

Numerične metode 2

Zapiski s predavanj prof. Marjetke Knez

Domen Vogrin

Ljubljana, pomlad 2023

Kazalo

1	Teorija aproksimacije	1
1.1	Aproksimacija funkcij	1
1.1.1	Splošen optimalni aproksimacijski problem	3
1.2	Aproksimacija po metodi najmanjših kvadratov (MNK)	4
1.2.1	Normalni oziroma Gramov sistem enačb	7
1.2.2	Povezava s predločenimi sistemi enačb	10
1.3	Enakomerna aproksimacija zveznih funkcij s polinomi	12
2	Interpolacija	17
2.1	Polinomska interpolacija	17
2.1.1	Lagrangeva oblika zapisa interpolacijskega polinoma . . .	18
2.1.2	Newtonova oblika zapisa interpolacijskega polinoma . . .	22
2.1.2.1	Trikotna shema	25
2.1.2.2	Posplošitev deljene difference	27
2.1.2.3	Posplošitev rekurzivne formule	27
2.1.2.4	Odsekoma polinomske funkcije (zlepki)	29
3	Numerično odvajanje	31
3.1	Ideja za izpeljavo aproksimacijskih formul	31
4	Numerična integracija	34
4.1	Nekaj osnovnih Newton-Cotesovih (N-C') pravil	35
4.1.1	Alternativna izpeljava integracijskih pravil - metoda nedoločenih koeficientov	39
4.2	Numerične napake pri integracijskih pravilih	40
4.3	Sestavljena integracijska pravila	40
4.3.1	Sestavljeno trapezno pravilo	41
4.3.2	Sestavljeno Simpsonovo pravilo	41
4.3.3	Ocena napake in Richardsonova ekstrapolacija	42
4.4	Gaussova integracijska pravila	43
4.5	Integrali v več dimenzijah	47
4.5.1	Dvojni integral	47
4.5.2	Tenzorska pravila	47
4.5.3	Simpsonovo pravilo	48
4.5.4	Metoda Monte-Carlo	48
5	Numerično reševanje navadnih diferencialnih enačb	49
5.0.1	Začetni problem reda 1	49
5.0.2	Začetni problem reda p	50

5.0.3	Robni problemi	50
5.1	Globalna in lokalna napaka	54
5.2	Runge-Kutta metode	56
5.2.1	Primeri R-K metod	57
5.2.2	Sistemi diferencialnih enačb 1. reda	57
5.2.3	Reševanje diferencialnih enačb višjega reda	58
5.3	Veččlenske metode	60
6	Numerično računanje lastnih vrednosti	61
6.1	Schurova forma	63
6.2	Metode za računanje lastnih vrednosti	64
6.2.1	Potenčna metoda	64
6.2.2	Inverzna iteracija (inverzna potenčna metoda)	66
6.2.3	Ortogonalna iteracija	67
6.2.4	QR iteracija (osnovna različica)	68
6.2.4.1	Householderjeva zrcaljenja	69
6.2.4.2	Premiki	70

1 Teorija aproksimacije

1.1 Aproksimacija funkcij

Denimo, da imamo podano funkcijo f . Radi bi jo aproksimirali s kakšno 'preprostejšo' funkcijo \tilde{f} , ki bi bila lažje izračunljiva, bi se jo dalo enostavno odvajati, integrirati ...

Primer.

$$\sin(x) \sim x - \frac{x^3}{3!} + \frac{x^5}{5!}$$

Ključna vprašanja, ki se nam postavijo, so:

- V kakšni množici/podprostoru naj iščemo aproksimant \tilde{f} ?
- V čem naj si bo \tilde{f} podobna/sorodna z f ?
- Ali \tilde{f} obstaja (v množici, kjer jo iščemo)?
- Če obstaja, ali je določen enolično?
- Kako konstruirati aproksimant \tilde{f} ?
- Kako dobro nadomestilo za f je izračunan \tilde{f} ?

V splošnem aproksimacijski problem formaliramo takole: z X označimo vektorski prostor, katerega elemente želimo aproksimirati, $S \subseteq X$ naj označuje podprostor/podmnožico v X , v katerem iščemo aproksimante. Aproksimacijska shema je operator

$$\mathcal{A}: X \rightarrow S$$

ki vsakemu elementu $f \in X$ priredi aproksimacijski element (aproksimant)

$$\tilde{f} = \mathcal{A}f \in S$$

Primer. Vektorski prostori:

- $X = \mathcal{C}([a, b])$, $X = \mathcal{C}^k([a, b])$
- $X = \ell^2_\rho([a, b]) = \{f: [a, b] \rightarrow \mathbb{R} \mid \int_a^b f^2(x)\rho(x)dx < \infty\}$,
pri čemer je ρ **pozitivna utež**: $\rho(x) > 0$ za vsak $x \in [a, b]$
- $X = \mathbb{R}^n$

Primer. Podprostori, v katerih iščemo aproksimante:

- $S = \mathbb{P}_n = \text{Lin}\{1, x, x^2, \dots, x^n\}$ polinomi stopnje $\leq n$:

$$S = \left\{ \sum_{i=0}^n a_i x^i; a_i \in \mathbb{R} \right\}$$

- **trigonometrični polinomi**

$$S = \text{Lin}\{1, \sin x, \cos x, \sin 2x, \cos 2x, \dots, \sin nx, \cos nx\}$$

- podprostor racionalnih funkcij, odsekoma polinomskih funkcij

Da bomo lahko definirali aproksimacijski problem in tudi ocenili napako aproksimacije, potrebujemo **normo**. Najbolj znane norme na prostoru funkcij so naslednje:

- neskončna norma ($\|f\|_\infty$)

$$f \in \mathcal{C}([a, b]), \|f\|_{\infty, [a, b]} = \max_{x \in [a, b]} |f(x)|$$

Za izračun numeričnega približka za neskončno normo na intervalu $[a, b]$ izberemo dovolj gosto zaporedje točk:

$$a \leq x_0 < x_1 < \dots < x_n \leq b, \mathbf{x} = (x_i)_{i=0}^N$$

in izračunamo

$$\|f\|_{\infty, \mathbf{x}} = \max_{i=0, \dots, N} |f(x_i)|$$

- druga norma ($\|\cdot\|_2$) - norma, porojena iz skalarne produkta. Naj bo vektorski prostor X opremljen s skalarnim produktom $\langle \cdot, \cdot \rangle$. Potem je

$$\|f\|_2 = \sqrt{\langle f, f \rangle}, f \in X$$

Primeri skalarnih produktov:

$$\cdot \quad \langle f, g \rangle = \int_a^b f(x)g(x)\rho(x)dx, f, g \in \mathcal{L}^2([a, b])$$

$$\cdot \quad \|f\|_2 = \sqrt{\int_a^b f^2(x)\rho(x)dx}$$

Za $\rho(x) \equiv 1$ to imenujemo **standardni skalarni produkt**

- diskretni semi-skalarni produkt

$$\mathbf{x} = (x_i)_{i=0}^N, a \leq x_0 < x_1 < \dots < x_n \leq b$$

$$\langle f, g \rangle = \sum_{i=0}^N f(x_i)g(x_i)\rho(x_i)$$

Če ga še delimo z dolžino intervala, dobimo približek za prejšnjega.

$$\|f\|_{2,\mathbf{x}} = \sqrt{\sum_{i=0}^N f^2(x_i)\rho(x_i)}$$

Za določanje aproksimanta \tilde{f} ločimo dva primera:

1. Optimalni aproksimacijski problemi
2. Interpolacija

1.1.1 Splošen optimalni aproksimacijski problem

Naj bo X vektorski prostor z normo $\|\cdot\|$, $S \subseteq X$. Za $f \in X$ iščemo $\tilde{f} \in S$, da velja

$$\|f - \tilde{f}\| = \inf_{s \in S} \|f - s\| = \text{dist}(f, S)$$

Torej, izmed možnih približkov izberemo najboljšega.

Pri tem predmetu si bomo ogledali:

1. aproksimacijo po metodi najmanjših kvadratov
(za normo izberemo drugo normo - normo iz skalarnega produkta)
2. enakomerna polinomska aproksimacija

$$X = C([a, b]), S = \mathbb{P}_n, \|\cdot\|_\infty$$

Polinomi so zelo uporabni pri aproksimaciji funkcij, saj so gosti v prostoru zveznih funkcij.

Izrek 1.1. (Weierstrassov izrek) Naj bo $f \in \mathcal{C}([a, b])$. Potem za vsak $\varepsilon > 0$ obstaja polinom p , da je $\|f - p\|_{\infty, [a, b]} < \varepsilon$. Drugače povedano:

$$\text{dist}(f, \mathbb{P}_n) \xrightarrow{n \rightarrow \infty} 0$$

Dokaz. (konstruktivni - ideja) Naj bo $[a, b] = [0, 1]$. Za $f \in \mathcal{C}([0, 1])$ definiramo t.i. **Bernsteinov polinom**:

$$\mathcal{B}_n f(x) = \sum_{i=0}^n f\left(\frac{i}{n}\right) B_i^n(x)$$

kjer je $B_i^n(x)$ **Bernsteinov bazni polinom**:

$$B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i}, \quad i = 0, 1, \dots, n$$

Da se pokazati, da gre $\|f - \mathcal{B}_n f\|_{\infty, [a, b]} \rightarrow 0$, ko gre $n \rightarrow \infty$. ■

Bernsteinov aproksimacijski polinom nam poda en možen način aproksimacije funkcije f (na $[0, 1]$).

Bernsteinov aproksimacijski operator:

$$\begin{aligned} \mathcal{B}_n : \mathcal{C}([a, b]) &\rightarrow \mathbb{P}_n \\ f &\mapsto \mathcal{B}_n f \\ \mathcal{B}_n f(x) &= \sum_{i=0}^n f\left(a + \frac{i}{n}(b-a)\right) B_i^n\left(\frac{x-a}{b-a}\right) \end{aligned}$$

Po Weierstrassovem izreku imamo zagotovljeno konvergenco v neskončni normi, žal pa je konvergenca zelo počasna.

1.2 Aproksimacija po metodi najmanjših kvadratov (MNK)

Sodi pod optimalne aproksimacijske probleme.

Naj bo X normiran vektorski prostor nad \mathbb{R} s skalarnim produktom $\langle \cdot, \cdot \rangle$ in naj bo $\|\cdot\|_2 = \sqrt{\langle \cdot, \cdot \rangle}$. $S \subseteq X$ naj bo končno dimenzionalen podprostor v X , $S = \text{Lin}\{\varphi_1, \varphi_2, \dots, \varphi_n\}$, $\dim S = n$. Za izbran $f \in X$ iščemo $f^* \in S$, da bo veljalo

$$\|f - f^*\|_2 = \min_{s \in S} \|f - s\|_2$$

f^* naj bo element najbližje aproksimacije (ENA) po MNK za $f \in X$.

Izrek 1.2. Naj bo $S \subseteq X$ končno dimenzionalen podprostor. Element $f^* \in S$ je element najbližje aproksimacije po MNK za $f \in X$ natanko takrat, ko je

$$f - f^* \perp S$$

oziroma

$$\langle f - f^*, S \rangle = 0$$

Dokaz.

(\Leftarrow) Predpostavimo, da je $f - f^* \perp S$. Dokazati moramo, da je

$$\|f - f^*\|_2 = \min_{s \in S} \|f - s\|_2$$

Izberimo poljuben $s \in S$.

$$\begin{aligned} \|f - s\|_2^2 &= \|f - f^* + f^* - s\|_2^2 \\ &= \langle (f - f^*) + (f^* - s), (f - f^*) + (f^* - s) \rangle \\ &= \|f - f^*\|_2^2 + 2 \cdot \langle f^* - s, f - f^* \rangle + \|f^* - s\|_2^2 \\ &\geq \|f - f^*\|_2^2 \end{aligned} \tag{1}$$

Neenakost 1 velja, saj zato, ker velja tako $f^* \in S$ kot $s \in S$, velja tudi $(f^* - s) \in S$, torej veljata tudi enakost $\langle f^* - s, f - f^* \rangle = 0$ in neenakost $\|f^* - s\|_2 \geq 0$.

(\Rightarrow) Predpostavimo, da je f^* ENA po MNK. Dokazati želimo

$$f - f^* \perp S$$

$\forall s \in S$ in $\forall \lambda > 0$ velja

$$\begin{aligned} \|f - f^*\|_2^2 &\leq \|f - (f^* - \lambda s)\|_2^2 \\ &= \langle f - f^* + \lambda s, f - f^* + \lambda s \rangle \\ &= \|f - f^*\|_2^2 + 2 \cdot \langle f - f^*, \lambda s \rangle + \lambda^2 \|s\|_2^2 \end{aligned}$$

$$\begin{aligned} 0 &\leq 2\langle f - f^*, \lambda s \rangle + \lambda^2 \|s\|_2^2 \\ 0 &\leq \lambda(2\langle f - f^*, s \rangle + \lambda \|s\|_2^2) \end{aligned} \tag{2}$$

$$0 \leq \langle f - f^*, s \rangle + \lambda \|s\|_2^2 \tag{3}$$

pri čemer iz 2 na 3 pridemo preko začetnega izbora za $\lambda > 0$. Ker lahko λ vzamemo tako majhno, da velikost člena $2\langle f - f^*, s \rangle$ prevlada nad $\lambda \|s\|_2^2$, vidimo, da mora biti $0 \leq \langle f - f^*, s \rangle$. Če sedaj v S izberemo element $-s$, potem po istem sklepu velja, da mora biti $0 \leq \langle f - f^*, -s \rangle$ oziroma $\langle f - f^*, s \rangle \leq 0$. Sledi, da mora biti

$$\langle f - f^*, s \rangle = 0$$

■

Iz izreka sledi konstrukcija.

Izberemo $f \in X$. Naj bodo $\varphi_1, \varphi_2, \dots, \varphi_n$ baza za $S \subseteq X$:

$$S = \text{Lin}\{\varphi_1, \varphi_2, \dots, \varphi_n\}$$

Iščemo $f^* \in S$ ENA po MNK.

$$f^* = \sum_{j=1}^n \alpha_j \varphi_j$$

kjer so $(\alpha_j)_{j=1}^n$ neznani koeficienti. Iz izreka sledi, da mora biti $f - f^* \perp S$. To bo res, ko bo

$$f - f^* \perp \varphi_i, i \in [n]$$

$$\begin{aligned} 0 &= \langle f - f^*, \varphi_i \rangle \\ &= \langle f - \sum_{j=1}^n \alpha_j \varphi_j, \varphi_i \rangle \\ &= \langle f, \varphi_i \rangle - \sum_{j=1}^n \alpha_j \langle \varphi_j, \varphi_i \rangle \end{aligned}$$

Za vsak i tako dobimo enačbo

$$\sum_{j=1}^n \alpha_j \langle \varphi_j, \varphi_i \rangle = \langle f, \varphi_i \rangle$$

iz česar skupaj dobimo sistem linearnih enačb. Če zgornje zapišemo po vektorjih, dobimo

$$\begin{bmatrix} \langle \varphi_1, \varphi_i \rangle & \langle \varphi_2, \varphi_i \rangle & \cdots & \langle \varphi_n, \varphi_i \rangle \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \langle f, \varphi_i \rangle, i \in [n]$$

V matrični obliki:

$$\begin{bmatrix} \langle \varphi_1, \varphi_1 \rangle & \langle \varphi_2, \varphi_1 \rangle & \cdots & \langle \varphi_n, \varphi_1 \rangle \\ \langle \varphi_1, \varphi_2 \rangle & \langle \varphi_2, \varphi_2 \rangle & \cdots & \langle \varphi_n, \varphi_2 \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \varphi_1, \varphi_n \rangle & \langle \varphi_2, \varphi_n \rangle & \cdots & \langle \varphi_n, \varphi_n \rangle \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \langle f, \varphi_1 \rangle \\ \langle f, \varphi_2 \rangle \\ \vdots \\ \langle f, \varphi_n \rangle \end{bmatrix}$$

1.2.1 Normalni oziroma Gramov sistem enačb

Gramova matrika G

$$G = (\langle \varphi_j, \varphi_i \rangle)_{i,j=1}^n$$

je **simetrična** matrika. Gramova matrika je tudi pozitivno definitna. To dokažemo tako, da izberemo $x \in \mathbb{R}^n, x \neq 0$.

$$\begin{aligned} x^T G x &= \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \begin{bmatrix} \sum_{j=1}^n x_j \langle \varphi_j, \varphi_1 \rangle \\ \vdots \\ \sum_{j=1}^n x_j \langle \varphi_j, \varphi_n \rangle \end{bmatrix} \\ &= \sum_{i=1}^n x_i \sum_{j=1}^n x_j \langle \varphi_j, \varphi_i \rangle \\ &= \sum_{i=1}^n \sum_{j=1}^n \langle x_j \varphi_j, x_i \varphi_i \rangle \\ &= \left\langle \sum_{j=1}^n x_j \varphi_j, \sum_{i=1}^n x_i \varphi_i \right\rangle \\ &= \left\| \sum_{i=1}^n x_i \varphi_i \right\|_2^2 \\ &> 0 \end{aligned} \tag{4}$$

Neenačaj 4 je strog, saj velja

$$\sum_{i=1}^n x_i \varphi_i \neq 0$$

To je res zato, ker je $x_i > 0$ in ker je

$$\varphi_i \in \text{Lin}\{\varphi_1, \varphi_2, \dots, \varphi_n\}$$

kar je baza za S . Dobljeni sistem enačb lahko rešimo z razcepom Choleskega.

Primer. Naj bo $f(x) = \sin(x)$, $\langle f, g \rangle = \int_0^\pi f(x)g(x)dx$. Aproksimiraj f po MNK v podprostoru \mathbb{P}_1 .

