

# Uvod v numerične metode

Ruslan Urazbakhtin

19. januar 2026

## Kazalo

<b>1</b>	<b>Uvod</b>	<b>3</b>
1.1	Absolutna in relativna napaka . . . . .	3
1.2	Predstavljiva števila . . . . .	3
1.3	Občutljivost problema . . . . .	4
1.4	Vrste napak pri numeričnem računanju . . . . .	5
1.5	Stabilnost numerične metode . . . . .	5
<b>2</b>	<b>Nelinearne enačbe</b>	<b>7</b>
2.1	Uvod . . . . .	7
2.2	Bisekcija . . . . .	7
2.3	Navadna iteracija . . . . .	8
2.4	Tangentna metoda . . . . .	10
2.5	Metode brez računanja odvoda . . . . .	11
2.6	Ničle polinomov . . . . .	12
<b>3</b>	<b>Sistemi linearnih enačb</b>	<b>14</b>
3.1	Uvod . . . . .	14
3.2	Vektorske in matrične norme . . . . .	16
3.3	Občutljivost sistemov linearnih enačb . . . . .	19
3.4	LU razcep . . . . .	20
3.5	Analiza zaokrožitvenih napak pri LU razcepu . . . . .	24
3.6	Sistemi posebne oblike . . . . .	24
<b>4</b>	<b>Sistemi nelinearnih enačb</b>	<b>25</b>
4.1	Newtonova metoda . . . . .	25
4.2	Kvazi Newtonove metode . . . . .	26
<b>5</b>	<b>Linearni problemi najmanjših kvadratov</b>	<b>28</b>
5.1	Uvod . . . . .	28
5.2	QR razcep . . . . .	28
5.3	Givensove rotacije . . . . .	29
5.4	Householderjeva zrcaljenja . . . . .	31
<b>6</b>	<b>Problemi lastnih vrednosti</b>	<b>32</b>
6.1	Potenčna metoda . . . . .	32
6.2	Inverzna iteracija . . . . .	34
6.3	Ortogonalna iteracija . . . . .	34
6.4	QR iteracija . . . . .	35
6.4.1	Redukcija na Hessenbergovo obliko . . . . .	35
6.4.2	Premiki . . . . .	36
<b>7</b>	<b>Polinomska interpolacija</b>	<b>38</b>
7.1	Uvod . . . . .	38
7.2	Lagrangeva interpolacija . . . . .	38
7.3	Deljene difference . . . . .	39

## 1 Uvod

*Numerična metoda* je postopek, ki iz danih numeričnih podatkov s končnim številom elementarnih operacij:  $+$ ,  $-$ ,  $/$ ,  $*$ ,  $\sqrt{\phantom{x}}$ , izračuna numerični rezultat.

### 1.1 Absolutna in relativna napaka

**Definicija 1.1.** *Napaka približka* je razlika med približkom  $\hat{x}$  in točno vrednostjo  $x$ .

- *Absolutna napaka* je  $d_a = \hat{x} - x$ .
- *Relativna napaka* je  $d_r = \frac{\hat{x} - x}{x}$ .

### 1.2 Predstavljliva števila

V računalniku so *predstavljliva* števila zapisana v premični piki kot

$$x = \pm m \cdot b^e,$$

kjer je  $m = 0.c_1c_2 \dots c_t$  *mantisa* in

- $b$ : baza (običajno 2 ali 10),
- $t$ : dolžina mantise,
- $e$ : eksponent v mejah  $L \leq e \leq U$ ,
- $c_i$ : številke v mejah od 0 do  $b - 1$ .

Če je  $c_1 \neq 0$ , potem je število *normalizirano*, sicer pa *subnormalizirano*. Zahtevamo, da lahko  $c_1 = 0$ , samo če  $e = L$ . Zapis označimo s  $P(b, t, L, U)$ .

**Zgled 1.2.** Naj bo  $x \in P(b, t, L, U)$ . Tedaj

$$x = \pm(c_1b^{-1} + c_2b^{-2} + \dots + c_tb^{-t}) \cdot b^e.$$

Na primer

$$0.1101_2 \cdot 2^2 = (2^{-1} + 2^{-2} + 2^{-4}) \cdot 2^2 = 3.25.$$

### Standard IEEE

- *single*:  $P(2, 24, -125, 128)$ .
- *double*:  $P(2, 53, -1021, 1023)$ .

### Zaokrožanje

Naj bo  $x$  pozitivno število z neskončnim zapisom

$$x = 0.d_1d_2 \dots d_t d_{t+1} \dots \cdot b^e.$$

Kandidata za predstavljliv približek, ki ga označimo s  $\text{fl}(x)$ , sta najbližji predstavljlivi števili z leve in z desne

$$\begin{aligned} x_- &= 0.d_1d_2 \dots d_t \cdot b^e, \\ x_+ &= (0.d_1d_2 \dots d_t + b^{-t}) \cdot b^e. \end{aligned}$$

Pri standardu IEEE uporabimo *zaokrožanje* in za  $\text{fl}(x)$  izberemo predstavljlivo število, ki je najbližje  $x$ .

### Osnovna zaokrožitvena napaka

**Izrek 1.3.** Če za število  $x$  velja, da  $|x|$  leži na intervalu med najmanjšim in največjim pozitivnim predstavljenim normaliziranim številom, potem velja

$$\frac{|\text{fl}(x) - x|}{|x|} \leq u,$$

kjer je  $u = \frac{1}{2}b^{1-t}$  osnovna zaokrožitvena napaka.

Dokaz. **TODO**

□

Velja

$$\text{fl}(x) = x(1 + \delta) \text{ za } |\delta| \leq u$$

- single:  $u = 2^{-24} \approx 6 \cdot 10^{-8}$ ,
- double:  $u = 2^{-53} \approx 1 \cdot 10^{-16}$ .

### Računanje po standardu IEEE

Velja *pravilo korektnega zaokroževanja*: Če na dveh predstavljenih številih izvedemo osnovno računsko operacijo in je rezultat spet v intervalu predstavljenih števil, dobimo isto, kot če bi zaokrožili točen rezultat.

Če sta  $x, y$  predstavljeni števili in je rezultat znotraj normaliziranih predstavljenih števil, potem velja:

- $\text{fl}(x \oplus y) = (x \oplus y)(1 + \delta)$ ,  $|\delta| \leq u$ ,
- $\text{fl}(\sqrt{x}) = \sqrt{x}(1 + \delta)$ ,  $|\delta| \leq u$ .

Izjeme so:

- če pride do *prekoračitve* (overflow) obsega predstavljenih števil, dobimo  $\pm\infty$ ,
- če pride do *podkoračitve* (underflow) obsega predstavljenih števil, dobimo 0,
- če dopuščamo subnormalizirana števila in je  $\text{fl}(x \oplus y)$  subnormalizirano število, lahko  $|\delta|$  naraste v najslabšem primeru do  $\frac{1}{2}$ .

