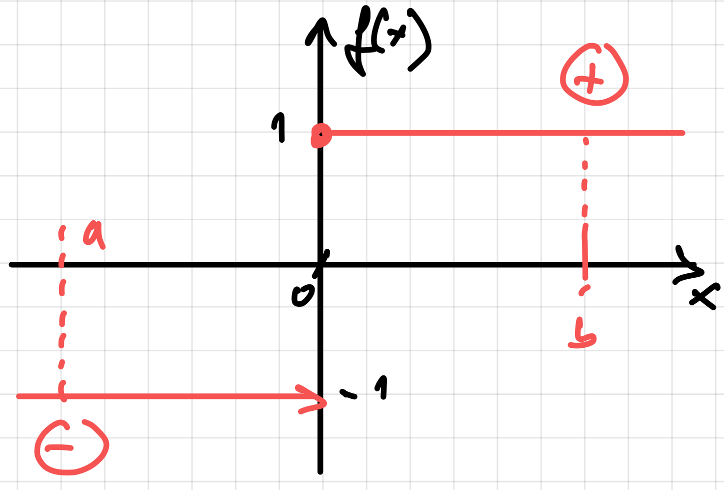




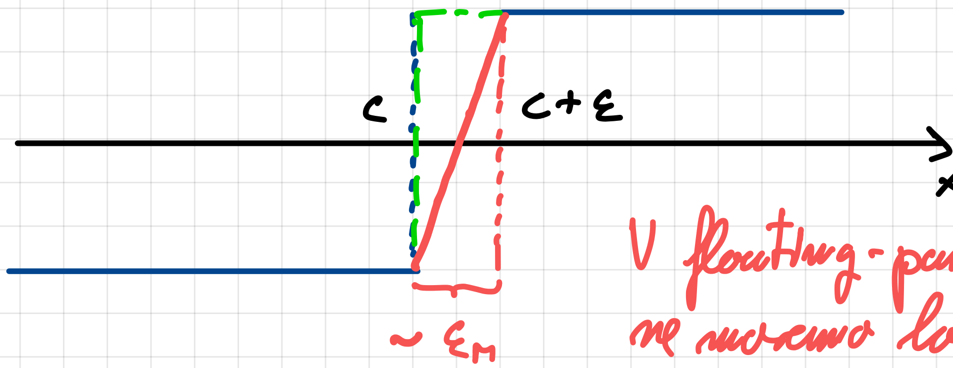
Pozor: Če f na intervalu ni zvezna, je pa omejena, potem ima funkcija na tem intervalu nezveznost, ki preide ničlo:

Npr.:

$$f(x) = \begin{cases} 1 & ; \quad x \geq 0 \\ -1 & ; \quad x < 0 \end{cases}$$



Numerični algoritmi takšne nezveznosti ne znajo ločiti od ničle! Prečkanje ničle bi lahko zgodilo med dvema sosednjima številoma, ki jih lahko predstavimo v računalniku s plavajočo vejico.



V floating-point reprezentaciji ne moremo ločiti točk dveh skokov.

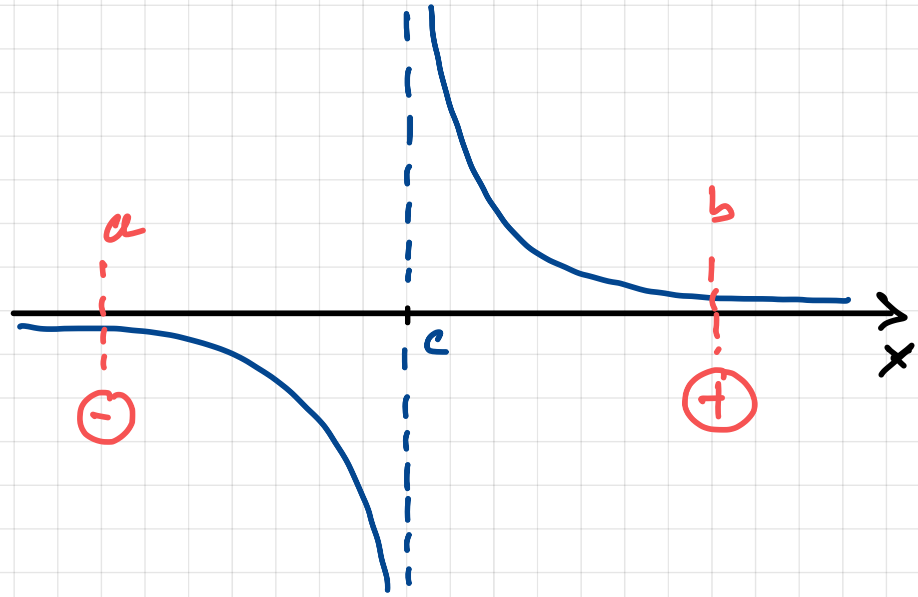
Nezvezna funkcija pa ima lahko na intervalu tudi singularnost. Nekateri algoritmi (npr. Bisekcija) znajo konvergirati tudi k singularnosti. Singularnosti ne zamešamo z ničlo tako, da preverimo vrednost funkcije. Velja:

NIČLA: $|f(x \to c)| \to 0$

SINGULARNOST: $|f(x \to c)| \to \infty$

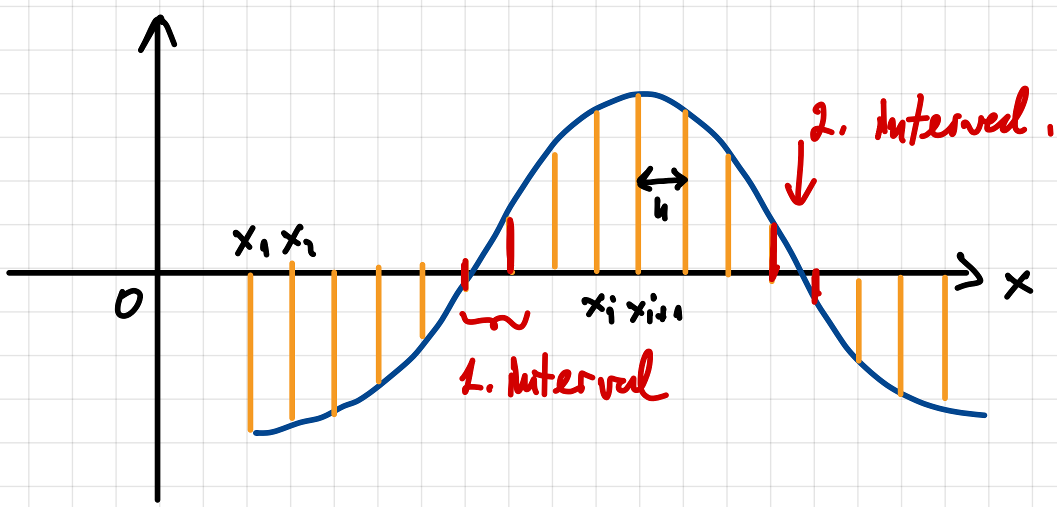
Npr:

$$f(x) = \frac{1}{x-c}$$



Iskanje ničel začnemo tako, da definicijsko območje funkcije, ki nas zanima, razdelimo z dovolj gosto mrežo točk x_i, ki so med seboj razmaknjene za h: $\{x_i\ ;\ i = 1, 2, \ldots N\}$

$$x_{i+1} = x_i + h.$$



Poiščemo vse intervale $[x_i, x_{i+1}]$, kjer je

$$f(x_i) \cdot f(x_{i+1}) < 0$$

Bisekcija (Deljenje intervala na polovice)

Ideja metode je ta, da interval $[a, b]$ delimo na vedno manjše polovice, ki pri tem izbiramo tiste polovice, pri katerih ima f na robovih intervala različen predznak. Tako postopoma ujamemo ničlo med vedno ožje meje intervala $[a_n, b_n]$; $n = 1, 2, \dots N$.

Koraki:

1.) Izberi začetni interval $[a_1, b_1]$, kjer je $f(a_1) \cdot f(b_1) < 0$.