Rešitev: Definirajmo X in S

$$X = \mathcal{C}([0, \pi])(X = \mathcal{L}^2([0, \pi]))$$

$$S = \mathbb{P}_1 = \text{Lin}\{1, x\}, \varphi_1(x) = 1, \varphi_2(x) = x$$

Zdaj definiramo f^*

$$f^*(x) = \alpha_1 \varphi_1(x) + \alpha_2 \varphi_2(x)$$

Imamo Gramovo matriko G

$$\begin{bmatrix} \langle \varphi_1, \varphi_1 \rangle & \langle \varphi_2, \varphi_1 \rangle \\ \langle \varphi_1, \varphi_2 \rangle & \langle \varphi_2, \varphi_2 \rangle \end{bmatrix} = \begin{bmatrix} \pi & \frac{\pi^2}{2} \\ \frac{\pi^2}{2} & \frac{\pi^3}{3} \end{bmatrix}$$

$$\text{desna stran} = \begin{bmatrix} \langle f, \varphi_1 \rangle \\ \langle f, \varphi_2 \rangle \end{bmatrix} = \begin{bmatrix} 2 \\ \pi \end{bmatrix}$$

Zgornji izračuni prihajajo iz postopkov

$$\langle \varphi_1, \varphi_1 \rangle = \int_0^\pi dx = \pi$$

$$\langle \varphi_1, \varphi_2 \rangle = \int_0^\pi x dx = \frac{\pi^2}{2}$$

$$\langle \varphi_2, \varphi_2 \rangle = \int_0^\pi x^2 dx = \frac{\pi^3}{3}$$

in

$$\langle f, \varphi_1 \rangle = \int_0^\pi \sin x dx = -\cos x \Big|_0^\pi = 2$$

$$\langle f, \varphi_2 \rangle = \int_0^\pi x \cdot \sin x dx = \dots = \pi$$

Dobimo:

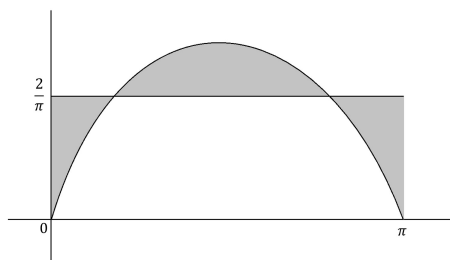
$$\begin{bmatrix} \pi & \frac{\pi^2}{2} \\ \frac{\pi^2}{2} & \frac{\pi^3}{3} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 2 \\ \pi \end{bmatrix}$$

Ko poračunamo sistem enačb, dobimo

$$\alpha = \begin{bmatrix} \frac{2}{\pi} \\ 0 \end{bmatrix}$$

Geometrijska interpretacija rešitve:

$$\min_{p \in \mathbb{P}_1} \|f - p\|_2 = \min_{p \in \mathbb{P}_1} \sqrt{\int_0^\pi (\sin x - p(x))^2 dx}$$



Slika 1: Z MNK želimo minimizirati ploščino sivega območja.

Primer. Točke $(1, 2), (2, 3), (3, 5), (4, 8)$ aproksimiraj po MNK s premico.

Rešitev: $S = \mathbb{P}_1 = \text{Lin}\{1, x\}$

$$\langle f, g \rangle = \sum_{i=1}^4 f(x_i)g(x_i), \quad x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

f , ki jo aproksimiramo, je znana le v točkah \mathbf{x} .

Izračunamo:

$$\langle 1, 1 \rangle = \sum_{i=1}^4 1 \cdot 1 = 4$$

$$\langle 1, x \rangle = \sum_{i=1}^4 1 \cdot x_i = 10$$

$$\langle x, x \rangle = \sum_{i=1}^4 x_i^2 = 30$$

$$\langle f, 1 \rangle = \sum_{i=1}^4 y_i \cdot 1 = 18$$

$$\langle f, x \rangle = \sum_{i=1}^4 y_i x_i = 55$$

Dobimo sistem

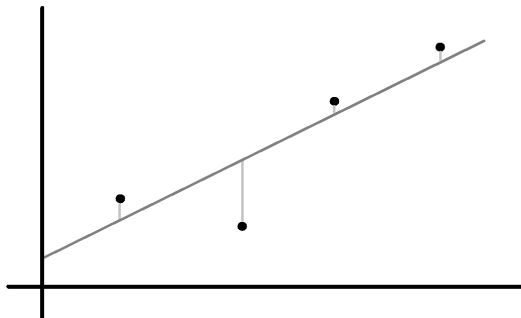
$$\begin{bmatrix} 4 & 10 \\ 10 & 30 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 18 \\ 55 \end{bmatrix}$$

iz katerega dobimo rezultat

$$\alpha = \begin{bmatrix} -\frac{1}{2} \\ 2 \end{bmatrix}$$

Geometrijska interpretacija rešitve:

$$\min_{p \in \mathbb{P}_1} \|f - p\|_2 = \sqrt{\sum_{i=1}^4 (y_i - p(x_i))^2}$$



Slika 2: Z MNK želimo najti temnosivo premico, ki minimizira svetlosive razdalje.

1.2.2 Povezava s predoločenimi sistemi enačb

$$Ax = b, A \in \mathbb{R}^{m \times n}, A = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}, b \in \mathbb{R}^m$$

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2 = \min_{z \in \text{Im} A} \|b - z\|_2$$

Aproksimiramo vektor $b \in \mathbb{R}^m$ ($X = \mathbb{R}^m$)

$$S = \text{Lin}\{a_1, a_2, \dots, a_n\} = \text{Im} A$$

$$b^* = \sum_{j=1}^n x_j a_j = Ax$$

$$\langle x, y \rangle = \sum_{i=1}^m x_i y_i = x^T y$$

$$\langle x, y \rangle = \sum_{i=1}^m x_i y_i:$$

$$G = (\langle a_j, a_i \rangle)_{i,j=1}^n = A^T A \quad (5)$$

$$\text{desna stran} = (\langle a_i, b \rangle)_{i=1}^n = A^T b \quad (6)$$

Primer. $X = \mathcal{C}([0, 1])$

$$S = P_{n-1} = \text{Lin}\{1, x, x^2, \dots, x^{n-1}\}$$

$$\langle f, g \rangle = \int_0^1 f(x)g(x)dx$$

$$\langle \varphi_i, \varphi_j \rangle = \int_0^1 x^{i-1}x^{j-1}dx = \int_0^1 x^{i+j-2}dx = \frac{1}{i+j-1}$$

$$G = (\frac{1}{i+j-1})_{i,j=1}^n$$

kjer je G Hilbertova matrika. Te so zelo občutljive.

Gramova matrika je lahko zelo občutljiva. Reševanju sistema linearnih enačb se izognemo, če v podprostoru S izberemo **ortonormirano bazo**:

$$\{\varphi_1, \varphi_2, \dots, \varphi_n\}$$

je ortonormirana baza, če

$$\varphi_i \perp \varphi_j \quad \forall i \neq j \text{ in } \|\varphi_i\|_2 = 1$$

V tem primeru je $G = I$ in $\alpha_i = \langle f, \varphi_i \rangle$, $f^* = \sum_{i=1}^n \langle f, \varphi_i \rangle \varphi_i$. Ortonormirano bazo izračunamo z **modificiranim Gram-Schmidtovim algoritmom**.

Algorithm 1 Modificiran Gram-Schmidtov algoritem

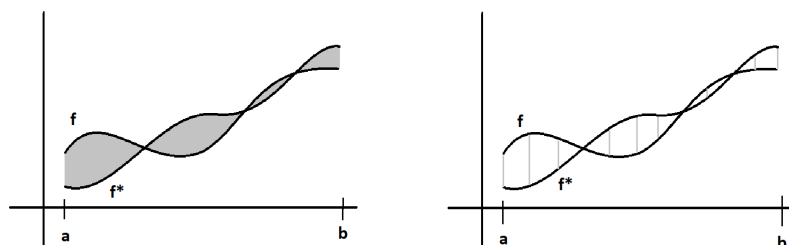
Input baza $\{\psi_1, \psi_2, \dots, \psi_n\}$

```

1: for  $i = 1 : n$  do
2:    $\varphi_i = \psi_i$ 
3: end for
4: for  $i = 1 : n$  do
5:    $\varphi_i = \frac{\varphi_i}{\|\varphi_i\|_2}$ 
6:   for  $j = i + 1 : n$  do
7:      $\varphi_j = \varphi_j - \langle \varphi_j, \varphi_i \rangle \varphi_i$ 
8:   end for
9: end for
```

Output ortonormirana baza $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$

$X \in \mathcal{C}([a, b])$



Slika 3: Druga norma, porojena iz zveznega (levo) in diskretnega (desno) skalarnega produkta.

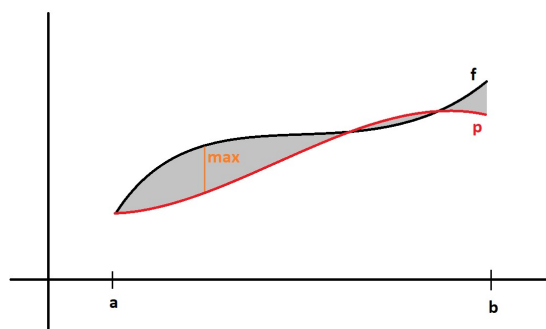
1.3 Enakomerna aproksimacija zveznih funkcij s polinomi

$$X = \mathcal{C}([a, b]), S = \mathbb{P}_n, \|\cdot\|_\infty$$

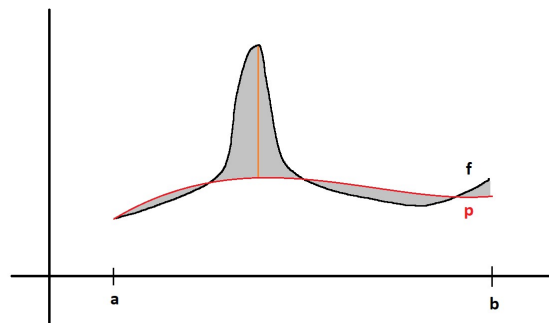
Problem: Za dano funkcijo $f \in \mathcal{C}([a, b])$ iščemo polinom $p^* \in \mathbb{P}_n$, za katerega velja

$$\|f - p^*\|_{\infty, [a, b]} = \min_{p \in \mathbb{P}_n} \|f - p\|_{\infty, [a, b]} = \min_{p \in \mathbb{P}_n} \max_{x \in [a, b]} |f(x) - p(x)|$$

p^* imenujemo **polinom najboljše enakomerne aproksimacije (PNEA)**. Problem je nelinearen.



Slika 4: Iščemo polinom p , ki se bo funkciji f najboljše prilegal po neskončni normi.



Slika 5: Prikaz razlike med drugo in neskončno normo. Medtem ko je druga norma tu relativno nizka, je neskončna zelo velika.

Nasledni izrek nam poda **zadostni pogoj**, da je nek polinom PNEA za neko funkcijo.

Izrek 1.3. Naj bo $f \in \mathcal{C}([a, b])$. Če je polinom $p \in \mathbb{P}_n$ tak, da **residual**

$$r = f - p \quad (7)$$

alternirajoče doseže svojo normo $\|p\|_{\infty, [a, b]}$ v vsaj $n+2$ različnih točkah $(x_i)_{i=0}^{n+1}$

$$a \leq x_0 < x_1 < \dots < x_{n+1} \leq b$$

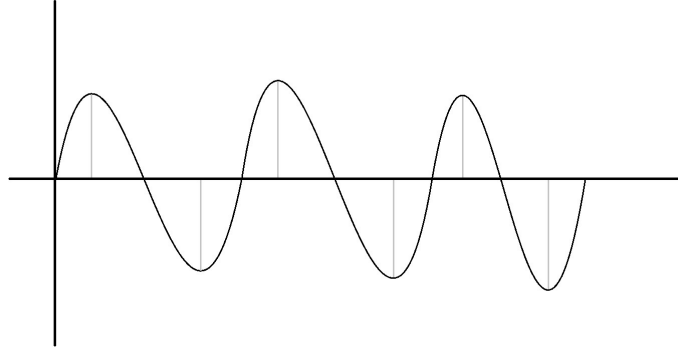
Potem je p polinom najboljše enakomerne aproksimacije za f na $[a, b]$.

Opomba. Kaj pomeni "alternirajoče doseže svojo normo"?

$$\|r\|_{\infty, [a, b]} = |r(x_i)| \quad \forall i \in [n]$$

in

$$r(x_i)r(x_{i+1}) < 0 \quad \forall i$$



Slika 6: Graf residuala.

Dokaz. Dokaz s protislovjem.

Recimo, da p ne bi bil PNEA za f . Tedaj bi obstajal nek drug polinom $q \in \mathbb{P}_n$, da bi veljalo

$$\begin{aligned}
 |f(x_i) - q(x_i)| &\leq \max_{x \in [a, b]} |f(x) - q(x)| \\
 &= \|f - q\|_{\infty, [a, b]} \\
 &< \|f - p\|_{\infty, [a, b]} \\
 &= |f(x_i) - p(x_i)| \quad \forall i = 0, 1, 2, \dots, n+1
 \end{aligned}$$

Torej za $\forall i$ velja:

$$|f(x_i) - q(x_i)| < |f(x_i) - p(x_i)|$$

To razvijemo v neenakosti

$$-sign(f(x_i) - p(x_i))(f(x_i) - p(x_i)) < f(x_i) - q(x_i)$$

in

$$f(x_i) - q(x_i) < sign(f(x_i) - p(x_i))(f(x_i) - p(x_i))$$

Če v neenakostih x_i spremenimo v x_{i+1} ter upoštevamo enakost

$$sign(f(x_{i+1}) - p(x_{i+1})) = -sign(f(x_i) - p(x_i))$$

dobimo neenakosti

$$sign(f(x_i) - p(x_i))(f(x_{i+1}) - p(x_{i+1})) < f(x_{i+1}) - q(x_{i+1})$$

in

$$f(x_{i+1}) - q(x_{i+1}) < -sign(f(x_i) - p(x_i))(f(x_{i+1}) - p(x_{i+1}))$$

Brez škode za splošnost (BŠS) lahko rečemo, da je $\text{sign}(f(x_i) - p(x_i)) = 1$. Potem je $f(x_i) - q(x_i) < f(x_i) - p(x_i)$, $p(x_i) - q(x_i) < 0$ in $f(x_{i+1}) - p(x_{i+1}) < f(x_{i+1}) - q(x_{i+1})$, torej $p(x_{i+1}) - q(x_{i+1}) > 0$.

Vidimo, da ima razlika $p - q$ ničlo na intervalu (x_i, x_{i+1}) za $i \in [n]$. Razlika $p - q$ je polinom stopnje n , ki ima $n + 1$ ničel. Torej mora biti $p \equiv q$. ■

Izkaže se, da je pogoj tudi potreben (torej da velja ekvivalenca), a je dokaz težek, zato ga bomo izpustili.

Iskanje/računanje PNEA se prevede na iskanje ustrezne množice točk $\{x_i, a \leq x_0 < x_1 < \dots < x_{n+1} \leq b\}$.

Definicija 1.4. Naj bo $E = \{x_i, a \leq x_0 < x_1 < \dots < x_{n+1} \leq b\}$. Definirajmo **minimaks** za f na E konstruirati

$$M_n(f, E) = \min_{p \in \mathbb{P}_n} \max_{x_i \in E} |f(x_i) - p(x_i)|$$

Polinom, pri katerem je ta minimum dosežen, imenujemo **polinom najboljše enakomerne aproksimacije za f na množici E** . Izračunamo ga tako, da rešimo naslednji sistem linearnih enačb: (brez izpeljave)

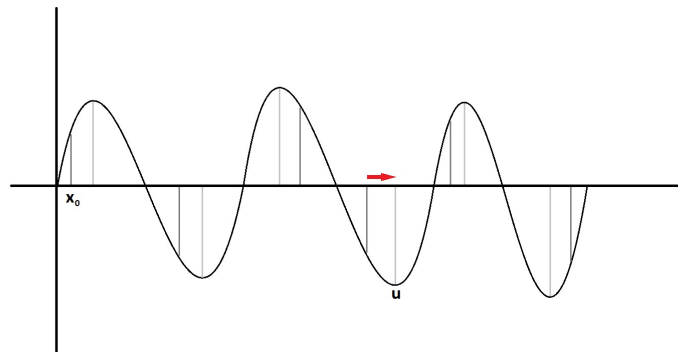
$$f(x_i) - p(x_i) = (-1)^i m, i \in [n + 1]$$

Imamo torej $n + 2$ enačb in $n + 2$ neznank ($n + 1$ v polinomu p in eno v m):

$$p(x) = \sum_{j=0}^n a_j x^j$$

ter koeficient m , za katerega velja

$$|m| = M_n(f, E)$$



Slika 7: Iskanje PNEA s pomočjo residuala.

V prvem koraku normalno pričnemo z ekvidistantnimi točkami. V drugem koraku poiščemo dejanski ekstrem in eno točko pomaknemo tja, pri čemer pazimo na alteriranje. Tako dobimo nov polinom.

2 Interpolacija

Problem: Podane imamo vrednosti izbrane funkcije f v $n+1$ paroma različnih točkah x_0, x_1, \dots, x_n na realni osi. Te točke bomo imenovali **interpolacijske točke**. Iščemo neko preprostejšo funkcijo g , ki zadošča pogojem

$$g(x_i) = f(x_i) \forall i \in [n]$$

g imenujemo **interpolacijska funkcija**. Za interpolacijske funkcije običajno izberemo polinome, odsekoma polinomske funkcije ...

Interpolacija se uporablja za

- aproksimacijo dane funkcije
- kadar funkcijo f poznamo le v točkah x_0, x_1, \dots, x_n , radi pa bi izračunali vrednost te funkcije tudi za x , ki ni ena izmed interpolacijskih točk.
- za izpeljavo formul za numerično integriranje, odvajanje, reševanje navadnih diferencialnih enačb (NDE) ...