### 1.3 Občutljivost problema

Če se rezultat pri majhni spremembi argumentov (motnji oz. perturbaciji) ne spremeni veliko, je problem *neobčutljiv*, sicer pa je *občutljiv*.

#### Stopnja občutljivosti

Občutljivost merimo s supremumom razmerja med spremembo rezultata in spremembo podatkov, ko gre sprememba podatkov proti 0.

**Zgled 1.4.** Naj bo  $f : \mathbb{R} \rightarrow \mathbb{R}$  odvedljiva funkcija. Zanima nas razlika med  $f(x)$  in  $f(x + \delta x)$ , kjer je  $\delta x$  majhna motnja.

Velja (diferencial)

$$|f(x + \delta x) - f(x)| \approx |f'(x)| \cdot |\delta x|,$$

torej je  $|f'(x)|$  *absolutna občutljivost*  $f$  v točki  $x$ .

Za oceno relativne napake dobimo

$$\frac{|f(x + \delta x) - f(x)|}{|f(x)|} \approx \frac{|f'(x)| \cdot |x|}{|f(x)|} \cdot \frac{|\delta x|}{|x|},$$

torej je  $\frac{|f'(x)| \cdot |x|}{|f(x)|}$  *relativna občutljivost*  $f$  v točki  $x$ .

## 1.4 Vrste napak pri numeričnem računanju

Imamo tri vrste napak:

- *Neodstranljiva napaka*: ker začetni podatki niso točni.
  - Namesto  $x$  računamo s približkom  $\bar{x}$  in zato namesto  $y = f(x)$  izračunamo  $\bar{y} = f(\bar{x})$ . Neodstranljiva napaka je

$$D_n = y - \bar{y}.$$

- *Napaka metode*: ker metoda s katero računamo, ni točna.
  - Namesto  $f$  računamo vrednost funkcije  $g$ , ki jo lahko izračunamo s končnim številom operacij. Namesto  $\bar{y} = f(\bar{x})$  tako izračunamo  $\tilde{y} = g(\bar{x})$ . Napaka metode je

$$D_m = \bar{y} - \tilde{y}.$$

- *Zaokrožitvena napaka*: ker pri vseh vmesnih izračunih zaokrožujemo.
  - Pri računanju  $\tilde{y} = g(\bar{x})$  se pri vsaki računski operaciji pojavi zaokrožitvena napaka, tako da namesto  $\tilde{y}$  izračunamo  $\hat{y}$ . Zaokrožitvena napaka je

$$D_z = \tilde{y} - \hat{y}.$$

Celotna napaka je

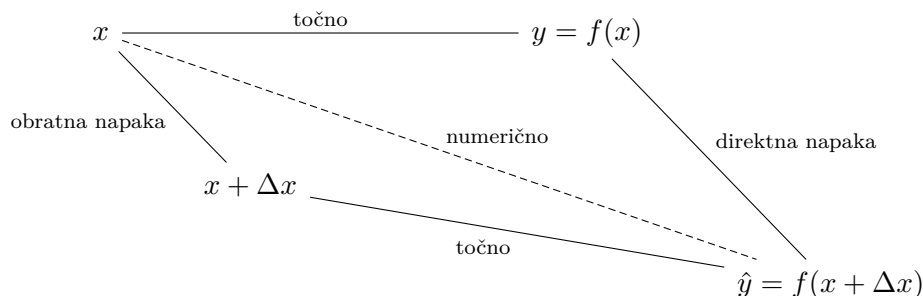
$$D = D_n + D_m + D_z.$$

Velja

$$|D| \leq |D_n| + |D_m| + |D_z|.$$

## 1.5 Stabilnost numerične metode

Numerična metoda iz  $x$  namesto  $y = f(x)$  izračuna  $\hat{y}$ .



Metoda je *natančna*, če je za vsak  $x$  direktna napaka majhna. Torej vedno dobimo bližnji odgovor.

Če za vsak  $x$  obstaja tak  $\hat{x} = x + \Delta x$  blizu  $x$  (absolutno oz. relativno), da je  $f(\hat{x}) = \hat{y}$ , je metoda *obratno stabilna* (absolutno oz. relativno). Obratno stabilna metoda vedno vrne točen odgovor na bližnje vprašanje.

### Povezava med občutljivostjo, direktno in obratno napako

Velja

$$|\text{direktna napaka}| \leq \text{občutljivost} \cdot |\text{obratna napaka}|$$

Torej, če je metoda obratno stabilna, je direktna napaka omejena s produktom občutljivosti in nekaj majhnega.

Če je za vsak  $x$  direktna napaka omejena s produktom občutljivosti in nekaj majhnega, je metoda *direktno stabilna*.

Obratno stabilna metoda je tudi direktno stabilna, obratno pa ni nujno res.

Metoda je *stabilna*, če je obratno ali direktno stabilna.

## 2 Nelinearne enačbe

### 2.1 Uvod

Iščemo rešitve (ničle) enačbe  $f(x) = 0$ , kjer je  $f : \mathbb{R} \rightarrow \mathbb{R}$  ali  $f : \mathbb{C} \rightarrow \mathbb{C}$ . Lahko imamo eno ničlo, več ničel, neskončno ničel ali nič ničel.

Naj bo  $f$  zvezno odvedljiva funkcija v okolici  $\alpha$  in  $f(\alpha) = 0$ .

- Če je  $f'(\alpha) \neq 0$ , je  $\alpha$  *enostavna ničla*.
- Če je  $f'(\alpha) = 0$ , je  $\alpha$  *večkratna ničla*.
  - Če je  $f \in C^m$  v okolici  $\alpha$  in  $f'(\alpha) = \dots = f^{(m-1)}(\alpha) = 0$ ,  $f^{(m)}(\alpha) \neq 0$ , je  $\alpha$  *m-kratna ničla*.

### Občutljivost ničel

Naj bo  $x$  približek za ničlo  $\alpha$  in  $|f(x)| \leq \varepsilon$ . Naj bo  $\alpha$  enostavna ničla funkcije  $f \in C^1$ . Vemo, da je  $f'(\alpha) \neq 0$ . Po Lagrangeovem izreku:

$$\begin{aligned} |f(x) - f(\alpha)| &= |f'(c)| |x - \alpha| \approx |f'(\alpha)| |x - \alpha| \\ \Rightarrow |x - \alpha| &\lesssim \frac{\varepsilon}{|f'(\alpha)|} \\ \Rightarrow \frac{1}{|f'(\alpha)|} &\text{ je občutljivost ničle } \alpha. \end{aligned}$$

Po drugi strani, vemo, da je občutljivost izračuna funkcije enaka absolutne vrednosti odvoda, tj.

$$\alpha = f^{-1}(0) \Rightarrow |(f^{-1})'(0)| = \frac{1}{|f'(\alpha)|}.$$

Naj bo zdaj  $\alpha$  dvojna ničla, torej  $f(\alpha) = f'(\alpha) = 0$ ,  $f''(\alpha) \neq 0$ . Tedaj

$$\begin{aligned} f(x) &= f(\alpha) + f'(\alpha)(x - \alpha) + \frac{f''(c)}{2}(x - \alpha)^2 \\ \Rightarrow |x - \alpha| &\lesssim \sqrt{\frac{2\varepsilon}{|f''(\alpha)|}} = O(\varepsilon^{1/2}). \end{aligned}$$

Torej potrebujemo manjši  $\varepsilon$ , da dobimo dobro aproksimacijo.