2.) Izračunaj točko na sredini intervala

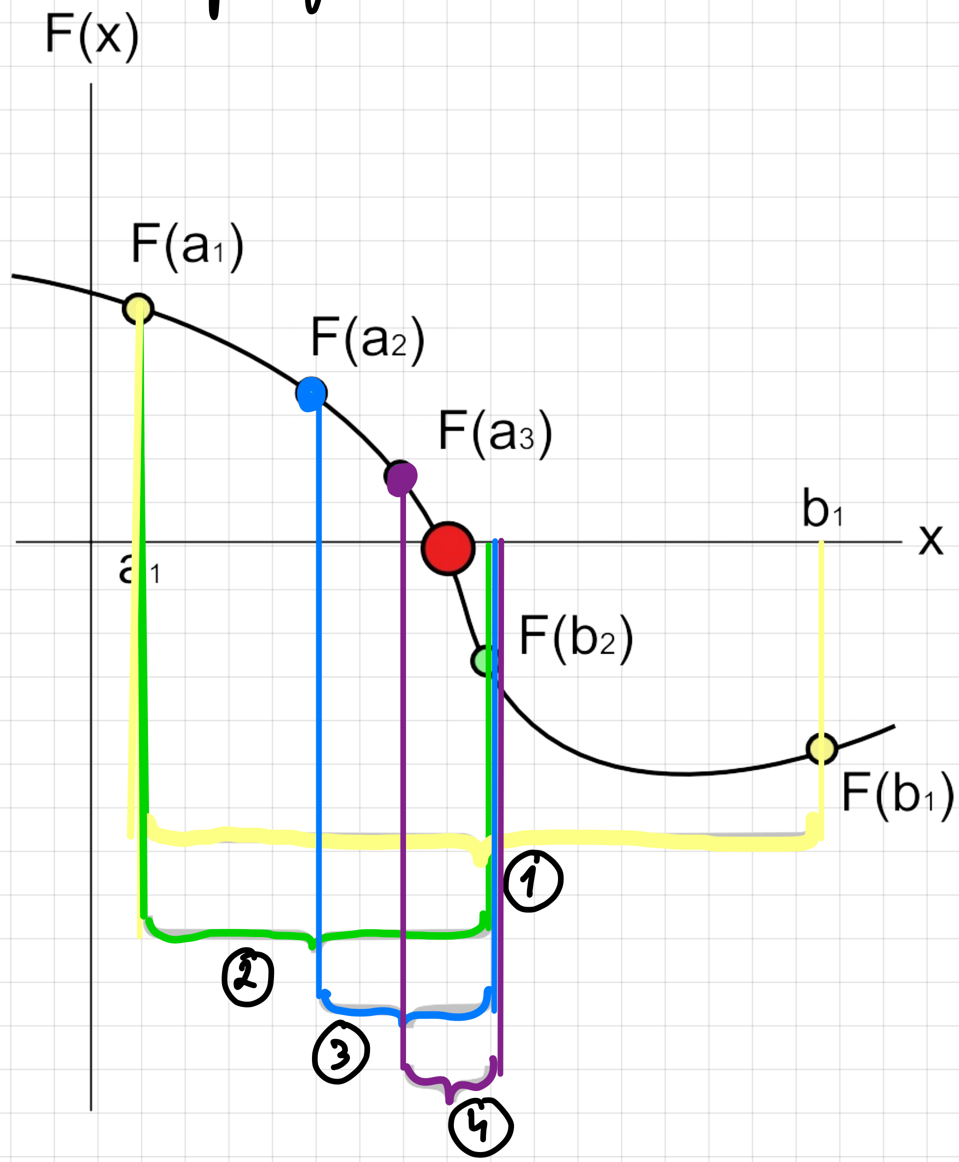
$$x = \frac{a_1 + b_1}{2}$$

3.) Določimo nov, za $\frac{1}{2}$ ožji interval, kjer je ničla:

- Če je $f(a_1) \cdot f(x) < 0$ potem $a_2 = a_1$, $b_2 = x$
- Če je $f(x) \cdot f(b_1) < 0$ potem $a_2 = x$, $b_2 = b_1$

4.) Koraka 2 in 3 ponavljamo dokler $\underbrace{|b_n - a_n|}_{\text{Širina intervala je efektivno 0}} < \varepsilon$ oziroma $\underbrace{f(x_n)}_{\text{Vrednost funkcije je cca. 0}} < \varepsilon$.

5.) Vrnemo najboljšo vrednost za ničlo: x_n.



Število korakov, ki jih v splošnem potrebujemo za določitev ničle z natančnostjo ε znamo oceniti.

Širina intervala v 1. koraku je: $h_1 = b_1 - a_1$

Širina intervala v $(n+1)$-vem koraku je:

$$h_{n+1} = \frac{1}{2} h_n = \frac{1}{2}\cdot\frac{1}{2} h_{n-1} = \frac{1}{2^n}\cdot h_1 = \frac{b_1 - a_1}{2^n}$$

Temu rečemo linearna konvergenca, saj natančnost pridobivamo linearno s številom korakov.

Nehamo pri Hotem n, kjer je $h_{n+1} < \varepsilon$:

$$\varepsilon = \frac{b_1 - a_1}{2^n} \Rightarrow 2^n = \frac{b_1 - a_1}{\varepsilon} \Rightarrow \boxed{n = \log_2 \frac{b_1 - a_1}{\varepsilon}}$$

<u>Zgled.</u> Ničla funkcije je na intervalu $[0, 1]$. Želimo jo določiti z natančnostjo $\varepsilon = 10^{-15}$. Število potrebnih korakov je:

$$n = \log_2 \frac{1}{\varepsilon} = \log_2 10^{15} = \frac{\log 10^{15}}{\log 2} = \frac{15}{0.3} = \underline{\underline{50}}$$

<u>Komentar:</u>

1.) <u>Bisekcija vedno uspe!!!</u>

2.) Konvergira linearno, kar ni slabo, ampak počasneje od ostalih metod.

Metoda napačnega pravila (lat. Regula falsi)

Tudi pri tej metodi ožamo interval okrog ničle, le da tu intervala v vsakem koraku ne delimo na polovice, pač pa delamo linearno interpolacijo med dvema točkama $[a_n, b_n]$, kjer velja $f(a_n) \cdot f(b_n) < 0$

1.) Izberemo začetni interval $[a_1, b_1]$, kjer velja $f(a_1) \cdot f(b_1) < 0$.

2.) Določi linearno funkcijo, ki povezuje, ti dve točki:

$$g(x) = kx + n = \frac{f(b_1) - f(a_1)}{b_1 - a_1}(x - a_1) + f(a_1)$$

3.) Poiščemo, kje premica seka os x: $g(x) = 0$

$$x = f(a_1) \cdot \frac{a_1 - b_1}{f(b_1) - f(a_1)} + a_1$$

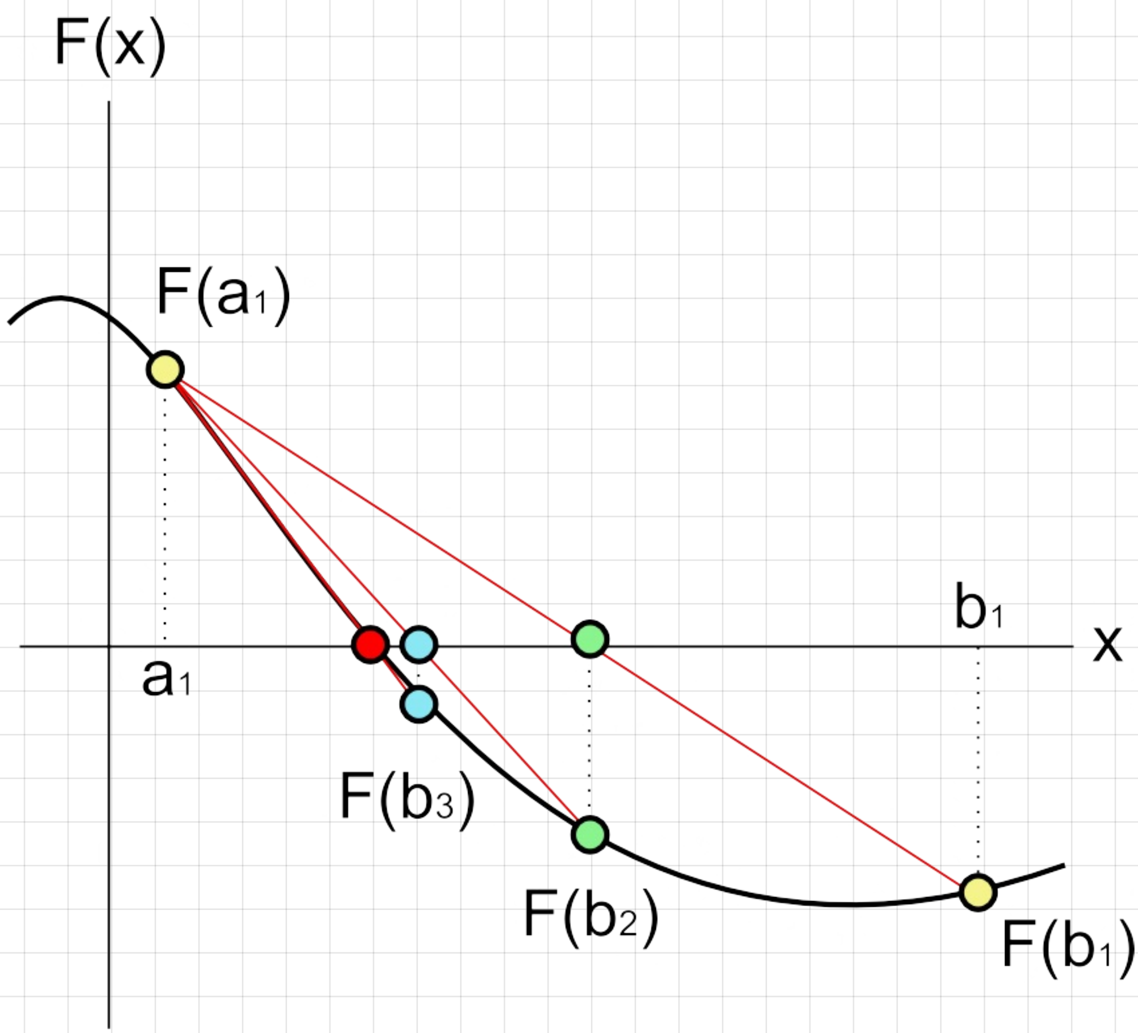
$$= \frac{\cancel{f(a_1) \cdot a_1} - f(a_1) b_1 + f(b_1) a_1 - \cancel{f(a_1) a_1}}{f(b_1) - f(a_1)}$$

$$= \frac{f(b_1) a_1 - f(a_1) b_1}{f(b_1) - f(a_1)}$$

4.) Uporabimo enak kriterij, kot pri bisekciji

- Če je $f(a_1) \cdot f(x) < 0$ potem $a_2 = a_1$, $b_2 = x$
- Če je $f(x) \cdot f(b_1) < 0$ potem $a_2 = x$, $b_2 = b_1$

5.) Korake 2,3,4 ponavljamo dokler $|b_n - a_n| < \varepsilon$ oziroma $f(x_n) < \varepsilon$.