2.1 Polinomska interpolacija

Za $f \in \mathcal{C}([a, b])$ in interpolacijske točke $a \leq x_0 < x_1 < \dots < x_m \leq b$ iščemo **polinom** p_i , ki zadošča enačbam

$$p(x_i) = f(x_i), i = 0, \dots, n$$

Enačb je $n + 1$. Da dobimo enako število enačb, moramo izbrati $p \in \mathbb{P}_n$.

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

Enačbe (za $i = 0, 1, \dots, n$)

$$a_0 + a_1x_i + a_2x_i^2 + \dots + a_nx_i^n$$

lahko zapišemo matrično

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$$

Matriko, ki jo uporabimo, imenujemo **vandermondova matrika**.

$$\det V(x_0, x_1, \dots, x_n) = \prod_{0 \leq i < j \leq n} (x_j - x_i)$$

Ker je Vandermondova matrika obrnljiva, sledi, da imamo enolično rešitev. Torej obstaja **enoličen** polinom stopnje n , ki interpolira $n + 1$ paroma različnih točk. Tak interpolacijski problem imenujemo **korenten** interpolacijski problem. Vandermondova matrika je primer **zelo občutljive** matrike. Poleg tega nimamo rešitve v **zaključeni obliki**. Spoznali bomo dva druga zapisa interpolacijskega polinoma:

- Lagrangeva oblika zapisa
- Newtonova oblika zapisa

2.1.1 Lagrangeva oblika zapisa interpolacijskega polinoma

Definiramo naslednje polinome:

$$\begin{aligned} \ell_{0,n}(x) &= \frac{(x - x_1)(x - x_2) \dots (x - x_n)}{(x_0 - x_1)(x_0 - x_2) \dots (x_0 - x_n)} \\ \ell_{1,n}(x) &= \frac{(x - x_0)(x - x_2) \dots (x - x_n)}{(x_1 - x_0)(x_1 - x_2) \dots (x_1 - x_n)} \\ &\vdots \\ \ell_{n,n}(x) &= \frac{(x - x_0)(x - x_1) \dots (x - x_{n-1})}{(x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1})} \end{aligned}$$

oziroma

$$\ell_{i,n}(x) = \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)}$$

za $i = 0, 1, \dots, n$. To imenujemo **Lagrangevi bazni polinomi**.

Velja:

$$\ell_{i,n}(x_j) = \delta_{i,j} = \begin{cases} 1 & ; i = j \\ 0 & ; i \neq j \end{cases}$$

Vsi ti polinomi so stopnje točno n .

Izrek 2.1. Polinomi $\ell_{i,n}$ za $i = 0, 1, \dots, n$ so baza za \mathbb{P}_n .

Dokaz. Dokazati moramo le, da so linearno neodvisni. Preveriti moramo, da je $\alpha_0 \ell_{0,n} + \alpha_1 \ell_{1,n} + \dots + \alpha_n \ell_{n,n} = 0 \Leftrightarrow \alpha_0 = \alpha_1 = \dots = \alpha_n = 0$.

(\Rightarrow)

$$\sum_{j=0}^n \alpha_j \ell_{j,n}(x) = 0 \quad \forall x$$

Vstavimo $x = x_i$ in dobimo

$$0 = \sum_{j=0}^n \alpha_j \ell_{j,n}(x_i) = \sum_{j=0}^n \alpha_j \delta_{i,j} = \alpha_i$$

(\Leftarrow) Očitno. ■

Iz dokaza izreka sledi, da lahko vsak polinom $p \in \mathbb{P}_n$ zapišemo kot linearno kombinacijo

$$p(x) = \sum_{j=0}^n c_j \ell_{j,n}(x)$$

za $c_j \in \mathbb{R}$.

Kako izbrati koeficiente, da bo polinom interpolacijski oziroma da bo zadoščal pogojem

$$p(x_i) = \sum_{j=0}^n c_j \ell_{j,n}(x_i) = f(x_i)$$

za $i = 1, \dots, n$. Za vsak i je tudi vsota enaka c_i .

Dobili smo

$$p(x) = \sum_{j=0}^n f(x_j) \ell_{j,n}(x)$$

kar je Lagrangeva oblika zapisa interpolacijskega polinoma.

Primer. Naj bo $f(x) = e^x$. Pošči interpolacijski polinom za f na točkah $x_0 = 0, x_1 = 1, x_2 = 3, x_3 = 4$. Za $n = 3$ izračunamo

$$\begin{aligned}
\ell_{0,3}(x) &= \frac{(x-1)(x-3)(x-4)}{((0-1)(0-3)(0-4))} \\
&= -\frac{1}{12}(x-1)(x-3)(x-4) \\
\ell_{1,3}(x) &= \frac{x(x-3)(x-4)}{(1(1-3)(1-4))} \\
&= \frac{1}{6}x(x-3)(x-4) \\
\ell_{2,3}(x) &= \frac{x(x-1)(x-4)}{(3(3-1)(3-4))} \\
&= -\frac{1}{6}x(x-1)(x-4) \\
\ell_{3,3}(x) &= \frac{x(x-1)(x-3)}{(4(4-1)(4-3))} \\
&= \frac{1}{12}x(x-1)(x-3)
\end{aligned}$$

in dobimo rešitev

$$p(x) = e^0 \ell_{0,3}(x) + e^1 \ell_{1,3}(x) + e^3 \ell_{2,3}(x) + e^4 \ell_{3,3}(x)$$

Časovna zahtevnost za evaluacijo $\ell_{i,n}(x)$ v x_i je $\mathcal{O}(n^2)$. Ker v praksi izpustimo skupen polinom, je končna časovna zahtevnost $\mathcal{O}(n)$, tako kot Hornerjev algoritem.

Lema 2.2. Če je $f \in \mathbb{P}_n$, potem je $\sum_{i=0}^n f(x_i) \ell_{i,n}(x) = f(x)$.

Dokaz. Sledi iz enoličnosti interpolacijskega polinoma. ■

Posledica 2.3.

$$\sum_{i=0}^n \ell_{i,n}(x) = 1 \tag{8}$$

Lagrangevi bazni polinomi tvorijo **razčlenitev** oziroma **particijo enote**, ki pozitivno vpliva na stabilnost baze.

Izrek 2.4 (O napaki interpolacije.). Naj bodo $a \leq x_0 < x_1 < \dots < x_n \leq b$, $f \in \mathcal{C}^{n+1}([a, b])$ in naj bo p interpolacijski polinom za f na teh točkah. Potem za vsak $x \in [a, b]$ obstaja $\xi_x \in (a, b)$, da velja

$$f(x) - p(x) = \omega(x) \frac{f^{(n+1)}(\xi_x)}{(n+1)!}$$

kjer velja

$$\omega(x) = (x - x_0)(x - x_1) \dots (x - x_n)$$

Dokaz. Če je $x = x_i$, potem $f(x_i) - p(x_i) = 0$ in $\omega(x_i) = 0$ ter enakost velja za vsak $\xi_x \in (a, b)$. Naj bo sedaj $x \neq x_i$, $i = 0, 1, \dots, n$ in naj bo ta x fiksni. Definirajmo $F(u) = f(u) - p(u) - c\omega(u)$ za neko konstanto c , pri čemer za F velja $F \in \mathcal{C}^{n+1}([a, b])$, $F(x_i) = f(x_i) - p(x_i) - c\omega(x_i) = 0$ za $i = 0, 1, \dots, n$. Konstanto c izberemo tako, da bo tudi $F(x) = 0$. Torej ima F na $[a, b]$ $n + 2$ različnih ničel. Potem ima F' na (a, b) $n + 1$ različnih ničel. Potem ima F'' na (a, b) n različnih ničel ... Potem ima $F^{(n+1)}$ na (a, b) vsaj eno ničlo. Označimo to ničlo z ξ_x . Torej je

$$\begin{aligned} 0 &= F^{(n+1)}(\xi_x) \\ &= f^{(n+1)}(\xi_x) - p^{(n+1)}(\xi_x) - c\omega^{(n+1)}(\xi_x) \end{aligned}$$

Uporabimo razmislek od zgoraj in dobimo

$$0 = f^{(n+1)}(\xi_x) - c(n + 1)!$$

Ko to preuredimo, dobimo

$$c = \frac{1}{(n + 1)!} f^{(n+1)}(\xi_x)$$

Vstavimo c in dobimo želeno enakost ■

Za poljuben $c \in [a, b]$ po izreku velja

$$|f(x) - p(x)| = |\omega(x)| \frac{1}{(n + 1)!} |f^{(n+1)}(\xi_x)| \leq \|\omega\|_{\infty, [a, b]} \frac{1}{(n + 1)!} \|f^{(n+1)}\|_{\infty, [a, b]}$$

Iz tega sledi

$$\|f - p\|_{\infty, [a, b]} \leq \frac{1}{(n + 1)!} \|\omega\|_{\infty, [a, b]} \|f^{(n+1)}\|_{\infty, [a, b]}$$

Ta ocena je uporabna v teoriji, ne pa tudi v praksi.

Lagrangeva oblika je zaradi enostavnosti zelo uporabna pri izpeljavi formul za numerično integracijo, odvajanje ..., ima pa tudi nekaj pomankljivosti pri praktični uporabi:

- numerično računanje vrednosti polinoma v Lagrangevi obliki ...
- Numerične težave, če so interpolacijske točke preblizu skupaj
- Konstrukcija ni rekurzivna. Dodajanje novih točk je zahtevno
- ne moremo interpolirati odvodov

2.1.2 Newtonova oblika zapisa interpolacijskega polinoma

Za bazo, v kateri bomo predstavili interpolacijski polinom, izberemo **prestavljene potence**.

$$\{1, x - x_0, (x - x_0)(x - x_1), \dots, (x - x_0)(x - x_1) \dots (x - x_n)\}$$

Očitno je to baza: vidimo, da se stopnje povečujejo in je posledično kolokacijska matrika spodnje trikotna. V nadaljevanju bomo naredili rekurzivno konstrukcijo.

Vsak $p \in \mathbb{P}_n$ lahko zapišemo kot

$$\sum_{i=0}^n \left(c_i \prod_{j=0}^{i-1} (x - x_j) \right)$$

Iščemo koeficiente $(c_i)_{i=0}^n$, da bo p interpolacijski. Ta p bomo konstruirali **rekurzivno**. Naj bo p_{k-1} interpolacijski polinom za f na točkah x_0, x_1, \dots, x_{k-1} . Kako poiskati p_k , ki bo interpolacijski za f na točkah x_0, x_1, \dots, x_k , kjer bo $p_{k-1} \in \mathbb{P}_{k-1}$ in $p_k \in \mathbb{P}_k$?

$$p_k(x) = p_{k-1}(x) + c(x - x_0)(x - x_1) \dots (x - x_{k-1})$$

Vstavimo $x = x_j$, $j \in \{0, 1, \dots, k-1\}$

$$p_k(x_j) = p_{k-1}(x_j) + c \cdot 0 = f(x_j)$$

Ostane le pogoj za $x = x_k$

$$p_k(x_k) = f(x_k)$$

kjer lahko zapišemo

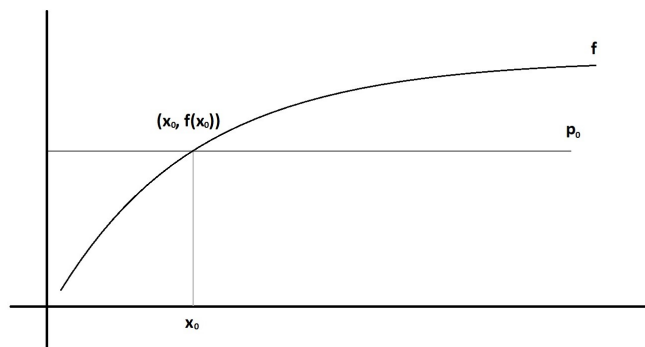
$$p_k(x_k) = p_{k-1}(x_k) + c \prod_{j=0}^{k-1} (x_k - x_j)$$

S tem je določen c , ki je kar **vodilni koeficient** od p_k . Označimo ga z $[x_0, x_1, \dots, x_k]f$ in ga imenujemo **deljena diferenca**.

Definicija 2.5. Deljena diferenca $[x_0, x_1, \dots, x_k]f$ je vodilni koeficient interpolacijskega polinoma stopnje k (koeficient pri x^k) za funkcijo f na točkah x_0, x_1, \dots, x_k .

Sledi, da lahko $p_k(x)$ zapišemo kot

$$p_k(x) = p_{k-1}(x) + [x_0, x_1, \dots, x_k]f \cdot (x - x_0)(x - x_1) \dots (x - x_{k-1})$$



V grafu: $p_0 \in \mathbb{P}_0$, $p_0(x) = x_0 \cdot 1$. Po definiciji je $[x_0]f = f(x_0)$.

Iz te rekurzivne konstrukcije dobimo

$$\begin{aligned} p_n(x) &= p_{n-1}(x) + [x_0, x_1, \dots, x_n]f(x - x_0)(x - x_1) \dots (x - x_{n-1}) \\ &= \dots \\ &= p_0(x) + [x_0, x_1]f(x - x_0) + [x_0, x_1, x_2]f(x - x_0)(x - x_1) + \\ &\quad + \dots + [x_0, x_1, \dots, x_n]f(x - x_0)(x - x_1) \dots (x - x_{n-1}) \\ p_n(x) &= \sum_{i=0}^n [x_0, x_1, \dots, x_i]f(x - x_0)(x - x_1) \dots (x - x_{i-1}) \end{aligned}$$

Temu rečemo **Newtonova oblika** zapisa interpolacijskega polinoma.

Kako izračunati deljene difference?

- $[x_0]f = f(x_0)$
- $[x_0, x_1]f = ?$

$$\begin{aligned} p_1(x) &= f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \\ &= [x_0]f \cdot 1 + [x_0, x_1]f(x - x_0) \end{aligned}$$

V zgornji enačbi je $\frac{f(x_1) - f(x_0)}{x_1 - x_0}$ vodilni koeficient, 1 in $(x - x_0)$ pa sta baza. Iz tega sledi

$$[x_0, x_1]f = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{[x_1]f - [x_0]f}{x_1 - x_0}$$

Izrek 2.6 (Rekurzivna formula za deljene difference). Naj bodo x_0, x_1, \dots, x_k paroma različne točke. Tedaj velja

$$[x_0, x_1, \dots, x_k]f = \frac{[x_1, x_2, \dots, x_k]f - [x_0, x_1, \dots, x_{k-1}]f}{x_k - x_0}$$

Opomba. Pazi na koeficiente! V števcu gredo pri prvem od x_1 do x_k , pri drugem pa od x_0 do x_{k-1} . Tista, ki smo ju izpustili, potem odštejemo v imenovalcu, torej $x_k - x_0$.

Dokaz. Naj bo q_0 interpolacijski za f na točkah x_0, x_1, \dots, x_{k-1} in q_1 interpolacijski za f na točkah x_1, x_2, \dots, x_k . Velja, da sta $q_0, q_1 \in \mathbb{P}_{k-1}$. Kako priti do polinoma p , ki bo interpolacijski na x_0, x_1, \dots, x_k ? Ta p bo tak, da bo veljalo $p \in \mathbb{P}_k$.

Sestavimo model za p

$$p(x) = \ell_0(x)q_0(x) + \ell_1(x)q_1(x), \ell_0, \ell_1 \in \mathbb{P}_1, \ell_0, \ell_1 = ?$$

$$x = x_j, j \in \{x_1, x_2, \dots, x_{k-1}\}$$

Dobimo tri pogoje:

$$(\star\star) \quad x = x_j$$

$$p(x_j) = \ell_0(x_j)q_0(x_j) + \ell_1(x_j)q_1(x_j) = (\ell_0(x_j) + \ell_1(x_j))f(x_j) \stackrel{?}{=} f(x_j)$$

$$(\star) \quad x = x_0$$

$$p(x_0) = \ell_0(x_0)f(x_0) + \ell_1(x_0)q(x_0) \stackrel{?}{=} f(x_0)$$

$$(\star) \quad x = x_k$$

$$p(x_k) = \ell_0(x_k)q(x_k) + \ell_1(x_k)f(x_k) \stackrel{?}{=} f(x_k)$$

Če izberemo

$$\ell_0(x) = \frac{x - x_k}{x_0 - x_k}$$

in

$$\ell_1(x) = \frac{x - x_0}{x_k - x_0}$$

zadostimo pogojema (\star) . Ker pa je $\ell_0(x) + \ell_1(x) = 1$ za $\forall x$, pa velja tudi $(\star\star)$.

Torej

$$p(x) = \frac{x - x_k}{x_0 - x_k}q_0(x) + \frac{x - x_0}{x_k - x_0}q_1(x)$$

Opomba. V nadaljevanju se bo namesto vodilni koeficient pisalo $v.k.$

Primerjamo vodilne koeficiente na levi in desni strani in dobimo:

$$v.k.(p) = [x_0, x_1, \dots, x_k]f$$

Na desni strani:

$$\frac{1}{x_0 - x_k} \cdot v.k.(q_0) + \frac{1}{x_k - x_0} \cdot v.k.(q_1) = \frac{[x_1, x_2, \dots, x_k]f - [x_0, \dots, x_{k-1}]f}{x_k - x_0}$$

■

2.1.2.1 Trikotna shema

$$[x_0, x_1, x_2]f = \frac{[x_1, x_2]f - [x_0, x_1]f}{x_2 - x_0}$$

Deljive difference, ki jih potrebujemo v zapisu interpolacijskega polinoma, računamo v **trikotni shemi**:

	$[\cdot]f$	$[\cdot, \cdot]f$	$[\cdot, \cdot, \cdot]f$	\dots	$[\cdot, \dots, \cdot]f$
x_0	$[x_0]f$				
x_1	$[x_1]f$	$[x_0, x_1]f$			
x_2	$[x_2]f$	$[x_1, x_2]f$	$[x_0, x_1, x_2]f$		
\vdots	\vdots	\vdots	\vdots	\ddots	
x_n	$[x_n]f$	$[x_{n-1}, x_n]f$	$[x_{n-2}, x_{n-1}, x_n]f$	\dots	$[x_0, x_1, \dots, x_n]f$

Primer. Poišči polinom p , za katerega velja $p(0) = 1$, $p(1) = 3$, $p(3) = 5$ in $p(4) = 2$.