### 2.2 Bisekcija

**Izrek 2.1** (o bisekciji). *Naj bo  $f : [a, b] \rightarrow \mathbb{R}$  zvezna in  $f(a) \cdot f(b) \leq 0$ . Tedaj obstaja  $c \in (a, b)$ , da je  $f(c) = 0$ .*

**Algorithm 1:** Bisekcija**Data:**  $a, b \in \mathbb{R}$ ,  $f \in C[a, b]$ ,  $\varepsilon \in (0, \infty)$ **Result:**  $c \in (a, b)$ , da je  $f(c) \approx 0$  $e \leftarrow b - a$ **while**  $e > \varepsilon$ /\*  $b - a > \varepsilon$  \*/**do**     $e \leftarrow e/2$      $c \leftarrow a + e$     **if**  $\text{sgn}(f(a)) = \text{sgn}(f(c))$  **then**         $a \leftarrow c$     **else**         $b \leftarrow c$     **end****end****Opomba 2.2.**

- Ne preverjamo, če je  $f(c) = 0$ , saj je to zelo redek dogodek.
- Uporabljamo  $e = e/2$  namesto  $c = (a + b)/2$ , da smo gotovi, da se postopek ustavi.

**Analiza števila korakov.** Program se ustavi, ko

$$(b - a) \cdot 2^{-k} < \varepsilon.$$

Sledi, da

$$k = \left\lceil \log_2 \left( \frac{b - a}{\varepsilon} \right) \right\rceil.$$

Torej je število korakov odvisno le od  $\varepsilon$  in začetne širine intervala.

Opazimo, da se napaka v vsakem koraku razpolovi, kar je tipičen primer linearne konvergence.

**2.3 Navadna iteracija**Iščemo rešitve enačbe  $f(x) = 0$ . Enačbo predelamo v ekvivalentno obliko  $g(x) = x$ , tj.

$$f(x) = 0 \text{ (} x \text{ je ničla } f) \Leftrightarrow x = g(x) \text{ (} x \text{ je negibna točka } g).$$

**Zgled 2.3.** Naj bo  $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$  funkcija. Lahko definiramo

- $g(x) := f(x) - x$ .
- $g(x) := x - c \cdot f(x)$ ,  $c \neq 0$ .
- $g(x) := x - h(x) \cdot f(x)$ ,  $h(x) \neq 0$ .

**Postopek.** Izberimo začetni približek  $x_0$  in računamo

$$x_{r+1} = g(x_r), \quad r = 0, 1, \dots$$

To je *navadna iteracija* in  $g$  je *iteracijska funkcija*.



Ob ustrezno izbrani iteracijske funkcije  $g$  in dobrem začetnem približku je  $\lim_{r \rightarrow \infty} x_r = \alpha$ , kjer je

$$\alpha = g(\alpha) \quad \text{oz.} \quad f(\alpha) = 0.$$

Da navadna iteracija deluje za ničlo  $\alpha$ , mora biti  $g$  skrčitev na neki okolici  $\alpha$ , tj.

$$|g(x) - g(y)| \leq m|x - y|, \quad m < 1.$$

**Izrek 2.4.** Naj bo  $\alpha = g(\alpha)$  in naj bo  $\alpha$  na  $I = [\alpha - \delta, \alpha + \delta]$  za nek  $\delta > 0$  zadošča Lipschitzovem pogoju, tj.

$$|g(x) - g(y)| \leq m|x - y| \quad \text{za } 0 \leq m < 1 \quad \text{za vse } x, y \in I.$$

Tedaj za vsak  $x_0 \in I$  zaporedje  $x_{r+1} = g(x_r)$  za  $r = 0, 1, \dots$  konvergira k  $\alpha$  in velja:

1.  $|x_r - \alpha| \leq m^r |x_0 - \alpha|$ ;
2.  $|x_{r+1} - \alpha| \leq \frac{m}{1-m} |x_{r+1} - x_r|$ .

Dokaz. **TODO**

□

**Opomba 2.5.** Ocena  $|x_{r+1} - \alpha| \leq \frac{m}{1-m} |x_{r+1} - x_r|$  nam pove, kako daleč smo od negibne točke.

**Posledica 2.6.** Naj bo  $\alpha = g(\alpha)$ . Naj bo  $g$  zvezno odvedljiva v okolici  $\alpha$  in  $|g'(\alpha)| < 1$ . Tedaj obstaja  $\delta > 0$ , da za vsak  $x_0 \in I = [\alpha - \delta, \alpha + \delta]$  zaporedje  $x_{r+1} = g(x_r)$  za  $r = 0, 1, \dots$  konvergira k  $\alpha$ .

Dokaz. **TODO**

□

Naj bo  $\alpha = g(\alpha)$ . Naj bo  $g$  zvezno odvedljiva v okolici  $\alpha$  in  $|g'(\alpha)| < 1$ :

- Če je  $|g'(\alpha)| < 1$ , potem je  $\alpha$  *privlačna* negibna točka.
- Če je  $|g'(\alpha)| > 1$ , potem je  $\alpha$  *odbojna* negibna točka.

**Definicija 2.7.** Recimo, da je  $\alpha = \lim_{r \rightarrow \infty} x_r$  in obstaja

$$\lim_{r \rightarrow \infty} \frac{|x_{r+1} - \alpha|}{|x_r - \alpha|^p} = c > 0.$$

Tedaj zaporedje  $x_r$  konvergira k  $\alpha$  z redom  $p$ .

- Če je  $p = 1$ , konvergenca *linearna*,
- Če je  $p = 2$ , konvergenca *kvadratična*,
- Če je  $p = 3$ , konvergenca *kubična*,
- Če je  $1 < p < 2$ , konvergenca *superlinearna*,
- Če je  $2 < p < 3$ , konvergenca *superkvadratična*.

**Izrek 2.8.** Naj bo  $\alpha = g(\alpha)$ . Naj bo  $g$   $p$ -krat zvezno odvedljiva funkcija v okolici  $\alpha$  in

$$g'(\alpha) = \dots = g^{(p-1)}(\alpha) = 0 \quad \text{in} \quad g^{(p)}(\alpha) \neq 0.$$

Tedaj je v bližini  $\alpha$  red konvergence  $x_{r+1} = g(x_r)$  enak  $p$ .