Komentarji:

1.) Linearno funkcijo $g(x)$ uporabimo na podlagi privzetka, da se funkcija $f(x)$ v bližini ničle obnaša linearno $\Rightarrow$ Linearna interpolacija.

2.) Regula-falsi konvergira hitreje kot bisekcija

$$\boxed{h_n = c\cdot(h_{n-1})^m \;;\quad m > 1.}$$

3.) Lahko se zgodi, da se pri določenih primerih nekaj časa premika le ena točka:

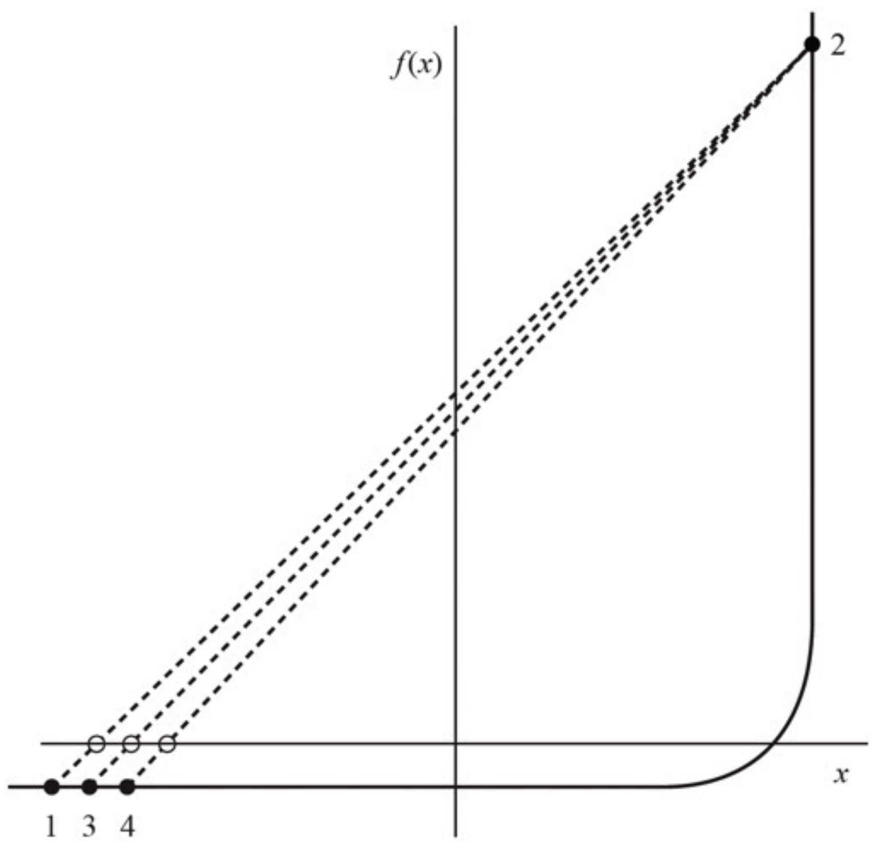


Figure 9.2.3. Example where both the secant and false position methods will take many iterations to arrive at the true root. This function would be difficult for many other root-finding methods.

Sekantna metoda:

Tudi pri tej metodi funkcijo v bližini ničle aproksimiramo z linearno funkcijo, a ne preverjamo predznakov funkcije na robu intervala, pač pa vrednosti, ki jih dobivamo – ničle linearne funkcije – uporabljamo kot zaporedne približke ničel.

1.) Meji začetnega intervala uporabimo kot začetna približka ničle: $x_0 = a$, $x_1 = b$.

2.) Naslednji približek določimo z enako interpolacijo kot pri Regula-Falsi:

$$\boxed{x_{n+1} = \frac{f(x_n)\,x_{n-1} - f(x_{n-1})\,x_n}{f(\underline{x_n}) - f(\underline{x_{n-1}})}}$$

V vsakem koraku si moramo zapomniti zadnja dva približka.

3.) Algoritem ponavljamo brez dodatnega preverjanja predznakov, dokler: $|x_{n+1} - x_n| < \varepsilon$.

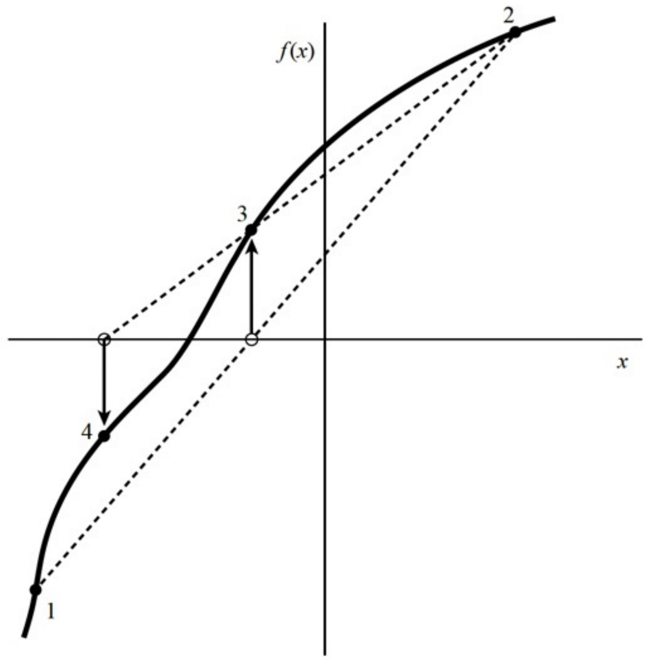


Figure 9.2.1. Secant method. Extrapolation or interpolation lines (dashed) are drawn through the two most recently evaluated points, whether or not they bracket the function. The points are numbered in the order that they are used.

Komentarji:

1.) Sekantna metoda konvergira hitreje kot bisekcija

$$h_{n+1} = C \cdot (h_n)^{1.618}$$

Vidimo, da konvergira superlinearno!

2.) Metoda ima slabost, da gredo lahko približki x_n izven začetnega intervala $[a, b]$.

3.) Če so začetni približki slabi, ali pa funkcija ni dovolj pohlevna, metoda lahko ne konvergira. ⇒ Lokalno obnašanje je lahko pobegne v ∞.

Tangentna ali Newtonova metoda
(tudi Newton - Raphson)

Metoda je podobna sekantni metodi, le da tu funkcijo v bližini ničle ne aproksimiramo s sekantami, pač pa s tangentami. V n-tem približku zapišemo:

2.) $g(x) \longrightarrow \boxed{t(x) = f(x_n) + f'(x_n)(x - x_n)}$

(sekantna metoda)

! Pri tej metodi potrebujemo poleg funkcijskih vrednosti tudi odvod funkcije.

Poiščemo ničlo tangente:

$$t(x_{n+1}) = f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0$$

3.) $$\boxed{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}}$$

Tako kot prej postopek ponavljamo toliko časa, da je $|x_{n+1} - x_n| < \varepsilon$.

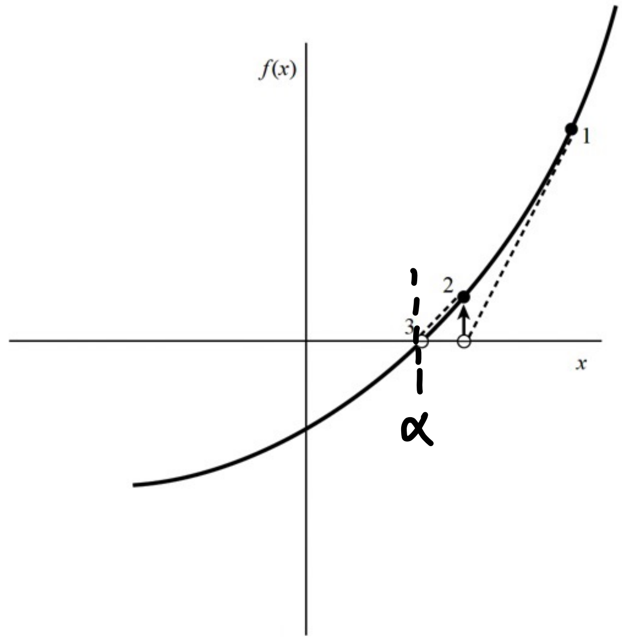


Figure 9.4.1. Newton's method extrapolates the local derivative to find the next estimate of the root. In this example it works well and converges quadratically.