$p \in \mathbb{P}_3$, $x_0 = 1, x_1 = 3, x_2 = 5, x_3 = 4$ baza:

$$\{1, x, x(x-1), x(x-1)(x-3)\}$$

$$\begin{array}{c|ccc} 0 & 1 & & \\ 1 & 3 & \frac{3-1}{1-0} = 2 & \\ 3 & 5 & \frac{5-3}{3-1} = 1 & \frac{1-2}{3-0} = -\frac{1}{3} \\ 4 & 2 & \frac{2-5}{4-3} = -3 & \frac{-3-1}{4-1} = -\frac{4}{3} \quad \frac{-\frac{4}{3}+\frac{1}{3}}{4-0} = -\frac{1}{4} \end{array}$$

$$p(x) = 1 \cdot 1 + 2 \cdot x - \frac{1}{3} \cdot x(x-1) - \frac{1}{4} \cdot x(x-1)(x-3)$$

Kako izračunati vrednost polinoma v Newtonovi bazi pri izbranem x ?

Označimo $d_i = [x_0, x_1, \dots, x_i]f, i = 0, 1, \dots, n, n = 4$

$$\begin{aligned} p(x) &= d_0 \cdot 1 + d_1(x - x_0) + d_2(x - x_0)(x - x_1) + \\ &\quad + d_3(x - x_0)(x - x_1)(x - x_2) + d_4(x - x_0)(x - x_1)(x - x_2)(x - x_3) \\ &= d_0 + (x - x_0)(d_1 - (x - x_1)(d_2 - (x - x_2)(d_3 - (x - x_3)d_4))) \end{aligned}$$

zapišemo

$$\begin{aligned} v_4 &= d_4 \\ v_3 &= d_3 + (x - x_3)v_4 \\ v_2 &= d_2 + (x - x_2)v_3 \\ v_1 &= d_1 + (x - x_1)v_2 \\ v_0 &= d_0 + (x - x_0)v_1 \end{aligned}$$

Algorithm 2 Posplošen Hornerjev algoritem

Input $\underline{d} = [d_0, d_1, \dots, d_n], \underline{x} = [x_0, x_1, \dots, x_n], x$
 $v_n = d_n$
for $i = n - 1 : -1 : 0$ **do**
 $v_i = d_i + (x - x_i)v_{i+1}$
end for
 $v_0 = p(x)$
Output v_0

To lahko rešimo tudi z uporabo tabele:

	d_n	d_{n-1}	\dots	d_1	d_0
x	$v_n(x - x_{n-1})$	\dots	$v_2(x - x_1)$	$v_1(x - x_0)$	
	d_n	$d_{n-1} + v_n(x - x_{n-1})$	\dots	$d_1 + v_2(x - x_1)$	$d_0 + v_1(x - x_0)$
	v_n	v_{n-1}	\dots	v_1	v_0

Primer. Poglejmo si za $n = 1$

$$\begin{aligned} p(x) &= f(x_0) \cdot 1 + [x_0, x_1]f(x - x_0) \\ &= f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \\ \xrightarrow{x_1 \rightarrow x_0} f(x_0) + \left(\lim_{x_1 \rightarrow x_0} \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) (x - x_0) p(x) &= f(x_0) - f'(x_0)(x - x_0) \end{aligned}$$

Iz tega dobimo sistem enačb:

$$\begin{aligned}p(x_0) &= f(x_0) \\ p'(x_0) &= f'(x_0)\end{aligned}$$

Definicija 2.7. Pravimo, da se polinom p s funkcijo f ujema v točki x_i $(k+1)$ -kratno, če se ujema v vrednosti in prvih k odvodih. Enačbe, ki določajo te pogoje:

$$\begin{aligned}p(x_i) &= f(x_i) \\ p'(x_i) &= f'(x_i) \\ p''(x_i) &= f''(x_i) \\ &\vdots \\ p^{(k)}(x_i) &= f^{(k)}(x_i)\end{aligned}$$

Tak polinom je Taylorjev polinom:

$$p(x) = f(x_i) + f'(x_i)(x - x_i) + \frac{f''(x_i)}{2!}(x - x_i)^2 + \dots + \frac{f^{(k)}(x_i)}{k!}(x - x_i)^k$$

Če bomo v točki x_i zahtevali $(k+1)$ -kratno ujemanje, potem bomo to točko podali $(k+1)$ -kratno:

$$x_{i-1} < x_i = x_{i+1} = x_{i+2} = \dots x_{i+k} < x_{i+k+1}$$

2.1.2.2 Posplošitev deljene difference

Poglejmo si, kako posplošimo deljenje difference.

$$[x_i, x_i, \dots, x_i]f = \frac{f^{(k)}(x_i)}{k!}$$

Opomba. V zgornji enačbi imamo $(k+1)$ x_i -jev

2.1.2.3 Posplošitev rekurzivne formule

Opomba. Vrstni red točk v deljeni diferenci **ni** pomemben.

Naj velja $x_i \leq x_{i+1} \leq x_{i+2} \leq \dots \leq x_{i+k}$. Potem je

$$[x_i, x_{i+1}, \dots, x_{i+k}]f = \begin{cases} \frac{f^{(k)}(x_i)}{k!} & x_i = x_{i+1} = \dots = x_{i+k} \\ \frac{[x_{i+1}, \dots, x_{i+k}]f - [x_i, \dots, x_{i+k-1}]f}{x_{i+k} - x_i} & \text{sicer} \end{cases}$$

Primer. Poišči polinom p , za katerega velja $p(0) = 1$, $p'(0) = 2$, $p''(0) = 3$, $p(1) = -1$, $p'(1) = 3$, $p(2) = 4$

Določimo točke x_0, x_1, \dots, x_5 :

$$x_0 = 0, x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 1, x_5 = 2$$

Potem sestavimo bazo:

$$\left\{ \underbrace{1}_{\text{vedno}}, \underbrace{x, x^2, x^3}_{\text{od } x=0}, \underbrace{x^3(x-1), x^3(x-1)^2}_{\text{od } x=1} \right\}$$

Z uporabo trikotne sheme izračunamo koeficiente za polinom:

$$\begin{array}{c|cc} 0 & 1 & \\ 0 & 1 & [0, 0]f = 2 \\ 0 & 1 & [0, 0]f = 2 \qquad [0, 0, 0]f = \frac{3}{2!} \\ 1 & -1 & [0, 1]f = -2 \quad [0, 0, 1]f = (\text{naprej po že poznanim postopku}) \\ 1 & -1 & [1, 1]f = 3 \quad [0, 1, 1]f = (\text{naprej po že poznanim postopku}) \\ 2 & 4 & [1, 2]f = 5 \quad [1, 1, 2]f = (\text{naprej po že poznanim postopku}) \end{array}$$

Iz trikotne sheme preberemo koeficiente in sestavimo interpolacijski polinom:

$$p(x) = 1 \cdot 1 + 2 \cdot x + \frac{3}{2}x^2 - \frac{11}{2}x^3 + \frac{29}{2}x^3(x-1) - \frac{79}{8}x^3(x-1)^2$$

Brez izpeljave povejmo še sledeče trditve:

Trditev 2.8. Za $f \in \mathcal{C}^k([a, b])$, $a \leq x_0 \leq x_1 \leq \dots \leq x_n \leq b$, velja

$$[x_i, x_{i+1}, \dots, x_{i+k}]f = \frac{f^{(k)}(\xi)}{k!}, \xi \in [x_i, x_{i+k}]$$

Trditev 2.9. Za $f \in \mathcal{C}^{n+1}([a, b])$, $a \leq x_0 \leq x_1 \leq \dots \leq x_n \leq b$ in interpolacijski polinom p za f na teh točkah velja

$$f(x) - p(x) = \omega(x)[x_0, x_1, \dots, x_n, x]f = \omega(x) \frac{f^{(n+1)}(\xi_x)}{(n+1)!}, \xi_x \in (a, b), x \in [a, b]$$

$$\omega(x) = \prod_{i=0}^n (x - x_i)$$

Kako izbrati interpolacijske točke na $[a, b]$? Obstaja več možnosti:

1. Ekvidistantne točke:

$$x_i = a + i \cdot h$$

kjer je $h = \frac{b-a}{n}$ in velja za $i = 0, 1, \dots, n$.

2. Čebiševe točke:

Izbira, pri kateri je neskončna norma polinoma ω najmanjša možna.

$$\min_{a \leq x_0 < x_1 < \dots < x_n \leq b} \|\omega\|_{\infty, [a, b]} = \min_{a \leq x_0 < x_1 < \dots < x_n \leq b} \max_{x \in [a, b]} |\omega(x)|$$

Rešitev so

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{2i+1}{2n+2}\pi\right)$$

za $i = 0, 1, \dots, n$

Recimo, da izberemo $f \in \mathcal{C}([a, b])$, izberemo zaporedje interpolacijskih točk $(\{x_0, x_1, \dots, x_n\})_n$ in povečujemo stopnjo n . Dobimo zaporedje interpolacijskih polinomov $(p_n)_n$. Zanima nas, kaj se dogaja z napako

$$\|f - p\|_{\infty, [a, b]} \xrightarrow{n \rightarrow \infty} ??$$

Žal ne velja nujno, da bi šla ta napaka proti 0.

Protiprimer: (Rungejev primer)

Za funkcijo $f(x) = \frac{1}{1+x^2}$ na intervalu $[-5, 5]$ interpoliramo z ekvidistantnimi točkami. Z večanjem n -ja gre napaka proti ∞ .

2.1.2.4 Odsekoma polinomske funkcije (zlepki)

Ideja: Interval $[a, b]$ razdelimo na m delov s stičnimi točkami

$$a = x_0 < x_1 < \dots < x_m = b$$

$$s : [a, b] \rightarrow \mathbb{R}$$

$$s|_{[x_i, x_{i+1}]} \in \mathbb{P}_n$$

V stičnih točkah predpišemo red gladkosti.

Primer. Odsekoma linearne interpolacijske funkcije - funkcija, označene točke, vmes povezane s premicami.

Primer. Odsekoma kubičen interpolacijski zlepek. Polinom na $[x_i, x_{i+1})$ določimo tako, da zadostimo pogojem

$$\begin{aligned}p_i(x_i) &= f(x_i) \\p'_i(x_i) &= f'(x_i) \\p_i(x_{i+1}) &= f(x_{i+1}) \\p'_i(x_{i+1}) &= f'(x_{i+1}), p_i \in \mathbb{P}_3\end{aligned}$$

3 Numerično odvajanje

Naloga: Iščemo približek za vrednost odvoda funkcije f pri nekem izbranem x -u. Približek bi radi izrazili s kombinacijo vrednosti funkcije f v bližnjih točkah x_0, x_1, \dots, x_n .

3.1 Ideja za izpeljavo aproksimacijskih formul

Kot približek za odvod funkcije f v izbranem x vzamemo vrednost odvoda interpolacijskega polinoma za f na točkah x_0, x_1, \dots, x_n pri izbranem x -u.

Za $f \in \mathcal{C}^{n+1}([a, b])$ vemo že:

$$\begin{aligned} f(x) &= p(x) + \omega(x)[x_0, x_1, \dots, x_n, x]f \\ p(x) &= \sum_{i=0}^n f(x_i)\ell_{i,n}(x) \\ \omega(x) &= \prod_{i=0}^n (x - x_i) \\ f'(x) &= p'(x) + (\omega(x)[x_0, x_1, \dots, x_n, x]f)' \\ f'(x) &= \sum_{i=0}^n f(x_i)(\ell'_{i,n}(x)) + (\omega(x)[x_0, x_1, \dots, x_n, x]f)' \end{aligned}$$

kjer prvi sumand imenujemo **aproksimacija odvoda**, drugi pa **napaka**, ki jo označujemo z $R(f)$ oziroma Rf .

Opomba. Odvod deljene difference (brez izpeljave):

$$\frac{d}{dx}[x_0, x_1, \dots, x_n, x]f = [x_0, x_1, \dots, x_n, x, x]f$$

Interpolacijske točke običajno izberemo **ekvidistantno**, za x pa izberemo eno od interpolacijskih točk, na primer x_k , $k \in \{0, 1, \dots, n\}$. Tedaj dobimo:

$$f'(x_k) = \sum_{i=0}^n f(x_i)\ell'_{i,n}(x_k) + \omega'(x_k)[x_0, x_1, \dots, x_k]f + \omega(x_k)[x_0, x_1, \dots, x_k]f$$

kjer je prvi sumand približek, drugi napaka, tretji pa je enak nič. Napako zapišemo kot

$$Rf = \omega'(x_k) \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

Primer. Vzemimo $n = 1$ in $x_1 = x_0 + h$. Zanima nas približek za $f'(x_0)$.

$$f(x) = f(x_0) \frac{x - x_1}{x_0 - x_1} + f(x_1) \frac{x - x_0}{x_1 - x_0} + (x - x_0)(x - x_1)[x_0, x_1, x]f$$

$$\begin{aligned} f'(x_0) &= f(x_0) \frac{1}{(-h)} + f(x_1) \frac{1}{h} + ((x_0 - x_1) + 0) \frac{f''(\xi)}{2} + 0 \\ &= \frac{f(x_1) - f(x_0)}{h} - \frac{h}{2} f''(\xi), \quad \xi \in [x_0, x_1] \end{aligned}$$

Ulomku, ki se pojavi v formuli, pravimo **enostranska diferenca**.

Primer. Za $n = 2$ izberemo $x_0, x_1 = x_0 + h, x_2 = x_0 + 2h$. Zanima nas približek za $f'(x_1)$ in $f''(x_1)$.

$$\begin{aligned} f(x) &= f(x_0) \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + \\ &+ f(x_2) \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} + (x - x_0)(x - x_1)(x - x_2)[x_0, x_1, x_2, x]f \end{aligned}$$

Funkcijo odvajamo

$$\begin{aligned} f'(x_1) &= f(x_0) \frac{x_1 - x_2}{2h^2} + f(x_1) \frac{x_1 - x_2 + x_1 - x_0}{-h^2} + f(x_2) \frac{x_1 - x_0}{2h^2} + Rf \\ &= f(x_0) \frac{1}{2h} + 0 + f(x_2) \frac{1}{2h} + Rf \\ &= \frac{f(x_2) - f(x_0)}{2h} + Rf \\ Rf &= \omega'(x_1)[x_0, x_1, x_2, x]f \\ &= (x_1 - x_0)(x_1 - x_2) \frac{f'''(\xi)}{3!} \\ &= -\frac{h^2}{6} f'''(\xi) \end{aligned}$$

Ko to dvoje združimo, dobimo prvi odvod

$$f'(x_1) = \frac{f(x_2) - f(x_0)}{2h} - \frac{h^2}{6} f'''(\xi)$$

Za izračun drugega ponovno odvajamo

$$\begin{aligned} f''(x_1) &= f(x_0) \frac{2}{2h^2} + f(x_1) \frac{2}{-h^2} + f(x_2) \frac{2}{2h^2} + (\omega(x)[x_0, x_1, x_2, x]f)'' \Big|_{x=x_1} \\ &= \frac{f(x_2) - 2f(x_1) + f(x_0)}{h} + Rf \end{aligned}$$

Ulomku se reče simetrična diferenca za drugi odvod.

$$Rf = \omega''(x_1)[x_0, x_1, x_2, x_1]f + 2 * \omega(x_1)[x_0, x_1, x_2, x_1, x_1]f + 0$$

Bralec naj preveri, da je $\omega''(x_1) = 0$ in $\omega'(x_1) = -h^2$.

Sledi

$$Rf = -2h^2 \frac{f^{(4)}(\xi)}{4!} = -\frac{1}{12}h^2 f^{(4)}(\xi)$$

$$f''(x_1) = \frac{f(x_2) - 2f(x_1) + f(x_0)}{h} - \frac{1}{12}h^2 f^{(4)}(\xi)$$

(slikca)

Imamo dve vrsti napake:

- napaka metode - Rf (Pada, ko h pada)
- neodstranljiva napaka - napake pri izračunu vrednosti funkcije (Raste, ko h pada)

Ko računamo vrednosti funkcije $f(x_i)$, zaradi zaokrožitvenih napak aritmetike dobimo približek $\tilde{f}(x_i)$, da velja

$$|f(x_i) - \tilde{f}(x_i)| \leq \varepsilon$$

Primer. Ocenimo za obe napaki odvod

$$f'(x_1) = \frac{f(x_2) - f(x_0)}{2h} - \frac{h^2}{6}f'''(\xi)$$

- Napaka metode:

$$D_M \leq \frac{h^2}{6} \|f'''\|_{\infty, [x_0, x_2]}$$

- Neodstranljiva napaka:

$$D_N \leq \frac{2\varepsilon}{2h} = \frac{\varepsilon}{h}$$

- Skupna napaka:

$$D_S = D_M + D_N$$

(slikca)

4 Numerična integracija

Naloga: Radi bi izračunali približek za integral

$$Sf = \int_a^b f(x)dx$$

$$S : \mathcal{C}([a, b]) \rightarrow \mathbb{R}$$

S je linearen funkcional.