V primeru  $p = 1$ , mora veljati še, da je  $|g'(\alpha)| < 1$ .

Dokaz. **TODO**

□

**Praktičen način.**

- Pri linearni konvergenci se število točnih decimalk povečuje linearno (ni nujno za celo število).
- Pri kvadratni konvergenci se število točnih decimalk iz koraka v korak približno podvoji.

**2.4 Tangentna metoda**

Naj bo  $f$  dvakrat zvezna odvedljiva funkcija v okolici  $\alpha$  in  $f(\alpha) = 0$ . Naj bo  $x_r$  približek za ničlo  $\alpha$ . Iščemo popravek  $\Delta x_r$ , da bo  $f(x_r + \Delta x_r) = 0$ .

**Ideja.** Razvijemo  $f(x_r + \Delta x_r)$  v Taylorjevo vrsto:

$$0 = f(x_r + \Delta x_r) = f(x_r) + f'(x_r) \cdot \Delta x_r + \underbrace{\frac{f''(c_r)}{2} \cdot \Delta x_r^2}_{\text{zanemarimo}}$$

$$\Rightarrow \Delta x_r \approx -\frac{f(x_r)}{f'(x_r)}.$$

Za novi približek vzamemo

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)}.$$

To je *tangentna metoda*.

**Opomba 2.9.** Tangentna metoda je poseben primer navadne iteracije za

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

**Geometrijska interpretacija**

TODO

**Analiza reda konvergence**

TODO

**Konvergenca tangentne metode**

Definiramo z  $e_r := x_r - \alpha$  napako  $i$ -tega približka. Naj bo  $f \in C^2$  in  $\alpha$  enostavna ničla. Teda

$$0 = f(\alpha) = f(x_r) + f'(x_r)(\alpha - x_r) + \frac{f''(c_r)}{2}(\alpha - x_r)^2$$

$$\Rightarrow 0 = \frac{f(x_r)}{f'(x_r)} + \alpha - x_r + \frac{f''(c_r)}{2f'(x_r)}(\alpha - x_r)^2$$

$$\Rightarrow e_{r+1} = \frac{f''(c_r)}{2f'(x_r)}e_r^2.$$

V bližini  $\alpha$  torej velja:

$$e_{r+1} \approx C e_r^2, \quad C = \frac{f''(\alpha)}{2f'(\alpha)}.$$

Torej za dvakrat zvezno odvedljivo funkcijo  $f$  imamo zagotovljeno lokalno konvergenco tangentne metode.

V določenih primerih imamo tudi globalno konvergenco.

**Izrek 2.10.** *Naj bo  $f$  na  $I = [a, \infty)$  dvakrat zvezno odvedljiva funkcija, ki ima ničlo  $\alpha \in I$ . Naj bo  $f$  naraščajoča in konveksna. Tedaj je  $\alpha$  edina ničla na  $I$  in za vsak  $x_0 \in I$  tangentna metoda konvergira k  $\alpha$ .*

*Dokaz.* **TODO**

□

## 2.5 Metode brez računanja odvoda

### Sekantna metoda

Namesto tangente uporabimo sekanto skozi  $(x_r, f(x_r))$ ,  $(x_{r-1}, f(x_{r-1}))$ . To je ekvivalentno temu, da v tangentni metodi  $f'(x_r)$  aproksimiramo z  $\frac{f(x_r) - f(x_{r-1})}{x_r - x_{r-1}}$ . Torej

$$x_{r+1} = x_r - \frac{f(x_r)(x_r - x_{r-1})}{f(x_r) - f(x_{r-1})}.$$

V vsakem koraku potrebujemo en nov izračun funkcije  $f$ , razen na začetku 2. Na začetku še potrebujemo sve začetni vrednosti  $x_0$ ,  $x_1$ , ni pa kakšnih dodatnih omejitev, kot pri bisekciji.

**Opomba 2.11.** Pričakujemo, da se sekantna metoda obnaša podobno kot tangentna metoda.

Za sekantno metodo se da pokazati zvezo

$$e_{r+1} \approx C e_r e_{r-1}.$$

Sekantna metoda ima red konvergence  $p \approx 1.62$ , če je  $\alpha$  enostavna ničla in  $f''(\alpha) \neq 0$ .

**Opomba 2.12.** Sekantna metoda ni primer navadne iteracije, saj je  $x_{r+1}$  odvisen od dveh prejšnjih členov:  $x_r$  in  $x_{r-1}$ .

### Mullerjeva metoda

Skozi točke  $(x_r, f(x_r))$ ,  $(x_{r-1}, f(x_{r-1}))$ ,  $(x_{r-2}, f(x_{r-2}))$  potegnemo kvadratni polinom in za naslednji približek vzamemo tisto izmed njegovih dveh ničel, ki je bližja  $x_r$ .

Potrebujemo tri začetne približke  $x_0, x_1, x_2$ . V vsakem koraku moramo izračunati eno vrednost funkcije, razen na začetku tri.

Velja zveza

$$|e_{r+1}| \approx C \cdot |e_r| \cdot |e_{r-1}| \cdot |e_{r-2}|.$$

Mullerjeva metoda ima red konvergence  $p \approx 1.84$ .

### Inverzna interpolacija

Zamenjamo vlogi  $x$  in  $y$  in čez točki  $(f(x_r), x_r)$ ,  $(f(x_{r-1}), x_{r-1})$ ,  $(f(x_{r-2}), x_{r-2})$  napeljemo kvadratni polinom. Ta kvadratni polinom aproksimira inverzno funkcijo. Vzamemo

$$x_{r+1} = p(0).$$

Inverzna interpolacija ima red konvergence  $p \approx 1.84$ .

### Kombinirane metode

Kombiniramo več metod in s tem zagotovimo robustnost, kot pri bisekciji, s hitrejšo konvergenco, kot npr. pri inverzni interpolaciji.

Imamo interval  $[a, b]$ , kjer je  $f(a) \cdot f(b) < 0$ . Naslednji potencialni približek  $c$  izračunamo npr. s sekantno metodo ali inverzno interpolacijo. Če je  $c$  izven intervala  $[a, b]$  namesto tega izberemo  $c = \frac{a+b}{2}$ . Nadaljujemo podobno kot pri bisekciji.

**Primer 2.13.** Brantova metoda (fzero) kombinira bisekcijo, sekantno metodo in inverzno interpolacijo.

## 2.6 Ničle polinomov

Imamo polinom  $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ , ki ima ničle  $\alpha_1, \dots, \alpha_n$ . Možnosti za izračun ničel so:

1. Ničle računamo eno po eno (ali dve po dve, če imamo konjugirane pare). Če je  $\alpha_1$  enostavna ničla, je

$$p(x) = (x - \alpha_1)q(x), \quad \text{st } q(x) = n - 1.$$

Nadaljujemo z iskanjem ničel polinoma  $q(x)$ .