Komentarji:

1.) Matematična podlaga za Newtonovo metodo je razvoj funkcije v Taylorjevo vrsto. Naj bo α ničla funkcije f: $f(\alpha) = 0$. Razvijmo funkcijo sedaj okrog x, ki se nahaja v okolici α:

$$f(\alpha) = f(x) + f'(x)\underbrace{(\alpha - x)}_{\delta} + \frac{f''(x)}{2}\underbrace{(\alpha - x)}_{\delta}^2 + \cdots$$

Če je $\delta = \alpha - x$ majhen, potem lahko upoštevamo le linearni člen. Dobimo:

$$f(\alpha) = f(x) + f'(x)(\alpha - x)$$

Od tod po:

$$\alpha = x - \frac{f(x)}{f'(x)}$$

Če je δ preveliko, potem linearni približek ni dovolj dober in ne zadanemo ničle v enem zamahu, pod pa moramo postopek iterativno ponavljati.

2.) Konvergenca metode je praviloma kvadratična

$$\alpha(x) = x - \frac{f(x)}{f'(x)}$$

$$\alpha'(x) = 1 - \frac{f'(x)^2 - f(x)f''(x)}{(f'(x))^2} = \frac{\overbrace{f(x)}^{\text{0 v bližini ničle}} f''(x)}{f'(x)^2} = 0$$

$$\alpha''(x) = \frac{[f'(x)f''(x) + \overset{=0}{f(x)} \cdot f'''(x)]f'(x)^2 - 2\overset{=0}{f(x)}f''(x) \cdot f'(x)f''(x)}{f'(x)^4}$$

$$= \frac{f''(x)}{f'(x)}$$

To pomeni (ker razvijemo $\alpha(x)$ v Taylorja)

$$h_{n+1} = \cancelto{0}{\alpha'(x)} h_n + \frac{\alpha''(x)}{2} h_n^2 + \cdots = C h_n^2$$

Če so ničle večkratne, $(x-\alpha)^m$, potem je konvergenca slabša.

3.) Kadar smo precej stran od ničle (kjer naš red razvoja postane nepomemben) lahko da Newtonova metoda precej slabe/nesmiselne rezultate, metoda ne konvergira

- Če zadanemo odvod $f'(x_n)=0$, lahko rezultat odnese $\to \infty$.

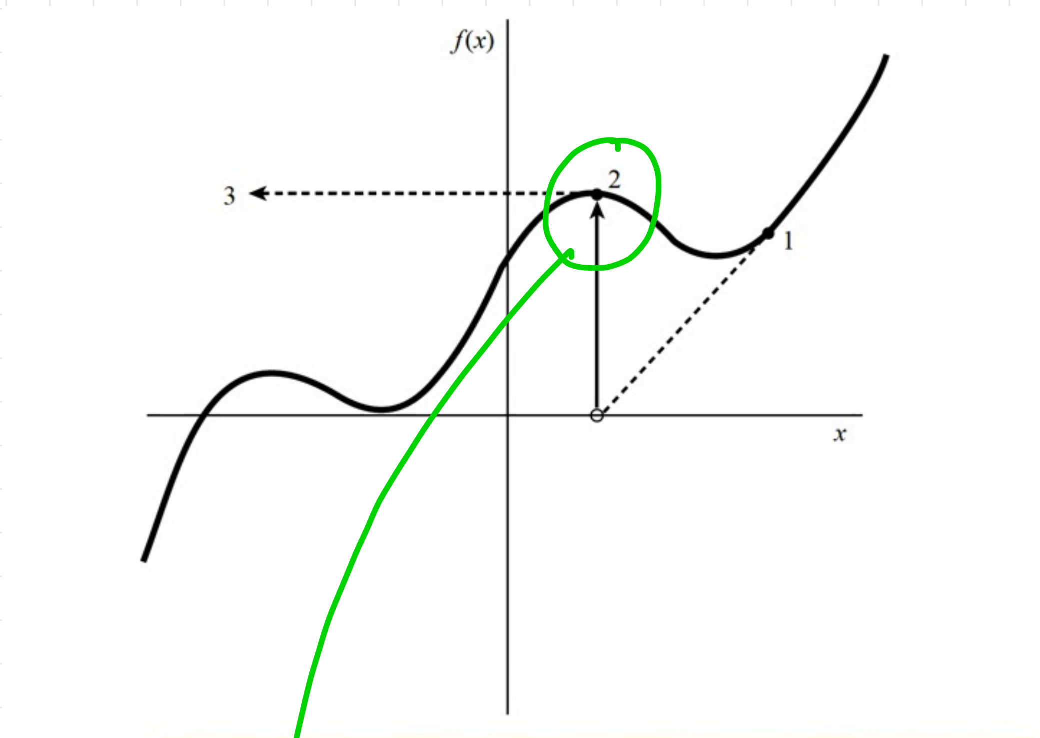
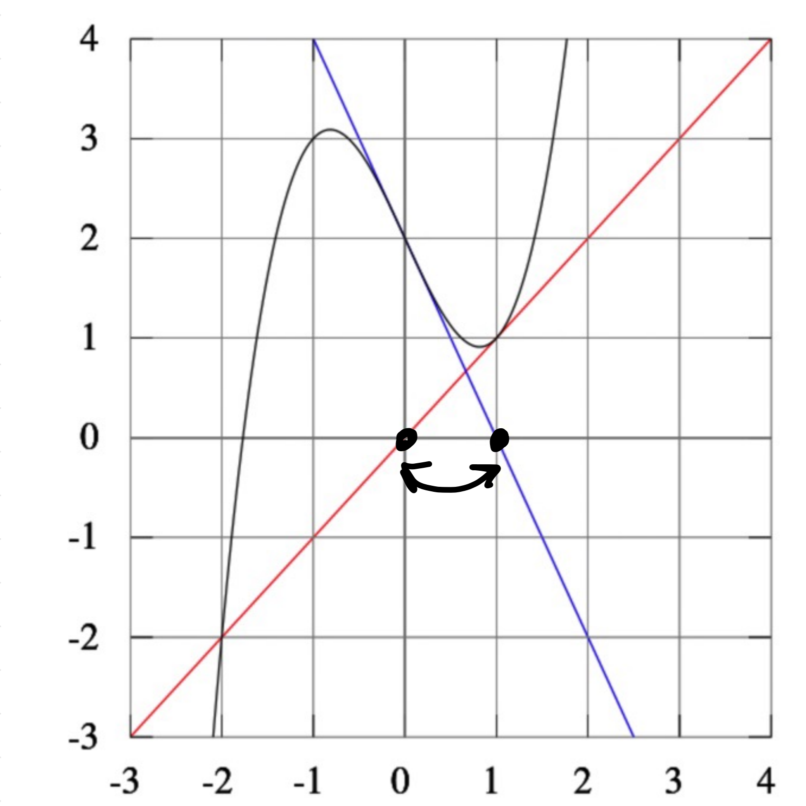


Figure 9.4.2. Unfortunate case where Newton's method encounters a local extremum and shoots off to outer space. Here bracketing bounds, as in rtsafe, would save the day.

Divje obnašanje v okolici $f'(x)=0$ daje zanimive slike Newtonovih fraktalov!

• Lahko pa se ujame v past in skače med dvema približkoma.



4.) Tudi če metoda ni a priori uporabna, ker dela na začetku slabše konvergenčne lastnosti, lahko z njo "spoliramo" ničle, ko smo jim enkrat blizu ⇒ kvadratična konvergenca.

5.) Metoda ni omejena le na 1D, pač pa jo lahko uporabljamo tudi za reševanje enačb tipa

$$\boxed{f(\vec{x}) = 0.}$$

⇒ Newtonovi fraktali.