Približek za integral bi radi izrazili s kombinacijo vrednosti funkcije f v izbranih točkah x_0, x_1, \dots, x_n iz intervala $[a, b]$.

IDEJA: Namesto funkcije f integriramo interpolacijski polinom za f na točkah $a \leq x_0 < x_1 < \dots < x_n \leq b$.

Vemo že:

$$f(x) = p(x) + \omega(x)[x_0, x_1, \dots, x_n, x]f \quad (za \ f \in \mathcal{C}^{n+1}([a, b])) \quad (9)$$

$$p(x) = \sum_{i=0}^n f(x_i) \ell_{i,n}(x)$$

Integriramo 9:

$$\int_a^b f(x)dx = \int_a^b p(x)dx + \int_a^b \omega(x)[x_0, x_1, \dots, x_n, x]f dx$$

$$Sf = \text{približek za integral} + \text{napaka}$$

$$Sf = Ff + Rf$$

To je integracijsko pravilo oziroma kvadratna formula.

$$Ff = \int_a^b p(x)dx = \sum_{i=0}^n f(x_i) \int_a^b \ell_{i,n}(x)dx = \sum_{i=0}^n A_i f(x_i)$$

Za $i = 0, \dots, n$ $\int_a^b \ell_{i,n}(x)dx$ označimo z A_i in jih imenujemo kot uteži integracijskega pravila, točke x_0, x_1, \dots, x_n pa imenujemo vozli integracijskega pravila.

Definicija 4.1. Red oziroma stopnja integracijskega pravila $Sf = Ff + Rf$ je enaka m , če je pravilo točno za vse polinome stopnje $\leq m$. To je, če velja $Rp = 0$ za $\forall p \in \mathbb{P}_m$ in $R(x^{m+1}) \neq 0$. To ekvivalentno zapišemo kot

$$Rx^j = 0 \text{ za } j = 0, 1, \dots, m$$

kar je ekvivalentno

$$Sx^j = Fx^j$$

Glede na izbiro vozlov ločimo dve vrsti pravil:

- Newton-Cotesova pravila - vozle izberemo ekvidistantno:

$$x_0 = a, x_i = x_0 + ih, h = \frac{b-a}{n}, i = 0, 1, \dots, n$$

Ločimo:

- zaprta pravila (upoštevamo krajišči)

$$\int_a^b f(x)dx = \sum_{i=0}^n A_i f(x_i) + Rf$$

- odprta pravila (ne upoštevamo krajišč):

$$\int_a^b f(x)dx = \sum_{i=1}^{n-1} A_i f(x_i) + Rf$$

- Gaussova pravila - vozle izračunamo oziroma določimo tako, da je pravilo čim višjega reda.

4.1 Nekaj osnovnih Newton-Cotesovih (N-C) pravil

Primer. $n = 1$, $a = x_0$, $b = x_1 = x_0 + h$, $h = b - a$

$$\begin{aligned} \int_a^b f(x)dx &= \int_a^b \left(f(x_0) \frac{x-x_1}{x_0-x_1} + f(x_1) \frac{x-x_0}{x_1-x_0} \right) dx \\ &\quad + \int_a^b (x-x_0)(x-x_1)[x_0, x_1, x] f dx \\ &= f(x_0) \frac{1}{-h} \int_{x_0}^{x_1} (x-x_1) dx + f(x_1) \frac{1}{h} \int_{x_0}^{x_1} (x-x_0) dx \\ &\quad + \int_{x_0}^{x_1} (x-x_0)(x-x_1)[x_0, x_1, x] f dx \\ &= \frac{h}{2} (f(x_0) + f(x_1)) + Rf \end{aligned} \quad (10)$$

Za reševanje Rf uporabimo izrek:

Izrek 4.2. Posplošen izrek o povprečni vrednosti:

če f na $[a, b]$ ne spremeni predznaka, je

$$\int_a^b f(x)g(x)dx = g(\xi) \int_a^b f(x)dx$$

za $\xi \in [a, b]$.

Z uporabo izreka dobimo

$$\begin{aligned} Rf &= \int_{x_0}^{x_1} (x - x_0)(x - x_1)[x_0, x_1, x]f dx \\ &= \dots \\ &= -\frac{1}{12}h^3 f''(\xi) \end{aligned}$$

Uporabili smo tudi trapezno pravilo:

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}(f(x_0) + f(x_1)) - \frac{1}{12}h^3 f''(\xi)$$

za $\xi \in [x_0, x_1]$ in za $f \in \mathcal{C}^2([x_0, x_1])$ (skica)

Določimo še red pravila

- Red je zagotovo vsaj 1
- $Rp = ?$ za $p \in \mathbb{P}_2$

$$\begin{aligned} R(x - x_0)^2 &= S(\cdot - x_0)^2 + F(\cdot - x_0)^2 \\ &= \int_{x_0}^{x_1} (\cdot - x_0)^2 d\cdot - \frac{h}{2}((x_0 - x_0)^2 + (x_1 - x_0)^2) \\ &= \frac{(x_1 - x_0)^3}{3} - \frac{h}{2}h^2 \\ &= \frac{h^3}{3} - \frac{h^3}{2} \\ &= -\frac{h^3}{6} \\ &\neq 0 \end{aligned}$$

Red pravila je torej 1

Primer. $n = 2, x_0 = a, x_1 = x_0 + h, x_2 = x_0 + 2h = b, h = \frac{b-a}{2}$

$$\begin{aligned} \int_{x_0}^{x_1} f(x)dx &= \int_{x_0}^{x_1} \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}dx + \int_{x_0}^{x_1} \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}dx + \\ &\quad + \int_{x_0}^{x_1} \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}dx + \\ &\quad + \int_{x_0}^{x_1} (x - x_0)(x - x_1)(x - x_2)[x_0, x_1, x_2, x]f dx \end{aligned}$$

Izračunamo uteži

$$\begin{aligned}
 A_0 &= \frac{1}{h^2} \int_{x_0}^{x_1} (x - x_1)(x - x_2) dx \\
 &= \frac{1}{h^2} \int_{x_0}^{x_1} h(t-1)h(t-2)h dt \\
 &= \dots \\
 &= \frac{h}{3}
 \end{aligned}$$

Podobno:

$$\begin{aligned}
 A_1 &= \frac{4}{3}h \\
 A_2 &= \frac{1}{3}h
 \end{aligned}$$

$$\begin{aligned}
 \int_{x_0}^{x_2} f(x) dx &= \frac{h}{3} (f(x_0) + 4f(x_1) + f(x_2)) + Rf \\
 Rf &= \int_{x_0}^{x_2} (x - x_0)(x - x_1)(x - x_2)[x_0, x_1, x_2, x] f dx = ??
 \end{aligned}$$

Določimo red pravila. Red pravila je vsaj 2.

$$\begin{aligned}
 R(x - x_0)^3 &= S(x - x_0)^3 - F(x - x_0)^3 \\
 &= \int_{x_0}^{x_2} (x - x_0)^3 dx - \frac{h}{3} ((x_0 - x_0)^3 + 4(x_1 - x_0)^3 + (x_2 - x_0)^3) \\
 &= \frac{(x_2 - x_0)^4}{4} - \frac{h}{3} (4h^3 + 8h^3) \\
 &= \frac{(2h)^4}{4} - 4h^4 \\
 &= 0
 \end{aligned}$$

Red je vsaj 3

$$\begin{aligned}
 R(x - x_0)^4 &= S(x - x_0)^4 - F(x - x_0)^4 \\
 &= \int_{x_0}^{x_2} (x - x_0)^4 dx - \frac{h}{3} ((x_0 - x_0)^4 + 4(x_1 - x_0)^4 + (x_2 - x_0)^4) \\
 &= \frac{(2h)^5}{5} - \frac{4}{15}h^5 \\
 &\neq 0
 \end{aligned}$$

Red je 3.

Uporabimo nastavek za napako:

$$Rf = Cf^{(m+1)}(\xi), \xi \in [a, b]$$

kjer je C konstanta in m red pravila.

Za f izberemo $f(x) = (x - x_0)^4$ in dobimo

$$R(x - x_0)^4 = C \cdot f^{(4)}(\xi) = C \cdot 4!$$

Sledi

$$C = -\frac{h^5}{90}$$

in dobimo

$$Rf = -\frac{h^5}{90}f^{(4)}(\xi)$$

Opomba. Ta nastavek sledi iz razvoja funkcije f v Taylorjevo vrsto

$$\begin{aligned} f(x) = f(a) + f(a)'(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \dots + \\ + \frac{f^{(m)}(a)}{m!}(x - a)^m + \frac{f^{(m+1)}(\xi)}{(m + 1)!}(x - a)^{m+1} \end{aligned}$$

Ostane je

$$Rf = \frac{f^{(m+1)}(\xi)}{(m + 1)!}R(x - a)^{m+1}$$

Iz ostanka izpostavimo konstanto C

$$C = \frac{R(x - a)^{m+1}}{(m + 1)!}$$

Izpeljali smo Simpsonovo pravilo

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) - \frac{h^5}{90}f^{(4)}(\xi)$$

(slikca?)

4.1.1 Alternativna izpeljava integracijskih pravil - metoda nedoločeni koeficientov

$$\int_a^b f(x)dx = \sum_{i=0}^n A_i f(x_i) + Rf$$

Ideja je, da neznane uteži določimo tako, da bo red pravila čim višji. Poglejmo si to na primeru Simpsonovega pravila:

Izberemo bazo polinomov

$$\{1, x - x_0, (x - x_0)^2, \dots\}$$

$$\int_{x_0}^{x_2} f(x)dx = A_0 f(x_0) + A_1 f(x_1) + A_2 f(x_2) + Rf$$

Če rečemo, da $f(x) = 1$:

$$\begin{aligned} \int_{x_0}^{x_2} 1dx &= A_0 + A_1 + A_2 \\ 2h &= A_0 + A_1 + A_2 \end{aligned}$$

Če rečemo, da $f(x) = x - x_0$:

$$\begin{aligned} \int_{x_0}^{x_2} (x - x_0)dx &= A_0(x_0 - x_0) + A_1(x_1 - x_0) + A_2(x_2 - x_0) \\ 2h &= A_1 + 2A_2 \end{aligned}$$

Če rečemo, da $f(x) = (x - x_0)^2$:

$$\frac{8}{3}h = A_1 + 4A_2$$

Rešimo sistem in dobimo rešitev:

$$\begin{aligned} A_0 &= A_2 = \frac{h}{3} \\ A_1 &= \frac{4h}{3} \end{aligned}$$

Primer. Primer odprtega N-C pravila - pravokotniško pravilo (skica)

$$\int_a^b f(x)dx = A_1 f(x_1) + Rf = 2hf(x_1) + \frac{h^3}{3}f''(\xi)$$

4.2 Numerične napake pri integracijskih pravilih

Poglejmo si neodstranljivo napako. Recimo, da velja $|f(x_i) - \tilde{f}(x_i)| \leq \epsilon$. Naj bo $Ff = \sum_{i=0}^n A_i f(x_i)$.

$$D_a = \left| \sum_{i=0}^n A_i (f(x_i) - \tilde{f}(x_i)) \right| \leq \sum_{i=0}^n |A_i| \epsilon = \epsilon \sum_{i=0}^n |A_i|$$

Upoštevamo, da je pravilo točno vsaj za konstante

$$\sum_{i=0}^n A_i \cdot 1 = F1 = S1 = \int_a^b 1 dx = b - a$$

Če so uteži vse pozitivne, potem $D_a \leq \epsilon(b - a)$ in ni numeričnih težav, če n večamo.

Žal pa tako pri zaprtih N-C pravilih za $n \geq 8$ kot tudi pri odprtih N-C pravilih za $n \geq 4$ dobimo negativne uteži.

Primer. $n = 50$, $[a, b] = [0, 1]$, velja $\sum_{i=0}^n |A_i| = 6.7 \cdot 10^{10}$ (zaprto N-C pravilo)

Namesto, da bi uporabljali pravila z visoko stopnjo n raje uporabljamo t.i. sestavljena pravila.

4.3 Sestavljena integracijska pravila

Računamo integral

$$\int_a^b f(x) dx$$

Ideja: Interval $[a, b]$ razdelimo na manjše podintervale in na vsakem od podintervalov uporabimo integracijsko pravilo nizkega reda in nato rezultate seštejemo.

Recimo, da želimo uporabiti m osnovnih pravil, vsak osnovno pravilo pa zahteva, da razdelimo podinterval na n delov. Imamo korak $h = \frac{b-a}{m \cdot n}$ in točke $x_i = a + i \cdot h$, kjer gre i od 0 do $m \cdot n$. Sestavljeno pravilo bo potem oblike

$$\int_a^b f(x) dx = \sum_{i=0}^{m \cdot n} A_i f(x_i) + Rf$$

Izpeljimo sestavljeno trapezno pravilo.

4.3.1 Sestavljeno trapezno pravilo

Osnovno:

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}(f(x_0) + f(x_1)) - \frac{h^3}{12}f''(\xi)$$

kjer je $h = x_1 - x_0$.

(skica) Sestavljeno pravilo:

$$\begin{aligned} \int_a^b f(x)dx &= \sum_{i=0}^{m-1} \int_{x_i}^{x_{i+1}} f(x)dx \\ &= (\text{vstavimo osnovno pravilo}) \\ &= \sum_{i=0}^{m-1} \left(\frac{h}{2}(f(x_i) + f(x_{i+1})) - \frac{h^3}{12}f''(\xi_i) \right) \\ &= \frac{h}{2}(1 \cdot f(x_0) + 2 \cdot f(x_1) + 2 \cdot f(x_2) + \dots \\ &\quad + 2 \cdot f(x_{m-1}) + 1 \cdot f(x_m)) + Rf \end{aligned}$$

Ostanek določimo

$$\begin{aligned} Rf &= \sum_{i=0}^{m-1} \left(-\frac{h^3}{12}f''(\xi_i) \right) \\ &= -\frac{h^3}{12} \sum_{i=0}^{m-1} f''(\xi_i) \\ &= -\frac{h^3}{12} m f''(\mu) \\ &= -\frac{h^3}{12} \frac{b-a}{h} f''(\mu) \\ &= -\frac{h^2}{12} (b-a) f''(\mu) \end{aligned}$$

Ker predpostavimo, da je $f \in \mathcal{C}^2([a, b])$, obstaja $\mu \in [a, b]$, da je

$$\frac{1}{m} \sum_{i=0}^{m-1} f''(\xi_i) = f''(\mu)$$

4.3.2 Sestavljeno Simpsonovo pravilo

Osnovno: $n = 2$

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) - \frac{h^5}{90}f^{(4)}(\xi)$$

kjer je $\xi \in [x_0, x_2]$ in $f \in \mathcal{C}^4([a, b])$

(skica)

$$\begin{aligned}
\int_a^b f(x)dx &= \sum_{i=0}^{m-1} \int_{x_{2i}}^{x_{2i+2}} f(x)dx \\
&= \sum_{i=0}^{m-1} \left(\frac{h}{3} (f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})) - \frac{h^5}{90} f^{(4)}(\xi) \right) \\
&= \frac{h}{3} (f(x_0) + 4 \cdot f(x_1) + 2 \cdot f(x_2) + 4 \cdot f(x_3) + 2 \cdot f(x_4) + \\
&\quad + \dots + 2 \cdot f(x_{2m-2}) + 4 \cdot f(x_{2m-1}) + 1 \cdot f(x_{2m})) + Rf \\
&= \frac{h}{3} f(x_0) + 4 \sum_{i=0}^{m-1} f(x_{2i+1}) + 2 \sum_{j=1}^{m-1} f(x_{2j}) + f(x_{2m}) + Rf \\
Rf &= -\frac{h^5}{90} \sum_{i=0}^{m-1} f^{(4)}(\xi_i) = -\frac{h^5}{90} m f^{(4)}(\mu) = -\frac{h^4}{180} (b-a) f^{(4)}(\mu)
\end{aligned}$$

4.3.3 Ocena napake in Richardsonova ekstrapolacija

Recimo, da računamo približek za integral s sestavljenim N-C pravilom s korakom h . Vprašanje je, kako v praksi določiti ta korak.

Označimo F_h približek, ki ga izračunamo s korakom h za neko fiksno funkcijo f . Predpostavimo, da velja

$$I = F_h + c_0 \cdot h^p + \mathcal{O}(h^{p+1})$$

kjer je c_0 konstanta.

Namesto s korakom h računamo s korakom $\frac{h}{2}$:

$$\begin{aligned}
I &= F_{\frac{h}{2}} + c_0 \left(\frac{h}{2}\right)^p + \mathcal{O}(h^{p+1}) \\
2^p I &= 2^p F_{\frac{h}{2}} + c_0 h^p + \mathcal{O}(h^{p+1}) (2^p - 1) I = 2^p F_{\frac{h}{2}} - F_h + \mathcal{O}(h^{p+1}) \\
I &= \frac{2^p F_{\frac{h}{2}} - F_h}{(2^p - 1)} + \mathcal{O}(h^{p+1})
\end{aligned}$$

Ulomku pravimo ekstrapoliran približek.

$$I = \frac{2^p F_{\frac{h}{2}} - F_h + F_{\frac{h}{2}} - F_h}{(2^p - 1)} + \mathcal{O}(h^{p+1}) = F_{\frac{h}{2}} + \frac{F_{\frac{h}{2}} - F_h}{2^p - 1} + \mathcal{O}(h^{p+1})$$

Ulomek je ocena za napako približka $F_{\frac{h}{2}}$.

Podobno:

$$I = \frac{2^p F_{\frac{h}{2}} - 2^p F_h + F_h - 2^p F_h}{(2^p - 1)} + \mathcal{O}(h^{p+1}) = F_h + 2^p \frac{F_{\frac{h}{2}} - F_h}{2^p - 1} + \mathcal{O}(h^{p+1})$$

Ulomek je ocena za napako približka F_h .