**Primer 2.14.** Laguerreova metoda, Bairstow-Hitchcode.

2. Problem prevedemo na računanje lastnih vrednosti matrike

$$C_p = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ -\frac{a_n}{a_0} & -\frac{a_{n-1}}{a_0} & \dots & -\frac{a_2}{a_0} & -\frac{a_1}{a_0} \end{bmatrix}$$

$C_p$  je spremljevalna matrika polinoma  $p(x)$ . Njen karakteristični polinom je s skalarjem pomnožen  $p(x)$ . Lastne vrednosti  $C_p$  so ničle  $p(x)$ .

**Primer 2.15.** Funkcija `roots` v MatLab.

3. Metode, ki vzporedno računajo vse ničle polinoma.

**Primer 2.16.** Ehrlich-Abertborva metoda, Durand-Kernerjeva metoda.

**Primer 2.17** (Durand-Kernerjeva metoda). Naj bo  $p$  polinom stopnje  $n$  z vodilnim koeficientom 1. Potem ima  $p$  ničle  $\alpha_1, \dots, \alpha_n$  in je  $p(x) = (x - \alpha_1) \cdots (x - \alpha_n)$ .

Naj bo  $x_1, \dots, x_n$  paroma različni približki za  $\alpha_1, \dots, \alpha_n$ . Iščemo popravke  $\Delta x_1, \dots, \Delta x_n$ , da bodo  $x_1 + \Delta x_1, \dots, x_n + \Delta x_n$  točne ničle  $p$ . Tedaj

$$\begin{aligned} p(x) &= (x - (x_1 + \Delta x_1)) \cdots (x - (x_n + \Delta x_n)) \\ &= \prod_{j=1}^n (x - x_j) - \sum_{j=1}^n \Delta x_j \prod_{k=1, k \neq j}^n (x - x_k) \\ &\quad + \underbrace{\sum_{j,k=1, j < k}^n \Delta x_j \Delta x_k \prod_{l=1, l \neq j, k}^n (x - x_l) + \dots}_{\text{zanemarimo}} \end{aligned}$$

Vstavimo  $x = x_i$ , dobimo

$$\begin{aligned} p(x_i) &= -\Delta x_i \prod_{k=1, k \neq i}^n (x_i - x_k) \\ \Rightarrow \Delta x_i &= -\frac{p(x_i)}{\prod_{k=1, k \neq i}^n (x_i - x_k)} \end{aligned}$$

Definiramo

$$\begin{aligned} x^{(0)} &= [x_1^{(0)}, \dots, x_n^{(0)}] \\ \Rightarrow x_i^{(r+1)} &= x_i^{(r)} - \frac{p(x_i^{(r)})}{\prod_{k=1, k \neq i}^n (x_i^{(r)} - x_k^{(r)})}, \quad r = 0, 1, \dots, \quad i = 1, \dots, n. \end{aligned}$$

### 3 Sistemi linearnih enačb

Naj bo  $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$ .

#### 3.1 Uvod

Rešujemo sistem linearnih enačb

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &= b_m, \end{aligned}$$

ki ga pišemo v obliki

$$Ax = b,$$

kjer

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}.$$

Označimo z  $a_i$   $i$ -ti stolpec matrike  $A$  in z  $\alpha_j^T$   $j$ -to vrstico matrike  $A$ . Naj bo

$$e_i = [0 \ \dots \ 0 \ 1_i \ 0 \ \dots \ 0]^T.$$

Tedaj velja

$$Ae_i = a_i, \quad e_j^T A = \alpha_j^T \quad \text{in} \quad e_i^T Ae_k = a_{ik}.$$

Naj bo  $x, y \in \mathbb{F}^n$ . Standardni skalarni produkt je

$$\langle x, y \rangle = \sum_{i=1}^n x_i \overline{y_i} = y^H x,$$

kjer je

$$A^H = \overline{A^T}$$

hermitsko transponiranje.

Če imamo sistem  $y = Ax$ ,  $A \in \mathbb{R}^{n \times n}$ ,  $x, y \in \mathbb{R}^n$ , si to lahko predstavljamo:

1. Po elementih, tj.

$$y_i = \sum_{k=1}^n a_{ik} x_k = \alpha_i^T X.$$

2. Kot linearno kombinacijo stolpcev, tj.

$$y = \sum_{i=1}^n x_i a_i \Rightarrow e_i^T y = \sum_{i=1}^n x_i e_i^T a_i.$$

Če imamo produkt  $C = A \cdot B$ , kjer so  $A, B, C \in \mathbb{R}^n$ , si to lahko predstavljamo:

1. Po elementih, tj.

$$c_{ik} = \sum_{l=1}^n a_{il}b_{lk} = \alpha_i^T b_k.$$

2. Po stolpcih, tj.

$$A \cdot [b_1 \ \dots \ b_n] = [A \cdot b_1 \ \dots \ A \cdot b_n] \Rightarrow c_i = A \cdot b_i.$$

3. Po vrsticah, tj.

$$\begin{bmatrix} \alpha_1^T \\ \vdots \\ \alpha_n^T \end{bmatrix} \cdot B = \begin{bmatrix} \alpha_1^T \cdot B \\ \vdots \\ \alpha_n^T \cdot B \end{bmatrix} \Rightarrow \gamma_i^T = \alpha_i^T \cdot B.$$

4. Kot vsoto  $n$  matrik ranga 1 (diade), tj.

$$C = \sum_{l=1}^n a_l \beta_l^T.$$

Vemo, da če  $x, y \in \mathbb{R}^n$ ,  $x, y \neq 0$ , potem  $xy^T$  matrika ranga 1.

Spomnimo se nekaj osnovnih definicij in trditev. Naj bo  $A \in \mathbb{F}^{n \times n}$ .

**Trditev 3.1.** Naj bo  $A \in \mathbb{F}^{n \times n}$ . NTSE:

1.  $A \in \mathbb{F}^{n \times n}$  je nesingularna.
2.  $\exists A^{-1} \cdot A \cdot A^{-1} = A^{-1} \cdot A = I$ .
3.  $\det A \neq 0$ .
4.  $\text{rang } A = n$ .
5.  $\ker A = \{0\}$ .
6. Vse lastne vrednosti so neničelne.