3. "Newtonov fraktal":

Imejmo enačbo: $\boxed{z^3 = 1}$

Enačba ima tri rešitve, eno realno in dve kompleksno konjugirana.

$$z = \rho\, e^{i\theta} = \rho\left[\cos\theta + i\sin\theta\right]$$

$$z^n = \rho^n e^{in\theta} = \rho^n\left[\cos(n\theta) + i\sin(n\theta)\right]$$

Za naš primer: $z^3 = \rho^3 e^{i3\theta} = 1$

Sledi: $3\theta_n = 2\pi n \quad ; \quad n = 0, 1, 2,$

$\rho = 1 \qquad \theta_n = \dfrac{2\pi n}{3}$

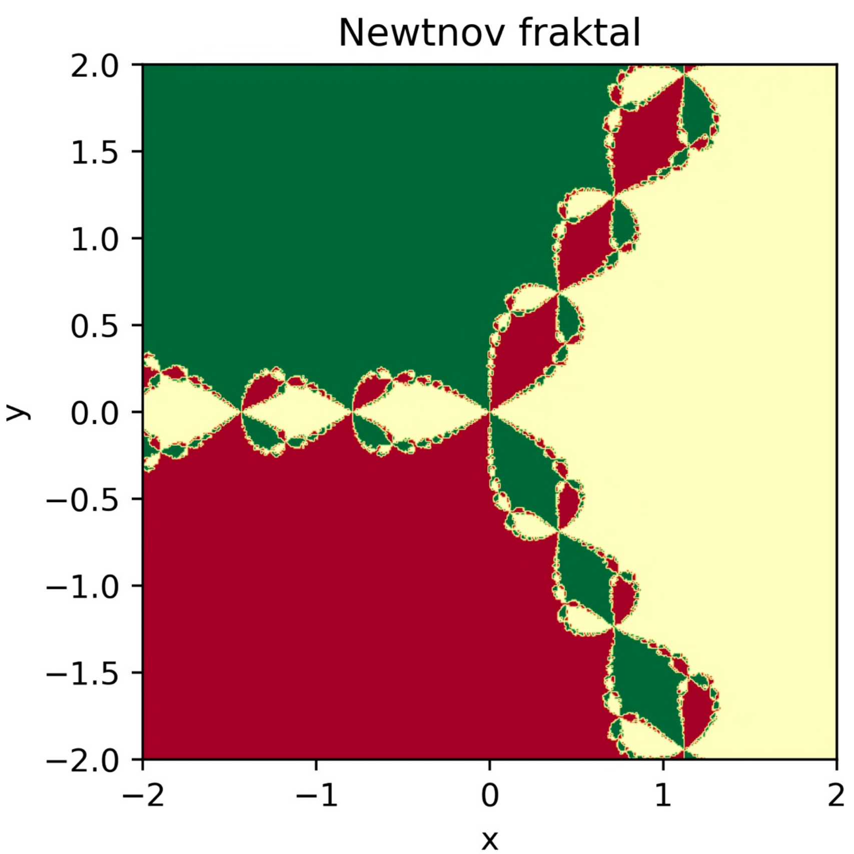
$$\theta_1 = 0, \quad \theta_2 = \frac{2\pi}{3}, \quad \theta_3 = \frac{4\pi}{3}$$

$$z_1 = 1, \quad z_{2,3} = -\frac{1}{2} \pm i\frac{\sqrt{3}}{2}$$

Newtnova metoda deluje tudi za kompleksne ničle:
Dobimo iterativno formulo:

$$\boxed{z_{n+1} = z_n - \frac{z_n^3 - 1}{3 z_n^2}}$$

Ko enačbo rešujemo numerično z Newtnovo metodo je rešitev precej odvisna od začetnega pogoja. Ta določa, v katero rešitev bomo konvergirali. (Določenosti znamo ena na enkrat. Slika, ki jo dobimo je fraktalna. Če jo "pozoomamo" se ponovi in to neštetokrat.



Brentova metoda:

(polno ime: Van Wijngaarden – Dekker – Brent)

- Je metoda s super linearno konvergenco, ki zagotavlja robustnost bisekcije.

- Metoda uporablja hitro metodo, ki temelji na inverzni kvadratni interpolaciji (sekantna metoda je linearna). Ko ta metoda ne konvergira dobro (npr. ko uide iz prvotnega intervala) pa preklopi na bisekcijo.
- Ključni del metode je spremljanje konvergence metode in odločitev, kdaj preklopiti na bisekcijo.

- Glej NRC, str. 359

Napotki (NRC):

- Najprej si ustvari sliko o tem, kako "izgleda" funkcija preden iščeš njene ničle. ⇒ NARIŠI JO!
- Nato identificiraj interval znotraj katerega veš, da se nahaja ničla, preden poženeš algoritem za iskanje ničel.
- Ničlo funkcije določimo tako, da jo "ujamemo" med dve mejni vrednosti, nato pa jo ulovimo kot zajca

- Konvergenca k ničli
- Hitra konvergenca.

- Uspeh algoritma je močno odvisen od začetnega približka, prve ideje o tem, kje bi morala ničla biti.

To je bolj odvisno od začetne analize in razumevanja problema, kot pa od numerike.

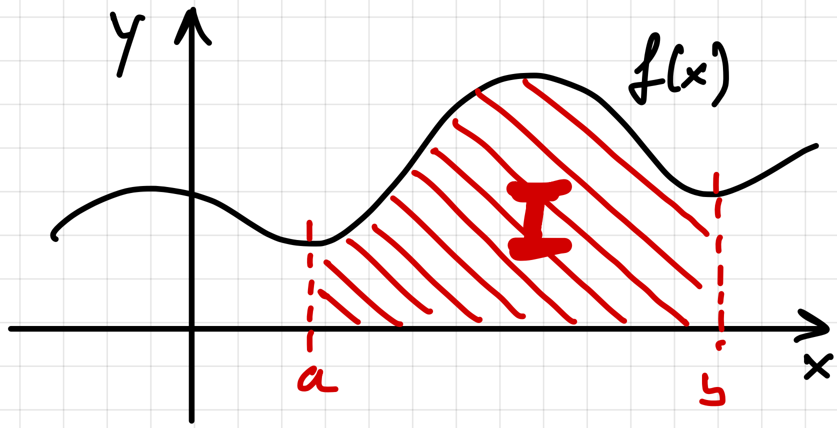
- Ne dovoli, da iterativna metoda med iskanjem ničle kadar koli skoči izven mej intervala na katerem poizkušamo ničlo.

Najboljša splošna metoda za iskanje "lokaliziranih" ničel na vnaprej določenih intervalih je Brentova metoda.

Numerična integracija funkcij

- Izračunati želimo integral funkcije $f(x)$ v 1D v mejah $[a, b]$.

$$I = \int_a^b f(x)\,dx$$



- Integral funkcije je enak ploščini pod krivuljo na izbranem intervalu. ⇒ Formulam zato rečemo tudi KVADRATURNE FORMULE (iz angleščine QUADRATURE (ploščina))

- Metodam za numerično integracijo so posvečali veliko pozornosti v 18. in 19. stoletju. Potreba po dobrih metodah je izhajala iz dejstva, da integralov funkcij (z razliko od odvodov) ne znamo analitično izračunati. ⇒ Potrebujemo dobre numerične približke.

Osnovna ideja:

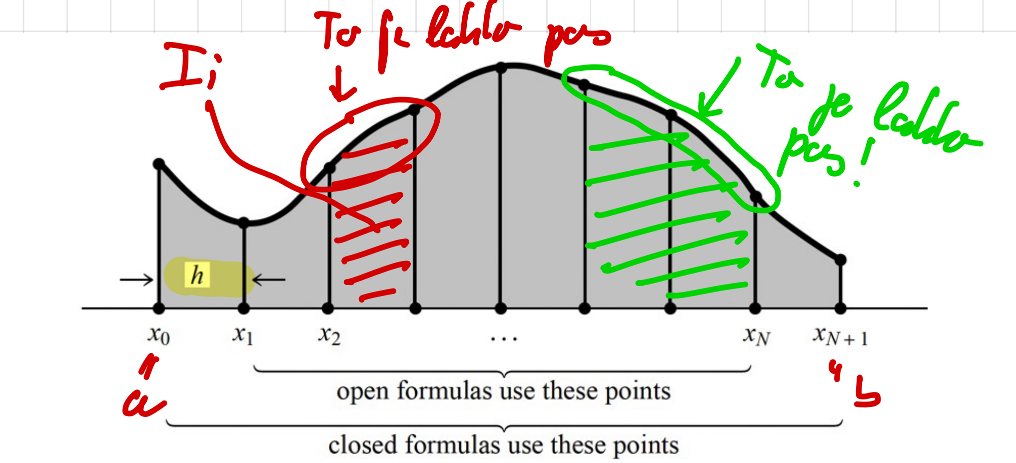


Figure 4.1.1. Quadrature formulas with equally spaced abscissas compute the integral of a function between x_0 and x_{N+1}. Closed formulas evaluate the function on the boundary points, while open formulas refrain from doing so (useful if the evaluation algorithm breaks down on the boundary points).

Na izbranem intervalu $[a, b]$ abscisno razdelimo na $N+1$ ekvidistantnih točk: $x_0, x_1, x_2, \ldots, x_N$, ki so med sabo razmaknjene za $h = \frac{b-a}{N}$. V teh točkah znamo izračunati vrednosti funkcije $f(x_i)$.

Pas lahko sestoji iz 2, 3, 4, 5.... točk.

Z izbranimi točkami celoten integral razdelimo na $M \leq N$ ožjih enako širokih pasov. Osnovni gradnik klasičnih integracijskih metod so pravila, ki povedo, kako izračunati integral funkcije na takem pasu: I_i

To so tako imenovana Newton-Cotesova pravila

Pravila se med sabo ločijo po tem, kako aproksimiramo funkcijo znotraj izbranega pasu. !