Pri sestavljenem Simpsonovem pravilu je $p = 4$ in

$$I = F_{\frac{h}{2}} + \frac{F_{\frac{h}{2}} - F_h}{15} + \mathcal{O}(h^{p+1})$$

$$\text{Ekstrapoliran približek} = \frac{16F_{\frac{h}{2}} - F_h}{15}$$

$$\text{Ekstrapoliran približek za sestavljeno trapezno pravilo} = \frac{4F_{\frac{h}{2}} - F_h}{3}$$

4.4 Gaussova integracijska pravila

Ideja za izpeljavo: Uteži in vozle izračunamo tako, da je pravilo čim višjega reda oziroma da je pravilo točno za polinome čim višjih stopenj.

$$\int_a^b f(x) dx = \sum_{i=0}^n A_i f(x_i) + Rf$$

Temu rečemo metoda nedoločenih koeficientov:

$$f(x) = x^k$$

$$\int_a^b x^k dx = \sum_{i=0}^n A_i x_i^k, k = 0, 1, \dots, 2n + 1$$

Imamo neznanke A_0, A_1, \dots, A_n , ki jim rečemo uteži, in x_0, x_1, \dots, x_n , ki jim pravimo vozli. Imamo torej $2n + 2$ neznank, torej rabimo tudi $2n + 2$ enačb. Zgornji sistem je nelinearen sistem $2n + 2$ enačb za $2n + 2$ neznank.

Ideja je ločiti enačbe za neznane vozle od enačb za neznane uteži. Kako?

$$f(x) = p(x) + \omega(x)[x_0, x_1, \dots, x_n, x]f$$

Če je $f \in \mathbb{P}_{n+r}$, je $[x_0, x_1, \dots, x_n, x]f \in \mathbb{P}_{r-1}$

$$Rf = \int_a^b \omega(x)[x_0, x_1, \dots, x_n, x]f dx$$

Definiramo

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx$$

Če hočemo, da bo pravilo reda vsaj $n + r$, mora biti

$$\int_a^b \omega(x)[x_0, \dots, x_n, x]f dx = 0$$

za nek $f \in \mathbb{P}_{n+r}$.

Če bo $\langle \omega, q \rangle = 0 \ \forall q \in \mathbb{P}_{r-1}$, bo pravilo zagotovo reda vsaj $n + r$. Drugače povedano: Če je $\omega \perp \mathbb{P}_{r-1}$, je pravilo zagotovo reda vsaj $n + r$. Koliko je lahko r največ?

$$\omega(x) = \prod_{i=0}^n (x - x_i)$$

ω je lahko pravokoten 'največ' na \mathbb{P}_n

$$r - 1 \leq n$$

$$r \leq n + 1$$

Če bo $\omega \perp \mathbb{P}_n$, potem bo pravilo reda $n + r = n + n + 1 = 2n + 1$. Če torej izberemo vozle x_0, x_1, \dots, x_n tako, da bo $\omega \perp \mathbb{P}_n$, potem je pravilo reda $2n + 1$. To pravilo imenujemo Gaussovo pravilo. (Prva trditev velja tudi v drugo smer oziroma velja ekvivalenca).

Za neznane vozle dobimo $n + 1$ pogojev za $n + 1$ neznank:

$$\omega \perp 1, \omega \perp x, \omega \perp x^2, \dots, \omega \perp x^n$$

Iz pogojev sestavimo sistem nelinearnih enačb:

$$\begin{aligned} \langle \omega, 1 \rangle &= \int_a^b \omega(x) \cdot 1 dx = 0 \\ \langle \omega, x \rangle &= \int_a^b \omega(x) \cdot x dx = 0 \\ &\vdots \\ \langle \omega, x^n \rangle &= \int_a^b \omega(x) \cdot x^n dx = 0 \end{aligned}$$

Opomba. Drug način: Iz baze $1, x, \dots, x^{n+1}$ izračunamo ortonormirano bazo polinomov P_0, P_1, \dots, P_{n+1} (glede na dan skalarni produkt). Za vozle izberemo ničle polinoma P_{n+1} .

Izrek 4.3. Naj bo $Ff = \sum_{i=0}^n A_i f(x_i)$ Gaussovo integracijsko pravilo ($Sf = Ff = Rf$) reda $2n + 1$. Uteži $A_i = \int_a^b \ell_{i,n}(x) dx$ so pozitivne. Za $f \in \mathcal{C}^{2n+2}([a, b])$ je napaka oblike

$$Rf = \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \int_a^b \omega^2(x) dx$$

Dokaz.

$$\int_a^b f(x) dx = \sum_{i=0}^n A_i f(x_i) + Rf$$

Če za f izberemo $f(x) = \ell_{j,n}^2(x)$, je $Rf = 0$.

$$\int_a^b \ell_{j,n}^2(x) dx = \sum_{i=0}^n A_i \ell_{j,n}^2(x_i) = A_j$$

iz česar sledi $A_j > 0 \ \forall j = 0, 1, \dots, n$.

Naj bo q interpolacijski polinom za f na točkah $x_0, x_0, x_1, x_1, \dots, x_n, x_n$.

Velja

$$f(x) = q(x) + \omega^2(x)[x_0, x_0, x_1, x_1, \dots, x_n, x_n, x]f$$

Vemo tudi, da je $q \in \mathbb{P}_{2n+1}$, zato sledi, da je

$$\int_a^b q(x) dx = \sum_{i=0}^n A_i q(x_i)$$

kjer je $q(x_i) = f(x_i)$.

Če to formulo integriramo

$$\int_a^b f(x) dx = \int_a^b q(x) dx + \int_a^b \omega^2(x)[x_0, x_0, x_1, x_1, \dots, x_n, x_n, x]f dx$$

Leva stran je enaka Sf , desna stran pa $Ff + Rf$.

$$Sf = \int_a^b f(x) dx$$

$$Ff = \int_a^b q(x) dx$$

$$Rf = \int_a^b \omega^2(x)[x_0, x_0, x_1, x_1, \dots, x_n, x_n, x]f dx$$

Ker $\omega^2(x)$ ne spremeni predznaka na intervalu $[a, b]$, dobimo

$$\begin{aligned} Rf &= \int_a^b \omega^2(x) [x_0, x_0, x_1, x_1, \dots, x_n, x_n, x] f dx \\ &= [x_0, x_0, x_1, x_1, \dots, x_n, x_n, \tilde{\xi}] f \int_a^b \omega^2(x) dx \\ &= \frac{f^{(2n+2)}(\xi)}{(2n+2)!} \frac{a}{b} \omega^2(x) dx \end{aligned}$$

■

Primer. Določite vozla x_0 in x_1 ter uteži A_0 in A_1 v Gaussovem integracijskem pravilu

$$\int_{-1}^1 f(x) dx = A_0 f(x_0) + A_1 f(x_1) + Rf$$

$$\omega(x) = (x - x_0)(x - x_1)$$

Enačbe: $\omega \perp 1, \omega \perp x$

$$0 = \int_{-1}^1 (x - x_0)(x - x_1) dx = \int_{-1}^1 (x^2 - (x_0 + x_1)x + x_0x_1) dx = 2\left(\frac{1}{3} - x_0x_1\right)$$

$$0 = \int_{-1}^1 (x - x_0)(x - x_1)x dx = \int_{-1}^1 (x^3 - (x_0 + x_1)x^2 + x_0x_1x) dx = -2(x_0 + x_1)\frac{1}{3}$$

Dobimo enačbi:

$$\begin{aligned} 2\left(\frac{1}{3} - x_0x_1\right) &= 0 \\ -\frac{2}{3}(x_0 + x_1) &= 0 \end{aligned}$$

Dobimo vozla:

$$\begin{aligned} x_0 &= -\frac{\sqrt{3}}{3} \\ x_1 &= \frac{\sqrt{3}}{3} \end{aligned}$$

Poračunamo tudi uteži:

$$A_0 = \int_{-1}^1 \frac{x - x_1}{x_0 - x_1} dx = \dots = 1 = A_1$$

kjer kot x_0 in x_1 vstavimo izračunani vozlišči.

Izračunamo še ostanek (za $f \in \mathcal{C}^4([-1, 1])$):

$$Rf = \frac{f^{(4)}(\xi)}{4!} \int_{-1}^1 \left(x - \frac{\sqrt{3}}{3}\right)^2 \left(x + \frac{\sqrt{3}}{3}\right)^2 dx = \dots = \frac{1}{135} f^{(4)}(\xi)$$

Opomba. V integral lahko dodamo tudi kako pozitivno utež ρ . Računamo torej približek za integral oblike $\int_a^b f(x) \rho(x) dx$.

4.5 Integrali v več dimenzijah

4.5.1 Dvojni integral

Za dano funkcijo f , zvezno na območju $\Omega = [a, b] \times [c, d]$ iščemo integral

$$\iint_{\Omega} f(x, y) dx dy$$

Integracijsko pravilo je oblike

$$\iint_{\Omega} f(x, y) dx dy = \sum_{i=0}^n \sum_{j=0}^m A_{i,j} f(x_i, y_j) + Rf$$

kjer velja $(x_i, y_j) \in \Omega$.

4.5.2 Tenzorska pravila

(slikca) $[a, b]$ razdelimo s korakom h na n delov: $h = \frac{b-a}{n}$, $x_i = a + i \cdot h$ za $i = 0, 1, \dots, n$. $[c, d]$ razdelimo s korakom k na m delov: $k = \frac{d-c}{m}$, $y_j = c + j \cdot k$ za $j = 0, 1, \dots, m$.

(mogoče bolje skica tu?)

Dobili smo mrežne točke (x_i, y_j) , določiti pa moramo še pripadajoče uteži $A_{i,j}$. Za to bomo uporabili Fubinijev izrek

$$\iint_{\Omega} f(x, y) dx dy = \int_a^b dx \int_c^d f(x, y) dy = \int_c^d dy \int_a^b f(x, y) dx$$

Poglejmo si natančneje uporabo trapeznega pravila.

$$\begin{aligned} \iint_{\Omega} f(x, y) dx dy &= \int_a^b dx \int_c^d f(x, y) dy \\ &= \frac{h}{2} (g(x_0) + 2 \sum_{i=1}^{n-1} g(x_i) + g(x_n)) - \frac{h^2}{12} (b-a) \frac{d^2}{dx^2} g(x) \Big|_{x=\xi} \end{aligned}$$

$$\begin{aligned}
g(x_i) &= \int_c^d f(x_i, y) dy \\
&= \frac{k}{2} (f(x_i, y) + 2 \sum_{j=1}^{m-1} f(x_i, y_j) + f(x_i, y_m)) - \frac{k^2}{12} (d - c) \frac{d^2}{dx^2} f(x_i, y) \Big|_{y=\mu}
\end{aligned}$$

Dobimo

$$\iint_{\Omega} f(x, y) dx dy = \frac{h \cdot k}{4} \sum_{i=0}^n \sum_{j=0}^m A_{i,j} f(x_i, y_j) + \mathcal{O}(h^2 + k^2)$$

Za izračun $A_{i,j}$ se uporabi

$$\begin{aligned}
A_{i,j} &= \alpha_i \beta_j \\
\alpha &= (\alpha_i)_{i=0}^n = (1, 2, 2, 2, \dots, 2, 1) \\
\beta &= (\beta_j)_{j=0}^m = (1, 2, 2, 2, \dots, 2, 1)
\end{aligned}$$

kjer imamo $n - 1$ dvojek pri α oziroma $m - 1$ dvojek pri β .

(še ena slika)

4.5.3 Simpsonovo pravilo

(nova skica) Napaka: $Rf = \mathcal{O}(h^4 + k^4)$

Recimo, da računamo integral

$$\int_{\Omega} f(x) dx$$

kjer je $\Omega \subseteq \mathbb{R}^d$ in velja $\Omega = [0, 1]^d$. Recimo tudi, da v vsaki dimenziji uporabimo n točk, v katerih evaluiramo funkcijo. Torej je korak $h = \frac{1}{n-1} \approx \frac{1}{n}$. Število vseh izračunov vrednosti funkcije je torej $N = n^d$ ($n = N^{\frac{1}{d}}$), torej $h \approx N^{-\frac{1}{d}}$. Pri sestavljenem trapeznem pravilu je napaka $\mathcal{O}(h^2) = \mathcal{O}(N^{-\frac{2}{d}})$. Pri sestavljenem Simpsonovem pravilu je napaka $\mathcal{O}(h^4) = \mathcal{O}(N^{-\frac{4}{d}})$.

Pri velikih dimenzijah d so alternativa determinističnim metodam verjetnostne metode.

4.5.4 Metoda Monte-Carlo

Z uporabo enakomerno porazdeljene slučajne spremenljivke na intervalu $[a, b]$ in krepkega zakona velikih števil pridemo do sledečega približka:

$$I = \int_a^b f(x)dx \approx \frac{b-a}{N} \sum_{i=1}^N f(x_i)$$

kjer so x_i naključne vrednosti iz intervala $[a, b]$. Za napako velja $I - I_N = \mathcal{O}(N^{-\frac{1}{2}})$. Za integral na $\Omega = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_d, b_d]$ velja enako, le da so sedaj X_i slučajni vektorji z vrednostmi v Ω .

$$I = \int_{\Omega} f(x)dx \approx \frac{\prod_{j=1}^d (b_j - a_j)}{N} \sum_{i=1}^N f(x_i)$$

Napaka je enaka kot prej, torej $I - I_N = \mathcal{O}(N^{-\frac{1}{2}})$.

5 Numerično reševanje navadnih diferencialnih enačb

Ponovitev teorije:

- NDE je funkcijska enačba, v kateri poleg neznane funkcije nastopajo tudi njeni odvodi.
- Red NDE je stopnja najvišjega odvoda, ki nastopa v enačbi.
- Rešitev diferencialne enačbe reda p je p -parametrična družina funkcij. Če hočemo točno določeno rešitev, moramo predpisati p pogojev.

Ločimo začetne in robne probleme.

5.0.1 Začetni problem reda 1

$$\begin{aligned} y' &= f(x, y) \\ y(a) &= y_a \end{aligned}$$

kjer je y_a podana konstanta in $f : [a, b] \times \mathbb{R} \rightarrow \mathbb{R}$.

5.0.2 Začetni problem reda p

$$\begin{aligned}y^{(p)} &= f(x, y, y', \dots, y^{(p-1)}) \\y(a) &= y_{a,0} \\y'(a) &= y_{a,1} \\\vdots \\y^{(p-1)}(a) &= y_{a,p-1}\end{aligned}$$

kjer so $y(a), y'(a), \dots, y^{(p-1)}(a)$ podane konstante.

Pri numeričnem reševanju in analizi rešitev se bomo omejili na probleme, pri katerih so izpolnjeni pogoji eksistenčnega izreka (E.I.).

5.0.3 Robni problemi

$$\begin{aligned}x^{(4)} + xy &= 0 \\y(a) &= y_{a,0} \\y'(a) &= y_{a,1} \\y(b) &= y_{b,0} \\y'(b) &= y_{b,1}\end{aligned}$$

Robnih primerov pri tem predmetu ne bomo reševali.

Numerične metode za reševanje teh problemov delimo na:

1. Diferenčne metode

Približek k rešitvi iščemo diskretno. Numerična rešitev je sestavljena iz zaporedja točk $a = x_0 < x_1 < x_2 < \dots < x_m = b$ in pripadajočega zaporedja vrednosti y_0, y_1, \dots, y_m , pri čemer je

$$y_n \approx y(x_n)$$

za $n = 0, 1, 2, \dots$, kjer je y_n numerični približek, $y(x_n)$ pa točna rešitev pri x_n . (slika)

2. Metode, pri katerih iščemo približek k rešitvi kot funkcijo v nekem izbranem podprostoru funkcij

Dve znani metodi sta kolokacija in metoda končnih elementov. Teh metod letos ne bomo obravnavali.

Diferenčne metode delimo na

1. enočlenske metode

približek za y_{n+1} izračunamo iz približka za y_n

2. veččlenske metode (k -členske metode)

približek za y_{n+1} izračunamo iz približkov za $y_n, y_{n-1}, \dots, y_{n-k+1}$

in na

1. eksplicitne metode

imamo direktno formulo za izračun približka y_{n+1}

2. implicitne metode

približek za y_{n+1} izračunamo tako, da rešimo (nelinearno) enačbo.

Poglejmo si izpeljavo najbolj preproste enočlenske metode za reševanje začetnega problema 1. reda oblike

$$\begin{aligned}y' &= f(x, y) \\ y(a) &= y_a\end{aligned}$$

Na vsakem 'koraku' moramo iz točke (x_n, y_n) določiti (x_{n+1}, y_{n+1}) . Poznamo začetno točko $x_0 = a$ ter $y_0 = y_a$. Za premik v x smeri uporabimo konstanten korak h (poenostavljen pristop). Vemo torej $x_0 = a$, $x_1 = a + h$, $x_2 = a + 2h$, ...

Kako do približka $y_1 \approx y(x_1)$?

(slika)

Iz DE dobimo vrednost $y'(a)$, to je $y'(a) = f(a, y(a)) = f(x_0, y_0)$. Zdaj znamo narisat tangento na graf funkcije v točki (x_0, y_0) . Dobimo dve novi točki: $(x_1, y(x_1))$, kar je točka na grafu funkcije f , in (x_1, y_1) , kar je točka na tangenti od prej, za njun izračun pa uporabimo

$$\begin{aligned}p(x) &= y(x_0) + (x - x_0)y'(x_0) \\ &= y(x_0) + (x - x_0)f(x_0, y(x_0)) \\ &= y_0 + (x - x_0)f(x_0, y(x_0))\end{aligned}$$

Isto idejo uporabimo za izračun vseh naslednjih približkov in dobimo metodo:

$$y_{n+1} = y_n + hf(x_n, y_n)$$

To metodo imenujemo Eksplicitna Eulerjeva metoda.