**Definicija 3.2.** Naj bo  $A \in \mathbb{F}^{n \times n}$ .

- $\lambda \in \mathbb{F}$  je lastna vrednost matrike  $A$ , če

$$\exists x \in \mathbb{F}^n \setminus \{0\} \cdot Ax = \lambda x.$$

- Matrika  $A$  je simetrična, če

$$A^T = A,$$

hermitska, če

$$A^H = A.$$

- Matrika  $A$  je simetrična pozitivno definitna, če

$$A = A^T \quad \text{in} \quad \forall x \in \mathbb{F}^n \setminus \{0\} \cdot x^T A x > 0,$$

simetrična pozitivno semidefinitna, če

$$A = A^T \quad \text{in} \quad \forall x \in \mathbb{F}^n \setminus \{0\} \cdot x^T A x \geq 0,$$

hermitska pozitivno semidefinitna, če

$$A = A^H \quad \text{in} \quad \forall x \in \mathbb{F}^n \setminus \{0\} \cdot x^H A x > 0,$$

hermitska pozitivno semidefinitna, če

$$A = A^H \quad \text{in} \quad \forall x \in \mathbb{F}^n \setminus \{0\} \cdot x^H A x \geq 0,$$

### 3.2 Vektorske in matrične norme

**Definicija 3.3.** *Vektorska norma* je preslikava  $\|\cdot\| : \mathbb{C}^n \rightarrow \mathbb{R}$ , za katero velja:

1. Pozitivna definitnost, tj.  $\forall x \in \mathbb{C}^n. \|x\| \geq 0 \wedge \|x\| = 0 \Leftrightarrow x = 0$ .
2. Homogenost, tj.  $\forall \lambda \in \mathbb{C}. \forall x \in \mathbb{C}^n. \|\lambda x\| = |\lambda| \cdot \|x\|$ .
3. Trikotniška neenakost, tj.  $\forall x, y \in \mathbb{C}^n. \|x + y\| \leq \|x\| + \|y\|$ .

Najbolj znane norme so

$$\begin{aligned}\|x\|_1 &= \sum_{i=1}^n |x_i| \\ \|x\|_2 &= \left( \sum_{i=1}^n |x_i|^2 \right)^{1/2} \\ \|x\|_\infty &= \max\{|x_i| \mid i \in \{1, \dots, n\}\}\end{aligned}$$

Vse vektorske norme so ekvivalentne, tj.

$$\forall \|\cdot\|_a, \|\cdot\|_b \in \mathbb{R}^{\mathbb{C}^n}. \forall x \in \mathbb{C}^n. \exists c_1, c_2 > 0. c_1 \|x\|_a \leq \|x\|_b \leq c_2 \|x\|_a.$$

Za  $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$  velja:

$$\begin{aligned}\|x\|_2 &\leq \|x\|_1 \leq \sqrt{n} \cdot \|x\|_2 \\ \|x\|_\infty &\leq \|x\|_2 \leq \sqrt{n} \cdot \|x\|_\infty \\ \|x\|_\infty &\leq \|x\|_1 \leq n \cdot \|x\|_\infty\end{aligned}$$

**Definicija 3.4.** *Matrična norma* je preslikava  $\|\cdot\| : \mathbb{C}^{n \times n} \rightarrow \mathbb{R}$ , za katero velja:

1. Pozitivna definitnost.
2. Homogenost.
3. Trikotniška neenakost.
4. Submultiplikativnost, tj.  $\forall A, B \in \mathbb{C}^{n \times n}. \|A \cdot B\| \leq \|A\| \cdot \|B\|$ .

**Definicija 3.5.** *Vektorizacija* matrike  $A \in \mathbb{F}^{n \times n}$  je preslikava

$$\begin{aligned}\text{vec}: \mathbb{C}^{n \times n} &\rightarrow \mathbb{C}^{n^2} \\ A &\mapsto [a_{11} \ \dots \ a_{1n} \ a_{21} \ \dots \ a_{nn}]^T\end{aligned}$$

Sedaj lahko definiramo vektorske norme

$$\begin{aligned}N_1(A) &= \|\text{vec}(A)\|_1 = \sum_{i,j=1}^n |a_{ij}| \\ N_2(A) &= \|\text{vec}(A)\|_2 = \left( \sum_{i,j=1}^n |a_{ij}|^2 \right)^{1/2} \\ N_\infty(A) &= \|\text{vec}(A)\|_\infty = \max\{|x_i| \mid i \in \{1, \dots, n\}\}\end{aligned}$$



**Trditev 3.6.** Normi  $N_1$  in  $N_2$  sta matrični normi. Norma  $N_\infty$  ni matrična norma.

Dokaz. **TODO** □

**Definicija 3.7.** Matrična norma  $N_2$  je *Frobeniusova norma*:

$$\|A\|_F = \left( \sum_{i,j=1}^n |a_{ij}|^2 \right)^{1/2}.$$

**Lema 3.8.** Naj bo  $\|\cdot\|_v$  vektorska norma na  $\mathbb{C}^n$ . Tedaj je

$$\|A\|_v := \max_{x \neq 0} \frac{\|Ax\|_v}{\|x\|_v}$$

matrična norma. Rečemo ji operatorska norma.

Dokaz. **TODO** □

Sedaj lahko definiramo *operatorsko  $p$ -normo*:

$$\|A\|_p := \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} \text{ za } p \in \{1, 2, \dots, \infty\}.$$

**Lema 3.9.** Naj bo  $A \in \mathbb{C}^{n \times n}$ . Tedaj

$$\|A\|_1 = \max \left\{ \sum_{i=1}^n |a_{ij}| \mid j \in \{1, \dots, n\} \right\} = \max \{ \|a_j\|_1 \mid j \in \{1, \dots, n\} \}.$$

Dokaz. **TODO** □

**Lema 3.10.** Naj bo  $A \in \mathbb{C}^{n \times n}$ . Tedaj

$$\|A\|_\infty = \max \left\{ \sum_{j=1}^n |a_{ij}| \mid i \in \{1, \dots, n\} \right\} = \max \{ \|\alpha_i^T\|_1 \mid i \in \{1, \dots, n\} \}.$$

Naj bo  $A \in \mathbb{C}^{n \times n}$ . Matrika  $B = A^H A$  je hermitska in pozitivno semidefinitna, saj

$$1. \ B^H = (A^H A)^H = A^H (A^H)^H = A^H A = B \text{ in}$$

$$2. \ x^H B x = x^H A^H A x = (Ax)^H Ax = \|Ax\|_2^2 \geq 0.$$

Vemo, da so torej vse njene lastne vrednosti nenegativne, zato jih lahko pišemo in uredimo kot

$$\sigma_1^2 \geq \sigma_2^2 \geq \dots \geq \sigma_n^2 \geq 0.$$

Vrednostim  $\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2$  pravimo *singularne vrednosti* matrike  $A$ .

**Lema 3.11.** Naj bo  $A \in \mathbb{C}^{n \times n}$ . Tedaj

$$\|A\|_2 = \sigma_1(A) = \sqrt{\lambda_{\max}(A^H A)}$$

Dokaz. **TODO** □

**Definicija 3.12.** Norma  $\|\cdot\|$  je *spektralna norma*.

**Opomba 3.13.** Vsako matriko  $A \in \mathbb{C}^{n \times n}$  lahko zapišemo v obliki

$$A = U \Sigma V^T,$$

kjer sta  $U, V$  ortogonalni matriki in

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix}.$$

To je *singularni razcep* matrike  $A$ .