Integral funkcije na celotnem pasu dobimo kot vsota integralov posameznih pasov:

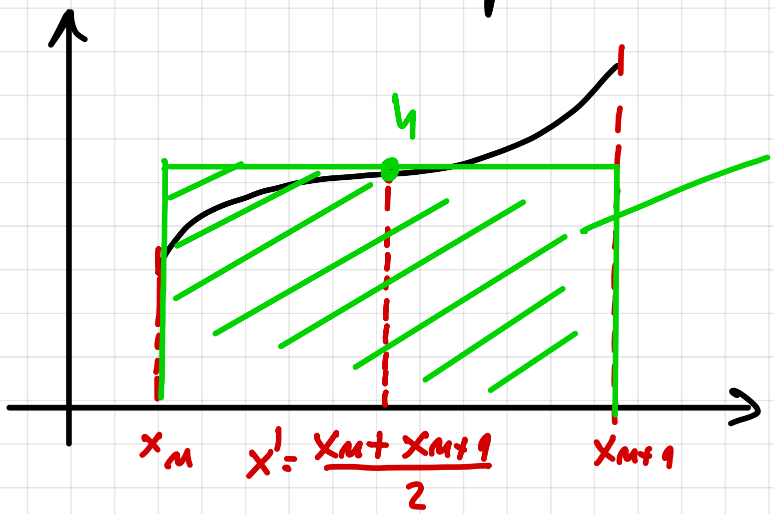
$$\boxed{I = \sum_{i=1}^{M} I_i}$$

Izpeljanim pravilom za celoten interval $[a, b]$ rečemo razširjena Newton–Cotesova pravila.

<u>Opomba:</u> N-C. pravila se delijo na zaprta ali odprta, glede na to, ali metoda upošteva (zaprta) ali ne upošteva (odprta) robni točki $x_0 = a$, $x_M = b$. Mi se bomo ukvarjali le z <u>zaprtimi pravili.</u> Odprta pravila pridejo prav, kadar integriramo funkcije, ki v robni točki podka 0, ali pa ima integrabilno singularnost. Iz odprtih N-C. pravil ne moremo sestaviti razširjenih pravil, zato se praktično ne uporabljajo. Tam kjer so uporabne, pa jih lahko nadomestimo z močnejšimi Gaussovimi integracijskimi formulami.

1.) Pravilo srednje vrednosti. (najpreprostejše)

Metoda po točkah za $f(x)=c$.



Integral funkcije aproksimiramo s pravokotnikom s ploščino $f(x')\cdot h$.

Vrednost II. odvoda v neki točki, ne vemo kateri.

$$\int_{x_n}^{x_{n+1}} f(x)\,dx = f\left(\frac{x_n+x_{n+1}}{2}\right)\cdot \underbrace{(x_{n+1}-x_n)}_{h} + \mathcal{O}\left(h^3 f''(\xi)\right)$$

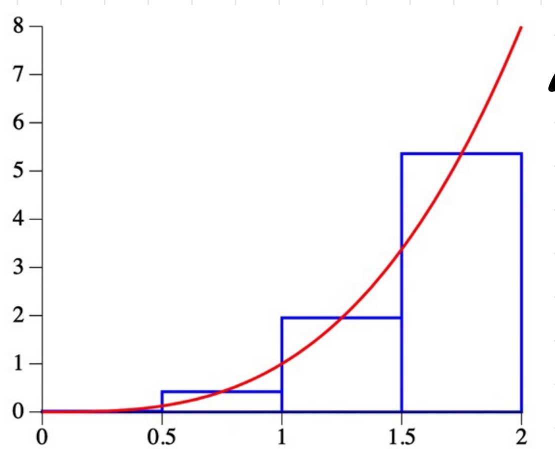
Velikostni red

Napaka pada s 3. potenco velikosti koraka

Formula za celoten interval:

$$\int_{x_0=a}^{x_n=b} f(x)\,dx = h\cdot\left[f\left(\frac{a+x_1}{2}\right)+f\left(\frac{x_1+x_2}{2}\right)+f\left(\frac{x_2+x_3}{2}\right)+\dots f\left(\frac{x_{n-1}+b}{2}\right)\right] + \mathcal{O}\left(\frac{(b-a)^3\cdot f''(\xi)}{N^2}\right)$$

$N\cdot\frac{1}{N^3}$ = ker N delov



To je ekvivalentno histogramiranju funkcije!

2.) Trapezna formula:

Točke interpoliramo z linearno funkcijo oziroma funkcijo $f(x)$ na intervalu $[x_n, x_{n+1}]$ aproksimiramo s premico. Integral $f(x)$ je potem $\sim$ enak ploščini trapeza.

$$\int_{x_n}^{x_{n+1}} f(x)\,dx \approx \int_{x_n}^{x_{n+1}} \frac{f(x_{n+1}) - f(x_n)}{h}(x - x_n) + f(x_n)\,dx =$$

$$= \frac{f(x_{n+1}) + f(x_n)}{h} \cdot \Bigg[\underbrace{\frac{x_{n+1}^2 + x_n^2}{2} - x_n \cdot x_{n+1} + \cancel{x_n^2}}_{\frac{1}{2}(x_{n+1}^2 - 2x_{n+1}x_n + x_n^2) = \frac{1}{2}(x_{n+1} - x_n)^2 = \frac{h^2}{2}} \Bigg] + \cancel{f(x_n)} \cdot h$$

$$= \frac{h}{2}\left[f(x_{n+1}) + f(x_n)\right] + O\left(h^3 \cdot f''(\xi)\right)$$

$f''(\xi) = 0$

Metoda je eksaktna za funkcije $f(x) = kx + n$.

Razširjena formula:

$$\int_a^b f(x)\,dx = h \cdot \left[\frac{1}{2} f(x_0) + f(x_1) + \dots + f(x_{N-1}) + \frac{1}{2} f(x_N)\right]$$

$$+ O\left(\frac{(b-a)^3}{N^2} f''(\xi)\right)$$

Globalna napaka.

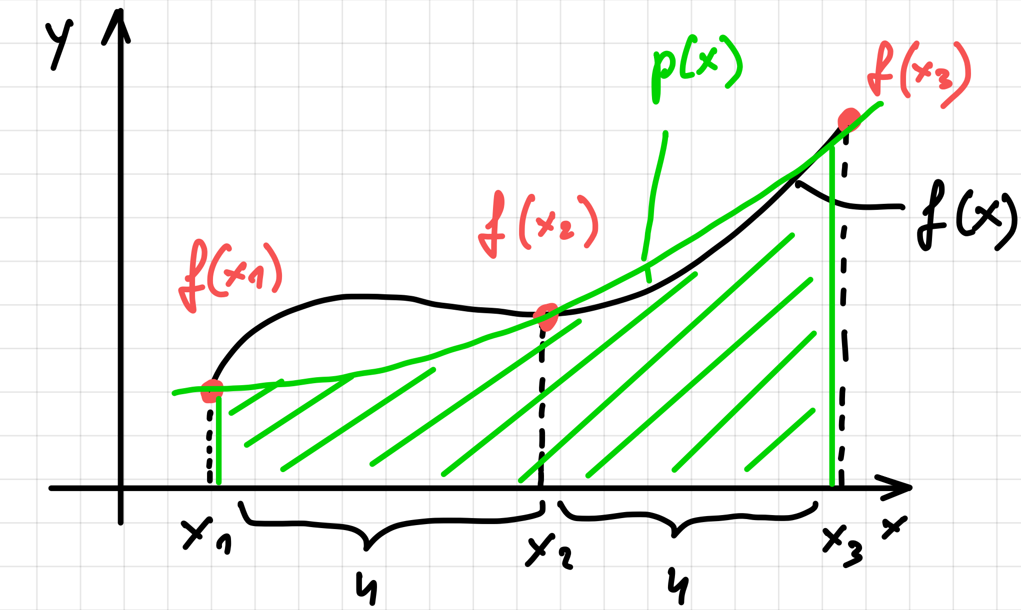
Napaka pada kvadratično z N.

Nrc:

4.1 Classical Formulas for Equally Spaced Abscissas

Where would any book on numerical analysis be without Mr. Simpson and his "rule"? The classical formulas for integrating a function whose value is known at equally spaced steps have a certain elegance about them, and they are redolent with historical association. Through them, the modern numerical analyst communes with the spirits of his or her predecessors back across the centuries, as far as the time of Newton, if not farther. Alas, times *do* change; with the exception of two of the most modest formulas ("extended trapezoidal rule," equation 4.1.11, and "extended

3.) Simpsonova formula:



Funkcijo aproksimiramo s polinomom II. stopnje.