Drug način izpeljave te metode:

$$\begin{aligned}y' &= f(x, y) \\ y(a) &= y_a\end{aligned}$$

Spomnimo se formule za aproksimacijo odvoda:

$$y'(x_0) = \frac{y(x_1) - y(x_0)}{h} - \frac{h}{2}y''(\xi)$$

Vstavimo to aproksimacijo v diferencialno enačbo:

$$\begin{aligned}y'(x_n) &= f(x_n, y(x_n)) \\ \frac{y(x_{n+1}) - y(x_n)}{h} - \frac{h}{2}y''(\xi) &= f(x_n, y(x_n))\end{aligned}$$

Napako zanemarimo in točne vrednosti nadomestimo s približki, da ohranimo enačaj:

$$\begin{aligned}\frac{y_{n+1} - y_n}{h} &= f(x_n, y_n) \\ y_{n+1} &= y_n + hf(x_n, y_n)\end{aligned}$$

Dobimo torej isto enačbo kot prej.

Če bi uporabili formulo

$$y'(x_{n+1}) = \frac{y(x_{n+1}) - y(x_n)}{h} + \frac{h}{2}y''(\xi)$$

bi dobili

$$\begin{aligned}y'(x_{n+1}) &= f(x_{n+1}, y(x_{n+1})) \\ \frac{y(x_{n+1}) - y(x_n)}{h} + \frac{h}{2}y''(\xi) &= f(x_{n+1}, y(x_{n+1}))\end{aligned}$$

Odrežemo napako in dobimo

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

oziroma

$$y_{n+1} = y_n + hf(x_{n+1}, y(x_{n+1}))$$

za $n = 0, 1, 2, \dots$. To je implicitna Eulerjeva metoda.

Pri implicitnih metodah moramo za izračun y_{n+1} rešiti (nelinearno) enačbo. Uporabimo lahko postopek navadne iteracije.

$$y_{n+1} = g(y_{n+1})$$

pri čemer je

$$g(y_{n+1}) = y_n + hf(x_{n+1}, y(x_{n+1}))$$

To računamo tako:

```

 $y_{n+1}^{(0)} = y_n$ 
do
   $y_{n+1}^{(r)} = g(y_{n+1}^{(r-1)})$ 
while  $|y_{n+1}^{(r)} - y_{n+1}^{(r-1)}| > \text{toleranca} |y_{n+1}^{(r)}|$ 

```

Primer.

$$y' = x^2 + y^2$$

$$y(0) = 1$$

Z Eulerjevo metodo s korakom $h = \frac{1}{10}$ določi približek za $y_1 \approx y(0.1)$ in $y_2 \approx y(0.2)$.

$$x_0 = 0, x_1 = 0.1, x_2 = 0.2, f(x, y) = x^2 + y^2$$

$$y_0 = 1$$

$$y_1 = y_0 + h(x_0^2 + y_0^2) = 1 + \frac{1}{10}(0 + 1^2) = 1.1$$

$$y_2 = y_1 + h(x_1^2 + y_1^2) = 1.1 + \frac{1}{10}(0.1^2 + 1.1^2) = 1.222$$

V splošnem bi do predpisov za približke y_n lahko prišli na naslednje načine:

1. Odvod y' v DE aproksimiramo s približki izraženimi z vrednostmi funkcije y v bližnjih točkah.
2. DE $y' = f(x, y)$ integriramo.

Primer.

$$\int_{x_n}^{x_{n+1}} y'(x) dx = \int_{x_n}^{x_{n+1}} f(x, y(x)) dx$$

$$y(x_{n+1}) - y(x_n) = \frac{h}{2}(f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))) - \frac{h^3}{12}f^{(2)}(\xi, y(\xi))$$

Napako odrežemo in točne vrednosti nadomestimo s približki in pridemo do metode:

$$y_{n+1} = y_n + \frac{h}{2}(f(x_n, y(x_n)) + f(x_{n+1}, y_{n+1}))$$

To je trapezna metoda za reševanje diferencialnih enačb.

5.1 Globalna in lokalna napaka

Rešujemo problem $y' = f(x, y)$, $y(a) = y_a$. Numerično rešitev iščemo na intervalu $[a, b]$, pri čemer predpostavimo, da so na tem intervalu izpolnjeni pogoji eksistenčnega izreka.

Interval $[a, b]$ razdelimo na m ekvidistantnih delov s korakom $h = \frac{b-a}{m}$. Dobimo $x_n = a + n \cdot h$, $n = 0, 1, \dots, m$. Numerične približke $(y_n)_{n=0}^m$ povežemo z odsekoma linearno funkcijo y_h .

(slikca)

Definicija 5.1. Metoda je konvergentna, če za vse DE, ki zadoščajo pogojem eksistenčnega izreka na $[a, b]$, velja

$$\|y - y_h\|_{\infty, [a, b]} \rightarrow 0$$

ko $h \rightarrow 0$ (oziroma $m \rightarrow \infty$)

Ekvivalentno:

$$\max_{0 \leq n \leq m} |y(x_n) - y_n|$$

ko $m \rightarrow \infty$

Definicija 5.2. Globalna napaka (na celem intervalu):

$$\max_{0 \leq n \leq m} |y(x_n) - y_n|$$

Definicija 5.3. Globalna napaka v točki x_n je $|y(x_n) - y_n|$

Definicija 5.4. Metoda je reda r , če velja

$$\max_{0 \leq n \leq m} |y(x_n) - y_n| = c \cdot h^r + \mathcal{O}(h^{r+1})$$

Pri določanju reda metode oziroma reda globalne napake si pomagamo z lokalno napako.

Definicija 5.5. Lokalna napaka v točki x_n je razlika med točno rešitvijo v tej točki in njenim numeričnim približkom ob predpostavki, da se točna in numerična rešitev ujemata na vseh prejšnjih korakih.

$$T_n(h) = y(x_n) - y_n$$

ob predpostavki

$$y(x_{n-j}) = y_{n-j}, j = 1, 2, \dots$$

Z lokalno napako si pomagamo pri določanju reda globalne napake.

Velja (brez dokaza): Metoda je reda r , brž ko je lokalna napaka reda $r + 1$.

Pri določanju reda lokalne napake si pomagamo z razvoji v Taylorjevo vrsto.

$$\begin{aligned} y(x_{n+1}) &= y(x_n + h) \\ &= y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{6}y'''(x_n) + \mathcal{O}(h^4) \end{aligned}$$

$$y'(x) = f(x, y(x))$$

$$\begin{aligned} y' &= f \\ y'' &= f_x + f_y y' = f_x + f_y f \\ y''' &= f_{xx} + 2f_{xy}f' + f_y(f_x + f_y f) \end{aligned}$$

$$f(x' \Delta x, y + \Delta y) = f(x, y) + f_x(x, y) \cdot \Delta x + f_y(x, y) \cdot \Delta y + \dots$$

Določi red lokalne napake eksplcitne Eulerjeve metode.

$$y_{n+1} = y_n + hf(x_n, y_n)$$

$$\begin{aligned} T_{n+1} &= y(x_{n+1}) - y_{n+1} \\ &= y(x_n) + hf(x_n, y(x_n)) \\ &\quad + \frac{h^2}{2}(f_x(x_n, y(x_n)) \\ &\quad + f_y(x_n, y(x_n)) \cdot f(x_n, y(x_n))) \\ &\quad + \mathcal{O}(h^3) - y_n - hf(x_n, y_n) \end{aligned}$$

Vstavimo $y(x_n) = y_n$ in dobimo

$$T_{n+1} = \frac{h^2}{2}(f_x(x_n, y_n) + f_y(x_n, y_n)f(x_n, y_n)) = \mathcal{O}(h^3)$$

Red lokalne napake je 2, torej je Eulerjeva metoda reda 1.

5.2 Runge-Kutta metode

s-stopenjska R-K metoda: Izračunamo s-koeficientov

$$k_i = hf(x_n + \alpha_i h, y_n + \sum_{j=1}^s \beta_{ij} k_j), i = 1, 2, \dots, s$$

$$y_{n+1} = y_n + \sum_{i=1}^s \gamma_i k_i$$

pri tem so $(\alpha_i)_{i=1}^s$, $(\beta_{ij})_{i,j=1}^s$ in $(\gamma_i)_{i=1}^s$ prosti koeficienti, ki jih izberemo tako, da bo metoda konvergentna in čim višjega reda.

$$\alpha_i \in [0, 1]$$

Velja še $\alpha_i = \sum_{j=1}^s \beta_{ij}$.

Da bo metoda konvergentna oziroma reda vsaj 1, če bo $\sum_{i=1}^s \gamma_i = 1$.

R-K metodo lahko predstavimo v t.i. Butcherjevi shemi

α_1	β_{11}	β_{12}	\dots	β_{1s}
α_2	β_{21}	β_{22}	\dots	β_{2s}
\vdots	\vdots			
α_s	β_{s1}	β_{s2}	\dots	β_{ss}
	γ_1	γ_1	\dots	γ_s

$$k_i = hf(x_n + \alpha_i h, y_n + \beta_{i1} k_1 + \beta_{i2} k_2 + \dots + \beta_{is} k_s)$$

za $i = 1 \dots s$

$$y_{n+1} = y_n + \gamma_1 k_1 + \gamma_2 k_2 + \dots + \gamma_s k_s$$

R-K metoda je eksplisitna, če bodo $\beta_{ij} = 0$ za vse $j \geq i$.

R-K metoda je diagonalno implicitna, če bodo $\beta_{ij} = 0$ za vse $j > i$ in bo vsaj en $\beta_{ii} \neq 0$. Sicer je metoda implicitna.

5.2.1 Primeri R-K metod

(manjka)

5.2.2 Sistemi diferencialnih enačb 1. reda

Začetni problem prvega reda za sistem je oblike

$$\begin{aligned}y'_1 &= f_1(x, y_1, x_2, \dots, y_d) + y_1(a) = y_{1,a} \\y'_2 &= f_2(x, y_1, x_2, \dots, y_d) + y_2(a) = y_{2,a} \\&\vdots \\y'_d &= f_d(x, y_1, x_2, \dots, y_d) + y_d(a) = y_{d,a}\end{aligned}$$

pri čemer za $j \in [1, d]$ velja

$$f_j : [a, b] \times \mathbb{R}^d \rightarrow \mathbb{R}$$

Definiramo

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_d \end{bmatrix}, y : [a, b] \rightarrow \mathbb{R}^d$$
$$F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_d \end{bmatrix}, F : [a, b] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$$

Sistem lahko sedaj zapišemo kot:

$$\begin{aligned}y' &= F(x, Y) \\Y(a) &= Y_a \\Y_a &= \begin{bmatrix} y_{1,a} \\ y_{2,a} \\ \vdots \\ y_{d,a} \end{bmatrix}\end{aligned}$$

Vse metode, ki smo jih spoznali, lahko direktno uporabimo za rešenje sistemov diferencialnih enačb prvega reda.

Uporabimo na primer Eulerjevo metodo:

$$Y_{n+1} = Y_n + hF(x, Y_n)$$

5.2.3 Reševanje diferencialnih enačb višjega reda

Imamo začetni problem reda p :

$$\begin{aligned}y^{(p)} &= f(x, y, y', y'', \dots, y^{(p-1)}) \\y(a) &= y_{a,1} \\y'(a) &= y_{a,2} \\\vdots \\y^{(p-1)}(a) &= y_{a,p}\end{aligned}$$

Problem prevedemo na sistem p diferencialnih enačb prvega reda:

Definiramo “nove” neznane funkcije z_1, z_2, \dots, z_{p-1} :

$$\begin{aligned}z_1 &= y' \\z_2 &= z_1' = y'' \\z_3 &= z_2' = y''' \\\vdots \\z_p &= z_{p-1}' = y^{(p)} \\z_{p-1}' &= y^{(p)} = f(x, y, z_1, z_2, \dots, z_{p-1})\end{aligned}$$

Zgornje funkcije spravimo v vektorsko obliko

$$\begin{bmatrix} y \\ z_1 \\ z_2 \\ \vdots \\ z_{p-2} \\ z_{p-1} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{p-1} \\ f(x, y, z_1, z_2, \dots, z_{p-1}) \end{bmatrix}$$

Zapišemo

$$Y = \begin{bmatrix} y \\ z_1 \\ z_2 \\ \vdots \\ z_{p-2} \\ z_{p-1} \end{bmatrix} = \begin{bmatrix} Y(1) \\ Y(2) \\ \vdots \\ Y(p) \end{bmatrix}$$

Z uporabo Y dobimo funkcijo F

$$F(x, Y) = \begin{bmatrix} Y(2) \\ Y(3) \\ \vdots \\ Y(p) \\ f(x, Y(1), \dots, Y(p)) \end{bmatrix}$$

Primer.

$$\begin{aligned} y'' &= xy' + y^2 \\ y(0) &= 1 \\ y'(0) &= 2 \end{aligned}$$

Reševanje:

$$\begin{aligned} z &= y' \\ z' = y'' &= xz + y^2 \end{aligned}$$

Dobimo sistem:

$$\begin{aligned} \begin{bmatrix} y \\ z \end{bmatrix}' &= \begin{bmatrix} z \\ xz + y^2 \end{bmatrix} \\ Y &= \begin{bmatrix} y \\ z \end{bmatrix}, F(x, y) = \begin{bmatrix} Y[2] \\ xY[2] + Y[1]^2 \end{bmatrix} \\ Y_0 &= \begin{bmatrix} 1 \\ 2 \end{bmatrix} \end{aligned}$$

Izberimo $h = \frac{1}{10}$ in shemo

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$$

Izračunamo

$$\begin{aligned} y_1 &\approx y\left(\frac{1}{10}\right), z_1 \approx y'\left(\frac{1}{10}\right) \\ K_1 &= hF(x_0, Y_0) = \frac{1}{10}F\left(0, \begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) = \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix} \\ K_2 &= hF\left(x_0 + \frac{1}{2}h, Y_0 + \frac{1}{2}K_1\right) = \dots = \begin{bmatrix} 0.205 \\ 0.13125 \end{bmatrix} \\ Y_1 &= Y_0 + K_2 = \begin{bmatrix} 1.205 \\ 2.13125 \end{bmatrix} = \begin{bmatrix} y\left(\frac{1}{10}\right) \\ y'\left(\frac{1}{10}\right) \end{bmatrix} \end{aligned}$$

Zgornji primer rešimo v Matlab-u s kodo:

```
1 F = @(x, y) ([y(2), x + y(2) + y(1)^2])
2
3 function [x, y] = IME(F, a, b, h, y0) % (komentar)
4     x = a:h:b;
5     y = zeros(length(y0), length(x));
6     y(:, 1) = y0;
7     for n=2:length(x)
8         k1 = h * F(x(n-1), y(:, n-1));
9         y(:, n) = y(:, n-1) + k1;
10    end
11 end
```

5.3 Veččlenske metode

k -členska metoda: pri izračunu približka $y_n \approx y(x_n)$ uporabimo k vrednosti:

$$y_{n-1}, y_{n-2}, y_{n-3}, \dots, y_{n-k}$$

Splošna linearna k -členska metoda je oblike:

$$y_n = \sum_{i=1}^k \alpha_i y_{n-i} + h \sum_{i=0}^k \beta_i F(x_{n-i}, y_{n-i})$$

Pri tem so α_i in β_i koeficienti, ki določajo metodo. Določimo jih tako, da je metoda konvergentna in čim višjega reda. Metoda je eksplisitna, če je $\beta_0 = 0$. Sicer je implicitna.

Prednost k -členskih metod je, da so hitrejšje od enočlenskih. Slabost pa je, da spreminjanje koraka h ni enostavno.

Prvih k vrednosti y_0, y_1, \dots, y_{k-1} moramo določiti z neko enočlensko metodo. Pri tem moramo paziti, da izberemo tako, ki ima enak ali višji red kot izbrana k -členska metoda.

Najbolj znani razredi teh metod so (probably na ustnih)

- Adamsove metode:

$$\begin{aligned} y' &= f(x, y) \\ y(a) &= y_a \end{aligned}$$

Ideja je, da to enačbo integriramo:

$$\int_{x_{n-1}}^{x_n}$$

(skica) brez x_n je eksplicitna, z njo pa implicitna. Funkcijo $f(x, y(x))$ nadomestimo z interpolacijskim polinomom. (primer)

- Milneove metode Ideja za izpeljavo: diferencialno enačbo $y' = f(x, y)$ integriramo na $[x_{n-k}, x_n]$, integral funkcije $f(x, y(x))$ pa nadomestimo z izbranim integracijskim pravilom.

– odprta pravila - eksplicitne metode

– zaprta pravila - implicitne metode

(primer)

- BDF metode Odvod $y'(x_n) = f(x_n, y(x_n))$ aproksimiramo z diferen-

cam. (primer? =)

$$y_n = \frac{4}{3}y_{n-1} - \frac{1}{3}y_{n-2} + \frac{2}{3}hf_n$$

Primer.

$$\begin{aligned} y' &= x^2 + y^2 \\ y(0) &= 1 \end{aligned}$$

Z dvočlensko eksplicitno Adamsovo metodo določi približek za $y(0, 2)$ s $h = \frac{1}{10}$.

$$f(x, y) = x^2 + y^2, x_0 = 0, y_0 = 1, f_0 = 1$$

Uporabimo $h = \frac{1}{10}$:

$x_1 = \frac{1}{10}$: y_1 določimo z modificirano Eulerjevo metodo in dobimo $y_1 = 1,1105$

$x_2 = \frac{2}{10}$:

$$y(0, 2) = y_2 = y_1 + h\left(-\frac{1}{2}f_0 + \frac{3}{2}f_1\right) = 1,24698$$

6 Numerično računanje lastnih vrednosti

Naj bo $A \in \mathbb{C}^{n \times n}$. Če neničeln vektor $x \in \mathbb{C}^n$ in skalar $\lambda \in \mathbb{C}$ zadoščata enačbi

$$Ax = \lambda x$$

Potem je x (desni) lastni vektor za lastno vrednost λ .