**Opomba 3.14.** Frobeniusova norma ni operatorska norma, saj je  $\|I\|_F = \sqrt{n}$ . Za operatorske norme pa velja, da je  $\|I\| = 1$ .

Tudi vse matrične norme so ekvivalentne. Velja:

$$\begin{aligned} \frac{1}{\sqrt{n}} \cdot \|A\|_F &\leq \|A\|_2 \leq \|A\|_F \\ \frac{1}{\sqrt{n}} \cdot \|A\|_1 &\leq \|A\|_2 \leq \sqrt{n} \cdot \|A\|_1 \\ \frac{1}{\sqrt{n}} \cdot \|A\|_\infty &\leq \|A\|_2 \leq \sqrt{n} \cdot \|A\|_\infty \end{aligned}$$

Poleg tega velja še:

$$\begin{aligned} N_\infty(A) &\leq \|A\|_2 \leq n \cdot N_\infty(A) \\ \|A\|_2 &\leq \sqrt{\|A\|_1 \cdot \|A\|_\infty} \\ \forall i \in \{1, \dots, n\} \cdot \|a_i\|_2, \|\alpha_i^T\|_2 &\leq \|A\|_2. \end{aligned}$$

Naj bo  $\|\cdot\|_m$  operatorska norma z vektorsko normo  $\|\cdot\|_v$ . Teda

$$\forall A \in \mathbb{C}^{n \times n} \cdot \forall x \in \mathbb{C}^n \cdot \|Ax\|_v \leq \|A\|_m \cdot \|x\|_v.$$

**Definicija 3.15.** Če za matrično normo  $\|\cdot\|_m$  in vektorsko normo  $\|\cdot\|_v$  za vsak par  $A \in \mathbb{C}^{n \times n}$ ,  $x \in \mathbb{C}^n$  velja:

$$\|Ax\|_v \leq \|A\|_m \cdot \|x\|_v,$$

pravimo, da sta normi *usklajeni*.

**Lema 3.16.** Za vsako matrično normo  $\|\cdot\|_m$  obstaja usklajena vektorska norma  $\|\cdot\|_v$ .

Dokaz. **TODO** □

**Lema 3.17.** Naj bo  $\lambda \in \mathbb{C}$  lastna vrednost matrike  $A \in \mathbb{C}^{n \times n}$ . Potem za poljubno matrično normo  $\|\cdot\|_m$  velja:

$$|\lambda| \leq \|A\|_m.$$

Dokaz. **TODO** □

Vse norme  $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty, \|\cdot\|_F$  se enostavno razširijo na pravokotne matrike  $A \in \mathbb{C}^{m \times n}$ . Če sta  $A, B$  ustreznih velikosti, da obstaja produkt  $A \cdot B$ , potem velja tudi submultiplikativnost. Ne veljajo pa vse prejšnje ocene za norme.

Velja pa

$$\begin{aligned}\|A\|_2 &= \|A^H\|_2 \\ \|A\|_F &= \|A^H\|_F \\ \|A\|_1 &= \|A^H\|_\infty\end{aligned}$$

Matrika  $U \in \mathbb{C}^{n \times n}$  je *unitarna*, če  $U \cdot U^H = U^H U = I$ .

**Lema 3.18.** Normi  $\|\cdot\|_2$  in  $\|\cdot\|_F$  sta invariantni na množenje z unitarno matriko.

Dokaz. **TODO**

□

**Lema 3.19.** Naj za  $X$  velja  $\|X\| < 1$ . Tedaj

1. Matrika  $I - X$  je obrnljiva.
2.  $(I - X)^{-1} = \sum_{k=0}^{\infty} X^k$ .
3. Če je  $\|I\| = 1$ , potem  $\|(I - X)^{-1}\| \leq \frac{1}{1 - \|X\|}$ .

Dokaz. **TODO**

□

### 3.3 Občutljivost sistemov linearnih enačb

**Definicija 3.20.** Naj bo  $A \in \mathbb{C}^{n \times n}$ . Občutljivost ali pogojnostno število matrike  $A$  je

$$\kappa(A) := \|A\| \cdot \|A^{-1}\|.$$

**Opomba 3.21.** Velja ocena:

$$\forall A \in \mathbb{C}^{n \times n}. \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = \kappa(A).$$

**Izrek 3.22.** Naj bo  $A \in \mathbb{C}^{n \times n}$  nesingularna matrika in  $Ax = b$ . Če  $A$  zmotimo v  $A + \Delta A$  in  $b$  v  $b + \Delta b$ , kjer velja

$$\|\Delta A\| \leq \frac{1}{\|A^{-1}\|},$$

potem za rešitev zmotenega sistema  $(A + \Delta A)(x + \Delta x) = b + \Delta b$  velja

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A) \cdot \frac{\|\Delta A\|}{\|A\|}} \left( \frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right),$$

kjer za  $\|\cdot\|$  velja  $\|I\| = 1$ .

**Opomba 3.23.** Velja:

$$\kappa_2(A) = \frac{\sigma_1(A)}{\sigma_n(A)}.$$

**Opomba 3.24.** Edine matrike z občutljivostjo 1 v  $\|\cdot\|_2$  so s skalarjem pomnožene unitarne matrike.

**Definicija 3.25.** Hilbertova matrika  $H \in \mathbb{R}^{n \times n}$  je matrika z elementi

$$h_{ij} = \frac{1}{i+j-1}.$$

**Opomba 3.26.** Hilbertove matrike so zelo občutljive.

TODO

### 3.4 LU razcep

Želimo rešiti sistem linearnih enačb

$$Ax = b, \quad A \in \mathbb{C}^{n \times n}, \quad x, b \in \mathbb{C}^n, \quad \det A \neq 0.$$

Kakšni metodi so na voljo?

Naj bo

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, \quad a_{11} \neq 0.$$

Definiramo

$$L_1 = \begin{bmatrix} 1 & & & \\ -l_{21} & 1 & & \\ -l_{31} & & \ddots & \\ \vdots & & & \ddots \\ -l_{n1} & & & & 1 \end{bmatrix}, \quad l_{j1} = \frac{a_{j1}}{a_{11}}, \quad j \in \{2, \dots, n\}.$$

Potem

$$L_1 A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ 0 & a_{32}^{(1)} & \dots & a_{3n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{bmatrix}, \quad a_{22}^{(1)} \neq 0.$$

Nadaljujemo in dobimo

$$L_i = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & -l_{i+1,i} & 1 \\ & & \vdots & \\ & & -l_{n,i} & & 1 \end{bmatrix}, \quad l_{ji} = \frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}}, \quad j \in \{i+1, \dots, n\}$$

ter

$$A^{(i-1)} = \begin{bmatrix} 0 & & & & \\ & \ddots & & & \\ & & 0 & & \\ & & & a_{ii}^{(i-1)} & \dots \\ & & & \vdots & \\ & & & a_{ni}^{(i-1)} & \dots \end{bmatrix}$$

V končnem

$$L_{n-1} \cdots L_2 L_1 A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ & a_{22}^{(1)} & a_{23}^{(1)} & \dots & a_{2n}^{(1)} \\ & & a_{33}^{(2)} & \dots & a_{3n}^{(1)} \\ & & & \ddots & \\ & & & & a_{nn}^{(n-1)} \end{bmatrix}$$

Definiramo  $L_{n-1} \cdots L_2 L_1 A =: U$ . Tedaj

$$A = \underbrace{L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1}}_L U = LU.$$

Kako dobimo matriko  $L$ ?