Tri nam označijo interval / pas 3. točke

Polinom II. stopnje:

$$p(x) = a + bx + cx^2$$

Izvrednotimo ga v teh znanih točkah:

$$p(x_1) = a + bx_1 + cx_1^2 = f(x_1)$$
$$p(x_2) = a + b(x_1+h) + c(x_1+h)^2 = f(x_2)$$
$$p(x_3) = a + b(x_1+2h) + c(x_1+2h)^2 = f(x_3)$$

$$b \cdot h + c(2hx_1 + h^2) = f(x_2) - f(x_1)$$
$$b \cdot h + c(2hx_1 + 3h^2) = f(x_3) - f(x_2)$$

$$c\,2h^2 = f(x_3) - 2f(x_2) + f(x_1)$$

$$c = \frac{f(x_3) - 2f(x_2) + f(x_1)}{2h^2}$$

$$p(x) = a + bx + cx^2$$

$$p(x) = a + b(x - x_1) + c(x - x_1)^2$$

$$p(x_1) = a = f(x_1)$$

$$p(x_2) = a + b(\overbrace{x_2 - x_1}^{h}) + c(\overbrace{x_2 - x_1}^{h^2})^2 = f(x_2)$$

$$p(x_3) = a + b(\underbrace{x_3 - x_1}_{2h}) + c(\underbrace{x_3 - x_1}_{(2h)^2})^2 = f(x_3)$$

$$\left.\begin{aligned} bh + ch^2 &= f(x_2) - f(x_1) \quad / \cdot 2 \\ 2bh + 4ch^2 &= f(x_3) - f(x_1) \end{aligned}\right] -$$

$$2ch^2 = f(x_3) - 2f(x_2) + f(x_1)$$

$$c = \frac{f(x_3) - 2f(x_2) + f(x_1)}{2h^2}$$

$$b = \frac{f(x_2) - f(x_1)}{h} - ch$$

$$= \frac{-f(x_3) - 3f(x_1) + 4f(x_2)}{2h}$$

$$p(x) = f(x_1) - \frac{f(x_3) + 3f(x_1) - 4f(x_2)}{2h}(x - x_1) + \frac{f(x_3) - 2f(x_2) + f(x_1)}{2h^2}(x - x_1)^2$$

$$\int_{x_1}^{x_3} f(x)\,dx \doteq \int_{x_1}^{x_3} p(x)\,dx = \left| \begin{array}{l} \text{Substitucija} \\ \tilde{x} = x - x_1 \end{array} \right| = \int_0^{2h} p(\tilde{x})\,d\tilde{x}$$

$$= f(x_1)\cdot 2h - \frac{f(x_3)+3f(x_1)-4f(x_2)}{2\cancel{h}}\cdot 2h^{\cancel{2}} +$$

$$+ \frac{f(x_3)-2f(x_2)+f(x_1)}{\cancel{2h^2}}\cdot\frac{{}^{4}\cancel{8}h^{\cancel{3}}}{3} =$$

$$= 2h\left[\frac{1}{6}f(x_1) + \frac{4}{6}f(x_2) + \frac{1}{6}f(x_3)\right] =$$

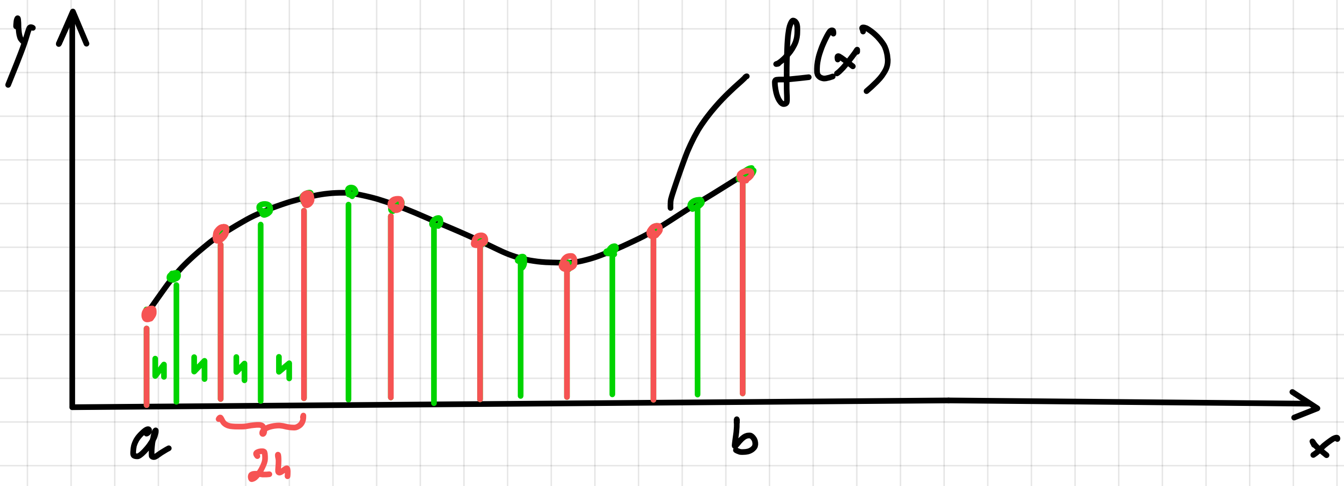
$$= \frac{2h}{6}\left[f(x_1) + 4f(x_2) + f(x_3)\right]$$

$$= h\left[\frac{1}{3}f(x_1) + \frac{4}{3}f(x_2) + \frac{1}{3}f(x_3)\right] + \mathcal{O}\left(h^5 f^{(IV)}(\xi)\right)$$

Opomba: ker je interval dolg $2h$, se koeficienti seštejejo v 2 in ne v 1, kot pri trapezni metodi.

Komentar: Glede na to, da metoda sloni na interpolaciji s kvadratno funkcijo skozi tri točke, bi pričakovali, da je eksaktna za $f(x) = x^2$. Pa se izkaže, da je metoda eksaktna za $f(x) = x^3$ zaradi simetričnosti formule.

Sestavljena Simpsonova formula:



Da dobimo integral na intervalu $[a, b]$, tega razdelimo na $N = 2M$ enakih intervalov velikosti:

$$\boxed{h = \frac{b-a}{2M}}$$

Celoten integral izračunamo kot vsoto integralov na intervalih dolžine $2h$.

$$\boxed{\begin{aligned}\int_a^b f(x)\,dx \simeq \int_{x_1=a}^{x_N=b} p(x)\,dx = \frac{h}{3}\Big(f(x_1) + 4f(x_2) + 2f(x_3) + 4f(x_4) + \\ + 2f(x_5) + 4f(x_6) + \ldots 4f(x_{N-1}) + f(x_N)\Big) \\ + \mathcal{O}\left(\frac{(b-a)^5 f^{(IV)}(\xi)}{N^4}\right)\end{aligned}}$$

Simpsonovo pravilo!

4.) Bodejeva formula: (v resnici Boolejeva f.)

Metoda sloni na interpolaciji skozi 5 točk s polinomom 4. reda. Metoda je zanimiva zato, ker je – tako kot Simpsonova metoda – za red bolj natančna.

$$\int_{x_0}^{x_4} f(x)\,dx \simeq h\left(\frac{14}{45} f(x_0) + \frac{64}{45} f(x_1) + \frac{24}{45} f(x_3) + \frac{64}{45} f(x_4) + \frac{14}{45} f(x_5)\right) + \mathcal{O}\left(h^7 f^{(VI)}(\xi)\right)$$

Vprašanje: Ali je bolje uporabljati metodo višjega reda, ali povečati število korakov?

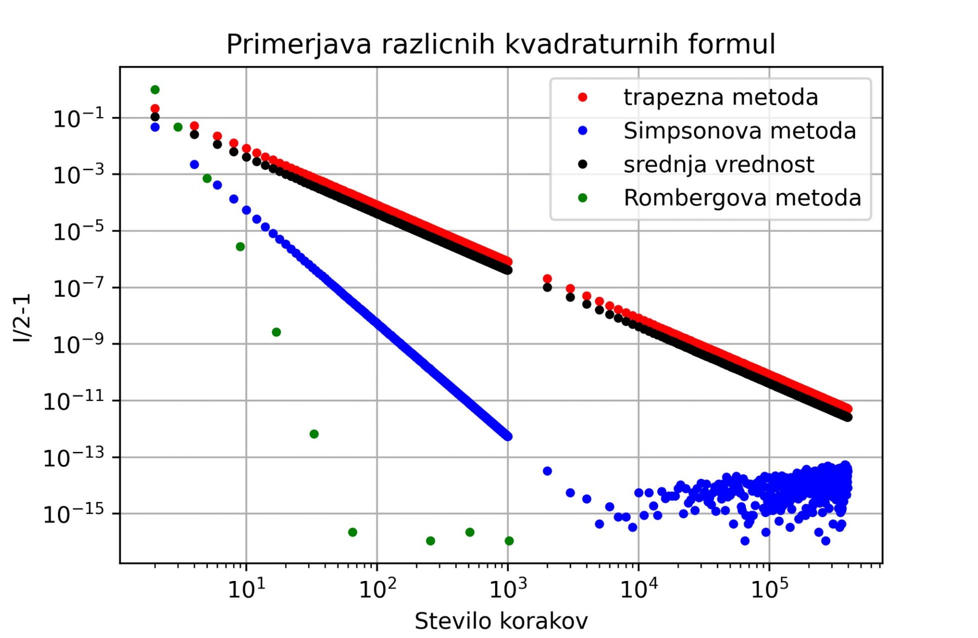
Visok red metode še ne pomeni, da je metoda tudi najbolj točna!