Levi lastni vektorji:

$$y^H A = \lambda y^H$$

Če to hermitsko transponiramo:

$$A^H y = \tilde{\lambda} y$$

Matriko A se da diagonalizirati, če obstajata nesingularna matrika $X = [x_1, x_2, \dots, x_n]$ in diagonalna matrika $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, da velja

$$A = X \Lambda X^{-1}$$

Če je S nesingularna matrika, potem sta A in $B = S^{-1} A S$ podobni matriki.

$$\begin{aligned} Ax &= \lambda x \\ S^{-1} A S S^{-1} &= S^{-1} \lambda x \\ B(S^{-1}x) &= \lambda(S^{-1}x) \end{aligned}$$

Lastne vrednosti so ničle karakterističnega polinoma

$$p(\lambda) = \det(A - \lambda I)$$

Algebraična in geometrijska večkratnost lastne vrednosti:

- algebraična - večkratnost lastne vrednosti v karakterističnem polinomu
- geometrijska - dimenzija jedra $A - \lambda I$

Ker so lastne vrednosti v bistvu ničle polinoma, so vsi algoritmi za računanje lastnih vrednosti iterativni. Numerično lastnih vrednosti ne računamo preko ničel karakterističnega polinoma. Zakaj? Ker je že računanje koeficientov karakterističnega polinoma težavno, ničle polinomov pa so zelo občutljive na minimalne spremembe koeficientov. Računanje preko karakterističnih polinomov je torej nestabilno.

Računanje Jordanove kanonične forme ni obratno stabilno.

6.1 Schurova forma

Izrek 6.1. Za vsako $n \times n$ matriko A obstajata unitarna matrika U in zgornje trikotna matrika T , da je

$$A = UTU^H$$

Temu pravimo Schurov razcep, T pa imenujemo Schurova forma.

Dokaz. Z indukcijo na dimenzijo matrike

$$\begin{aligned} A &= [a_{11}], \\ A &= [1][a_{11}][1]^H \end{aligned}$$

Recimo, da trditev velja za vse $(n-1) \times (n-1)$ matrike. Naj bo A $n \times n$ matrika. Naj bo x njen normiran lastni vektor in λ pripadajoča lastna vrednost

$$Ax = \lambda x$$

Naj bo U_1 taka unitarna matrika, da je $U_1^H x = e_1$. Naj bo $B = U_1^H A U_1$.

$$B e_1 = U_1^H A U_1 e_1 = \lambda U_1^H x = \lambda e_1$$

torej je B oblike

$$B = \begin{bmatrix} \lambda & \mathbf{d}^T \\ 0 & \mathbf{C} \\ \vdots & \\ 0 & \end{bmatrix} = \begin{bmatrix} 1 & \\ & V_1 \end{bmatrix} \cdot \begin{bmatrix} \lambda & \\ & T_1 \end{bmatrix} \cdot \begin{bmatrix} 1 & \\ & V_1^H \end{bmatrix}$$

Po I.P. obstaja Schurov razcep matrike $C = V_1 T_1 V_1^H$.

$$\begin{aligned} B &= V T V^H \\ U_1^H A U_1 &= V T V^H / U_1^H \\ A &= (U_1 V) T (U_1 V)^H \\ A &= U T U^H \end{aligned}$$

■

Schurova forma ni enolična. Vrstni red diagonalnih elementov je lahko poljuben.

Tudi če je $A \in \mathbb{R}^{n \times n}$, je lahko njen schurov razcep kompleksen. Da se izognemo kompleksni aritmetiki, uporabimo tako imenovani Realni Schurov razcep.

Izrek 6.2. Za vsako matriko $A \in \mathbb{R}^{n \times n}$ obstajata ortogonalna matrika Q in kvazi-zgornje trikotna matrika R , da je

$$A = QRQ^T$$

To imenujemo realni Schurov razcep, R pa imenujemo realna Schurova forma.

Opomba. Matrika je kvazi-zgornje trikotna, če je bločno zgornje trikotna, pri čemer dopuščamo na diagonalni 2×2 bloke, katerih lastne vrednosti ustrezajo parom kompleksno konjugiranih lastnih vrednosti.

Dokaz. Izpustimo. ■

6.2 Metode za računanje lastnih vrednosti

- Potenčna metoda
- Inverzna iteracija (inverzna potenčna metoda)
- Ortogonalna iteracija
- QR metoda

6.2.1 Potenčna metoda

Za dano matriko $A \in \mathbb{R}^{n \times m}$ iščemo dominantno lastno vrednost in pripadajoč lastni vektor.

Opomba. Dominantna lastna vrednost je tista, ki je po absolutni vrednosti največja.

Dobimo zaporedje vektorjev z_0, z_1, z_2, \dots

Izrek 6.3. Naj bo $A \in \mathbb{R}^{n \times m}$ in naj bo $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \geq 0$ njene lastne vrednosti. Potem za poljubni začetni vektor z_0 zaporednje vektorjev $(z_k)_k$, ki ga tvori potenčna metoda, konvergira (numerično) proti dominantnemu lastnemu vektorju, to je lastnemu vektorju za λ_1 .

Algorithm 3 Osnovna različica algoritma

```
1:  $z_0 \in \mathbb{R}^n$ 
2:  $k \leftarrow 0$ 
3: while true do
4:    $y_{k+1} = Az_k$ 
5:    $z_{k+1} = \frac{y_{k+1}}{\|y_{k+1}\|}$ 
6:    $k \leftarrow k + 1$ 
7:   if  $\|Az_k - \rho_k z_k\| < \epsilon$  then ▷ Stop condition
8:     break
9:   end if
10: end while
```

Dokaz. Izrek velja za poljubno matriko, dokazali pa ga bomo za matriko A , ki se da diagonalizirati. To je

$$A = X\Lambda X^{-1}, \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n), X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{n \times m}$$

Začetni vektor z_0 razvijemo po bazi lastnih vektorjev:

$$\begin{aligned} z_0 &= \sum_{i=1}^n \alpha_i x_i \\ z_k &= \frac{Az_{k-1}}{\|Az_{k-1}\|} = \frac{A(\frac{Az_{k-1}}{\|Az_{k-1}\|})}{A(\frac{Az_{k-2}}{\|Az_{k-2}\|})} = \dots \frac{A^k z_0}{\|A^k z_0\|} = (*) \\ A^k z_0 &= \sum_{i=1}^n \alpha_i A^k x_i = \sum_{i=1}^n \alpha_i \lambda_i^k x_i \\ z_k &= \frac{\alpha_1 \lambda_1^k x_1 + \sum_{i=1}^n \alpha_i \lambda_i^k x_i}{\|\alpha_1 \lambda_1^k x_1 + \sum_{i=1}^n \alpha_i \lambda_i^k x_i\|} = \pm \frac{\alpha_1 x_1 + \sum_{i=1}^n \alpha_i (\frac{\lambda_i}{\lambda_1})^k x_i}{\|\alpha_1 x_1 + \sum_{i=1}^n \alpha_i (\frac{\lambda_i}{\lambda_1})^k x_i\|} \\ &\xrightarrow{k \rightarrow \infty} \pm \frac{\alpha_1 x_1}{\|\alpha_1 x_1\|} = \pm \frac{x_1}{\|x_1\|} \end{aligned}$$

Pri čemer mora veljati, da $\alpha_1 \neq 0$. ■

Opomba. V izreku nismo navedli predpostavke, da mora biti $\alpha_1 \neq 0$, saj je ta predpostavka vedno izpolnjena zaradi zaokrožitvenih napak.

Potrebujemo še približek za lastno vrednost (da bomo lahko ugotovili, ali je postopek že skonvergiral).

Naj bo x približek za lastni vektor.

$$Ax = \lambda x$$

Imamo sistem n enačb z eno neznanko, torej predoločen sistem enačb. Zapišemo ga lahko kot

$$x\lambda = Ax$$

$$\begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} Ax \\ x\lambda \end{bmatrix}$$

Rešitev po metodi najmanjših kvadratov:

$$\min_{\lambda \in \mathbb{R}} \|Ax - \lambda x\|_2$$

$$x^T x \lambda = x^T Ax$$

$$\lambda = \frac{x^T Ax}{x^T x}$$

Označimo $\rho(x, A) = \frac{x^T Ax}{x^T x}$, kar imenujemo Rayleighov kvocient.

Zaustavitveni kriterij pri potenčni metodi:

$$z_k$$

$$\rho_k = z_k^T A z_k$$

$$\|A z_k - \rho_k z_k\| < \textit{toleranca}$$

Algorithm 4 Potenčna metoda

```

1: Input:  $A$ , toleranca  $tol$ 
2: Izberi normiran vektor  $z_0$ 
3:  $y_1 = A z_0$ 
4:  $\rho_0 = z_0^T y_1$ 
5:  $k = 0$ 
6: while ( $\|y_{k+1} - \rho_k z_k\| > tol$ ) and ( $k < \text{max\_steps}$ ) do
7:    $k = k + 1$ 
8:    $z_k = \frac{y_k}{\|y_k\|}$ 
9:    $y_{k+1} = A z_k$ 
10:   $\rho_k = z_k^T y_{k+1}$ 
11: end while

```

Konvergenca potenčne metode je linearna in je odvisna od razmerja $|\frac{\lambda_1}{\lambda_2}|$.

6.2.2 Inverzna iteracija (inverzna potenčna metoda)

Naj bo σ približek za lastno vrednost. Kako bi izračunali tisto lastno vrednost, ki je najbližja σ in pripadajoč lastni vektor.

Naj velja

$$\|\lambda_i - \sigma\| < \|\lambda_j - \sigma\|, j \neq i$$

Poglejmo si matriki $(A - \sigma I)^{-1}$. Naj velja $A = X\Lambda X^{-1}$.

$$\begin{aligned} (A - \sigma I)^{-1} &= (X\Lambda X^{-1} - \sigma X X^{-1})^{-1} \\ &= X(\Lambda - \sigma I^{-1})^{-1} X^{-1} \\ &= X \begin{bmatrix} \frac{1}{\lambda_1 - \sigma} & & & \\ & \ddots & & \\ & & \frac{1}{\lambda_i - \sigma} & \\ & & & \ddots \\ & & & & \frac{1}{\lambda_n - \sigma} \end{bmatrix} X^{-1} \end{aligned}$$

Vidimo, da je $\frac{1}{\lambda_i - \sigma}$ dominantna lastna vrednost za $(A - \sigma I)^{-1}$. Lastni vektorji matrike A in $(A - \sigma I)^{-1}$ so enaki.

Algorithm 5 Inverzna iteracija

- 1: Izberi normiran vektor z_0 (normaliziran).
 - 2: **while** $k = 0, 1, 2, \dots$ **do**
 - 3: $y_{k+1} = (A - \sigma I)^{-1} z_k$ \triangleright numerično: rešimo sistem linearnih enačb
 $(A - \sigma I)y_{k+1} = z_k$.
 - 4: $z_{k+1} = \frac{y_{k+1}}{\|y_{k+1}\|}$
 - 5: **end while**
-

Opomba. Če inverzno iteracijo kombiniramo z Rayleighovim kvocientom dobimo Rayleighovo iteracijo

Algorithm 6 Rayleighovo iteracija

- 1: Izberi normiran vektor z_0
 - 2: **while** $k = 0, 1, 2, \dots$ **do**
 - 3: $\sigma_k = \rho(z_k, A)$
 - 4: Reši sistem $(A - \sigma_k I)y_{k+1} = z_k$
 - 5: $z_{k+1} = \frac{y_{k+1}}{\|y_{k+1}\|}$
 - 6: **end while**
-

6.2.3 Ortogonalna iteracija

Podprostor $N \subseteq \mathbb{R}^n$ je invarianten za $A \in \mathbb{R}^{n \times n}$, če je $AN \subseteq N$.

Naj za A velja $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_p| > |\lambda_{p+1}| \geq \dots \geq |\lambda_n| \geq 0$

Podprostor $\text{Lin}\{x_1, x_2, \dots, x_p\}$ se imenuje dominantni invariantni podprostor. Izračunamo ga z ortogonalno iteracijo.

Algorithm 7 Ortogonalna iteracija

Require: Matrika $A \in \mathbb{R}^{n \times n}$, $p \in \{1, 2, \dots, n\}$

- 1: Izberi začetno matriko $T_0 \in \mathbb{R}^{n \times p}$ z ortonormiranimi stolpci
 - 2: **while** $k = 0, 1, 2, \dots$ **do**
 - 3: $Y_{k+1} = AZ_k$
 - 4: Izračunaj QR razcep matrike Y_{k+1} , torej $Y_{k+1} = QR$
 - 5: $Z_{k+1} = Q$
 - 6: **end while**
-

Izrek 6.4. Naj bo $A = X\Lambda X^{-1}$, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, $X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$ in naj velja $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_p| > |\lambda_{p+1}| \geq \dots \geq |\lambda_n| \geq 0$. Potem za poljubno začetno matriko $Z_0 \in \mathbb{R}^{n \times p}$, $Z_0^T Z_0 = I$, zaporedje matrik $(Z_k)_k$ konvergira protu matriki, katere stolpci tvorijo bazo za dominantni invariantni podprostor $\text{Lin}\{x_1, x_2, \dots, x_p\}$.

Izrek 6.5. Naj bo $p = n$ in naj velja $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| \geq 0$. Definiramo matriko $A_k = Z_k^T A Z_k \in \mathbb{R}^{n \times n}$. Potem zaporedje matrike $(A_k)_k$ konvergira proti zgornje trikotni matriki (iz Schurovega razcepa).

Dokaz. Izpustimo. ■

6.2.4 QR iteracija (osnovna različica)

Algorithm 8 QR iteracija

Require: Matrika A

- 1: $A_0 = A$
 - 2: **while** $k = 0, 1, 2, \dots$ **do**
 - 3: Izračunaj OR razcep matrike A_k : $A_k = Q_k R_k$
 - 4: $A_{k+1} = R_k Q_k$
 - 5: **end while**
-

Matriki A_k in A_{k+1} sta ortogonalno podobni matriki:

$$A_{k+1} = R_k Q_k = Q_k^T A_k Q_k$$

Notri smo vstavili

$$Q_k^T A_k = Q_k^T Q_k R_k = I R_k$$

QR iteracija tvori zaporedje ortogonalnih podobnih matrik, ki so ortogonalno podobne matriki A .

Izrek 6.6. Za matriko A_k iz QR iteracije velja, da je enaka matriki $Z_k^T A Z_k$, kjer je Z_k matrika, ki jo dobimo pri postopku ortogonalne iteracije, če postopek začnemo z matriko $Z_0 = I$.

Dokaz. Z indukcijo in primerjavo obeh algoritmov. ■

Posledica 6.7. Če velja $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| \geq 0$, potem zaporedje matrik $(A_k)_k$ konvergira proti zgornje trikotni matriki.

Zaustavitveni kriterij:

$$\max_{1 \leq j < i \leq n} |a_{ij}| < tol$$

Za izračun QR razcepa $n \times n$ matrike potrebujemo $\mathcal{O}(n^3)$ operacij.

Izboljšave:

1. Redukcija na zgornje Hessenbergovo matriko
2. Uporaba premikov

Zgornje Hessenbergova matrika je matrika z ničelnimi elementi pod prvo poddiagonalno ($a_{ij} = 0$ za $i \geq j + 2$).

Opomba. Ta oblika se tekom QR iteracije ohranja.

Vsako matriko A se da z direktnim algoritmom pretvoriti v ortogonalno podobno zgornje Hessenbergovo matriko:

$$\begin{aligned} A &= Q H Q^T \\ Q Q^T &= Q^T Q = I \end{aligned}$$

Do tega pridemo z uporabo Householderjevih zrcaljenj.

6.2.4.1 Householderjeva zrcaljenja

A zapišemo kot

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

Na podlagi rdeče obkroženega vektorja določimo Householderjevo zrcaljenje, ki uniči elemente od prvega navzdol, torej od drugega navzdol v celotnem stolpcu. Dobimo \tilde{P}_1 , ki je 4×4 matrika.

$P_1 A$ izgleda tako:

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix}$$

Obkroženi elementi se pri tem ne spremenijo.

Iz tega sledi, da $(P_1 A) P_1^T$ izgleda tako:

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix}$$

Prvi stolpec se pri tem ne spremeni.

Postopek nadaljujemo. Za $n \times n$ matriko potrebujemo $n-2$ korakov. Dobimo:

$$P_3 P_2 P_1 A P_1^T P_2^T P_3^T = Q A Q^T = H$$

Potrebujemo $\mathcal{O}(n^2)$ operacij.

6.2.4.2 Premiki

Algorithm 9 QR iteracija s premiki

Require: Zgornje Hessenbergova matrika H

- 1: $A_0 = H$
 - 2: **while** $k = 0, 1, 2, \dots$ **do**
 - 3: Izberi premik σ_k
 - 4: Izračunaj QR razcep matrike A_k : $A_k - \sigma_k I = Q_k R_k$
 - 5: $A_{k+1} = R_k Q_k + \sigma_k I$
 - 6: **end while**
-

Enostavni premik:

$$\sigma_k = A_k(n, n)$$

Dvojni premik Vzamemo 2×2 matriko v spodnjem desnem kotu, izračunamo lastne vrednosti in iz njih dobimo σ . (ne vem, kako)

Če se na prvi poddiagonali v zgornji Hessenbergovi matriki pojavi 0, problem razpade na dva problema.