Doseganje željene natančnosti:

Število točk, v katerih moramo izvrednotiti funkcijo, da bomo dosegli željeno natančnost ne poznamo vnaprej. Integracijo je potrebno zasnovati tako, da povečujemo število točk in pogledamo, kaj se dogaja z vrednostjo integrala, ter odnehamo ko je

$$\boxed{|I_{n+1} - I_n| < \varepsilon}$$

Opomba: Zdrava mera je tu $\varepsilon \sim 10^{-9}$. Če želimo iti čez tega, se nam lahko zgodi, da bo zaokrožitvena napaka, ki se nabira med opravljenim računanjem narasla do te mere, da metoda ne bo nikoli konvergirala.



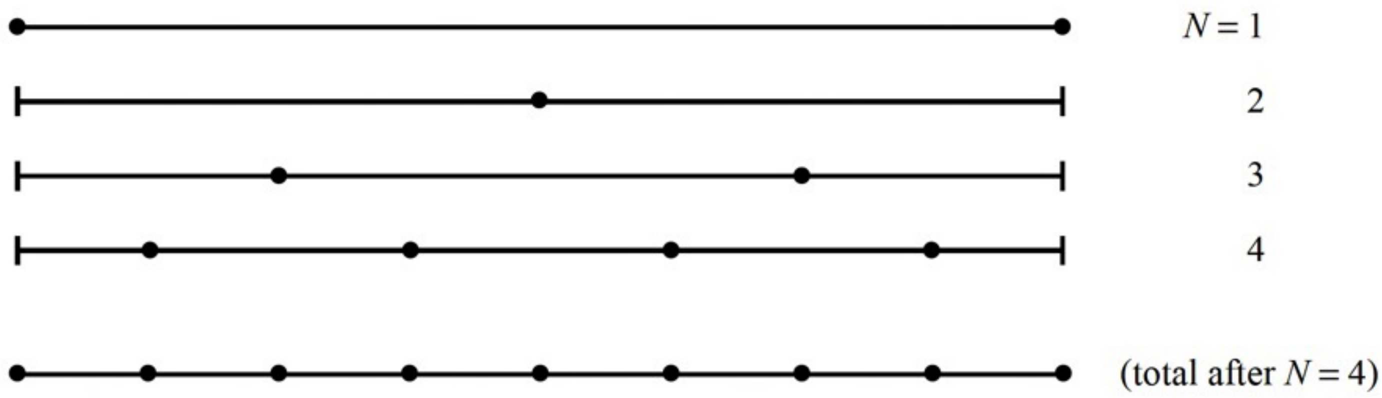


Figure 4.2.1. Sequential calls to the routine `trapzd` incorporate the information from previous calls and evaluate the integrand only at those new points necessary to refine the grid. The bottom line shows the totality of function evaluations after the fourth call. The routine `qsimp`, by weighting the intermediate results, transforms the trapezoid rule into Simpson's rule with essentially no additional overhead.

<u>Napotek:</u> Pri računanju naslednje iteracije integrala I_{m+1}, ne računamo vsega od začetka, pač pa integral računamo tako, da rekurzivno razpolavljamo podintervale in sproti dodajamo točke prejšnji iteraciji. Dobre metode imajo to že vgrajeno.

NRC: trapzd.

Praktična ocena napake integracijske metode:

(npr. Simpsonove)

Za dodatne podrobnosti glej: Širca, RMF, str. 622.

- Integral izračunajmo v $(N+1)$ točkah:

$$I = I_N + \mathcal{O}\left(\frac{(b-a)^5 \cdot f''(\xi_1)}{N^4}\right)$$

$$= I_N + C \cdot \frac{(b-a)^5 f''(\xi_1)}{N^4} = R_N$$

- Integral izračunajmo v $(2N+1)$ točkah:

$$I = I_{2N} + R_{2N}$$

$$= I_{2N} + C \cdot \frac{(b-a)^5 \cdot f''(\xi_2)}{16 N^4}$$

Če privzamemo, da je $f''(\xi_2) \simeq f''(\xi_1)$, potem:

$$\left|\frac{R_{2N}}{R_N}\right| = \frac{1}{16}$$

Hkrati velja: $|I_{2N} - I_N| = |R_N - R_{2N}|$:

$$|R_N - R_{2N}| = \frac{15}{16}|R_N| = |I_{2N} - I_N|$$

$$\boxed{|R_N| = \frac{16}{15}|I_{2N} - I_N|}$$

Ocena napake in Richardsonova ekstrapolacija

- Za trapezno formulo smo ugotovili, da velja:

$$I(f) = \int_a^b f(x)\,dx = \underbrace{T_h}_{\text{Približek pri koraku } h} + \underbrace{R_h}_{\text{Napaka}} = T_h + c f''(\xi)\, \underline{h^2}$$

- Izračunajmo integral s korakoma h in $h/2$:

$$I(f) = T_h + R_h = T_{\frac{h}{2}} + R_{\frac{h}{2}}$$

$$R_h = c f''(\xi_1)\cdot h^2 \qquad R_{\frac{h}{2}} = c f''(\xi_2)\frac{h^2}{4}$$

- Če privzamemo, da je $f''(\xi_1) \doteq f''(\xi_2)$, ugotovimo:

$$\boxed{R_h = 4 R_{\frac{h}{2}}}$$

- Od tod pa sledi praktična ocena napake:

$$R_h - \frac{1}{4} R_h = T_{\frac{h}{2}} - T_h$$

$$\boxed{R_h = \frac{4}{3}\left|T_{\frac{h}{2}} - T_h\right|}$$

• Še bolj pa se splača postopati drugače:

$$T_h + 4R_{\frac{h}{2}} = T_{\frac{h}{2}} + R_{\frac{h}{2}}$$

$$R_{\frac{h}{2}} = \frac{T_{\frac{h}{2}} - T_h}{3}$$

← To upoštevamo v integralu.

$$I(f) = T_{\frac{h}{2}} + R_{\frac{h}{2}} = T_{\frac{h}{2}} + \frac{T_{\frac{h}{2}} - T_h}{3} = \frac{4T_{h/2} - T_h}{3} = T_{\frac{h}{2}}^{(1)}$$

Z zaporedno ekstrapolacijo smo dosegli višjo natančnost. To se vedno splača narediti. Sedaj je napaka $O(h^4)$.

Rombergova metoda upošteva razvoj napake do še višjih redov in lahko potem nadaljujemo ekstrapolacijo gradi še naprej. Dobimo shemo

	$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$
h	$T_h^{(0)}$			
$\frac{h}{2}$	$T_{h/2}^{(0)}$	$T_{h/2}^{(1)}$		
$\frac{h}{4}$	$T_{h/4}^{(0)}$	$T_{h/4}^{(1)}$	$T_{h/4}^{(2)}$	
$\frac{h}{8}$	$T_{h/8}^{(0)}$	$T_{h/8}^{(1)}$	$T_{h/8}^{(2)}$	$T_{h/8}^{(3)}$

Komentarji:

1.) Problem določitve integrala $I(b)=\int_a^b f(x)\,dx$ je ekvivalenten reševanju diferencialne enačbe:
$\frac{dI}{dx} = f(x)$ z robnim pogojem $I(a)=0$.
To bomo še srečali!

2.) Kaj storiti, ko je integral zbran v enem ali več ostrih vrhovih ali pa če imamo funkcijo, ki gre na osi x preko več velikostnih redov.
$\Rightarrow$ Tedaj bi prav prišel adaptivni korak. Torej, da so točke bolj goste, ko se funkcija hitro spreminja, in bolj redke, kjer se funkcija spreminja le počasi. Z adaptivnimi metodami se bomo ukvarjali pri reševanju diferencialnih enačb. V takšnem primeru problem integracije prevedemo na reševanje diferencialne enačbe.

3.) Kvadraturne formule se da uporabiti za računanje integralov v več dimenzijah. Vendar mreža točk narašča kot N^d, natančnost pa le počasi narašča. Bolj se splača uporabiti integracijo Monte-Carlo.

Primer Trapezne metode v 2D.

$$\int_a^b \int_c^d f(x,y)\,dy\,dx = \frac{h\cdot k}{4}\sum_{i=0}^{n}\sum_{j=0}^{m} A_{ij}\, f(x_i,y_j) + O(h^2+k^2)$$

Za koeficiente A_{ij} velja: $\underline{\underline{A}} = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 4 & 4 & 2 \\ 2 & 4 & 4 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix}$

4.) Pri N.-C. pravilih smo spreminjali utežne koeficiente točk na ekvidistantnih legah. Kaj če bi dovolili, da točke niso enako oddaljene druga od druge? To je ideja Gaussovih kvadraturnih formul. V tem primeru imamo na voljo 2× več prostostnih stopenj (uteži + lege) s čimer lahko dosežemo enako natančnost z 2× manj točkami – To dobro velja za funkcije, ki se jih da dobro aproksimirati s polinomi.

↓ Samo v primeru polinomov visok red metode pomeni tudi visoko natančnost metode.

5.) Še vrsta drugih bolj "naprednih metod"! Glej NRC. poglavje 4.