$L_j = I - l_j e_j^T$  je eliminacijska matrika

$\Rightarrow L_j^{-1} = I + l_j e_j^T$ , kjer

$$l_j = [0 \ \dots \ 0 \ l_{j+1,j} \ \dots \ l_{n,j}]^T$$

$$\Rightarrow L = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & & \ddots & \\ l_{n1} & l_{n2} & \dots & l_{n,n-1} & 1 \end{bmatrix}$$

S tem dobimo  $LU$  razcep brez pivotiranja, kjer je  $L$  spodnje trikotna z 1 na diagonalni in  $U$  zgornje trikotna matrika.

---

**Algorithm 2:** LU razcep

---

**Data:**  $A \in \mathbb{C}^{n \times n}$

**Result:**  $L \in \mathbb{C}^{n \times n}$ ,  $U \in \mathbb{C}^{n \times n}$

**for**  $i = 1, \dots, n-1$  **do**

**for**  $j = i+1, \dots, n$  **do**

$l_{ji} \leftarrow a_{ji}/a_{ii}$

**for**  $k = i+1, \dots, n$  **do**

$a_{jk} \leftarrow a_{jk} - l_{ji}a_{ik}$

**end**

**end**

**end**

$U \leftarrow A$

---

Kako zdaj rešimo sistem  $Ax = b$ ?

$$Ax = b \Leftrightarrow L \underbrace{Ux}_y = b.$$

**Postopek.**

1.  $A = LU$ ,
2. Reši  $Ly = b$ ,
3. Reši  $Ux = y$ .

Sistem  $Ly = b$  rešimo s *premo substitucijo* (od zgoraj navzdol), tj.

$$l_{j1}y_1 + \dots + l_{j,j-1}y_{j-1} + y_j = b_j.$$

---

**Algorithm 3:** Prema substitucija

---

**Data:**  $L \in \mathbb{C}^{n \times n}$ ,  $b \in \mathbb{C}^n$

**Result:**  $y \in \mathbb{C}^n$

**for**  $j = 1, \dots, n$  **do**

$y_j \leftarrow b_j - \sum_{k=1}^{j-1} l_{jk}y_k$

**end**

---

Sistem  $Ux = y$  rešimo s *obratno substitucijo* (od zgoraj navzdol), tj.

$$u_{jj}x_j + \dots + u_{j,n}x_n = y_j.$$

---

**Algorithm 4:** Obratna substitucija

---

**Data:**  $U \in \mathbb{C}^{n \times n}$ ,  $y \in \mathbb{C}^n$

**Result:**  $x \in \mathbb{C}^n$

**for**  $j = n, n-1, \dots, 1$  **do**

$x_j \leftarrow (1/u_{jj}) \cdot (y_j - \sum_{k=j+1}^n u_{jk}x_k)$

**end**

---

Elementom  $a_{11}, a_{22}^{(1)}, \dots, a_{n-1,n-1}^{(n-2)}$ , s katerimi delimo, pravimo *pivotni elementi*.

**Izrek 3.27.** Za  $A \in \mathbb{C}^{n \times n}$  NTSE:

1. Obstaja enoličen LU razcep  $A = LU$ , kjer je  $L$  spodnje trikotna z 1 na diagonalni in  $U$  nesingularna zgornje trikotna.
2.  $\det(A_k) \neq 0$  za vse  $A_k = A(1:k, 1:k)$ ,  $k \in \{1, \dots, n\}$ .
3. Vse vodilne podmatrike  $A$  so nesingularne.

**Zgled 3.28.** Pri numeričnem računanju so težave lahko tudi zaradi pivotov, ki so sicer neničelni, a blizu 0. **TODO**

Rešitev za težave s (skoraj) ničelnimi pivoti je pivotiranje, kjer dopuščamo:

1. Menjave vrstic - *delno pivotiranje*.
2. Menjave vrstic in stolpcev - *kompletno pivotiranje*.

### Delno pivotiranje

V koraku  $i$  poiščemo največjega izmed elementov  $|a_{ij}|, |a_{i+1,j}|, \dots, |a_{ni}|$  in zamenjamo ustrezni vrstici.

---

**Algorithm 5:** LU razcep z delnim pivotiranjem
 

---

**Data:**  $A \in \mathbb{C}^{n \times n}$

**Result:**  $L \in \mathbb{C}^{n \times n}$ ,  $U \in \mathbb{C}^{n \times n}$

```

for  $i = 1, \dots, n$  do
    find  $p \in \{i, \dots, n\}$ , da  $|a_{pi}| = \max_{l \in \{i, \dots, n\}} |a_{li}|$ 
    swap the  $i$ -th and  $p$ -th rows
    for  $j = i + 1, \dots, n$  do
         $l_{ji} \leftarrow a_{ji}/a_{ii}$ 
        for  $k = i + 1, \dots, n$  do
             $a_{jk} \leftarrow a_{jk} - l_{ji}a_{ik}$ 
        end
    end
end
end
  
```

---

Dobimo razcep  $PA = LU$ , kjer je  $P$  permutacijska matrika, ki ustreza zamenjavi vrstic. Kako zdaj rešimo sistem?

$$Ax = b \Rightarrow PAx = Pb \Rightarrow LUx = Pb$$

#### Postopek.

1.  $PA = LU$ ,
2. Reši  $Ly = Pb$ ,
3. Reši  $Ux = y$ .

**Izrek 3.29.** Če je  $A$  nesingularna, potem obstaja taka permutacijska matrika  $P$ , da za  $PA$  obstaja LU razcep brez pivotiranja.

Dokaz. **TODO**

□

### Kompletno pivotiranje

Poiščemo največji element bloka in zamenjamo ustrezni vrstici in stolpca. Dobimo sistem

$$\underbrace{PAQ}_{LU} \underbrace{Q^T}_{\tilde{x}} x = Pb$$

#### Postopek.

1.  $PAQ = LU$ ,
2. Reši  $Ly = Pb$ ,
3. Reši  $U\tilde{x} = y$ ,
4.  $x = Q\tilde{x}$ .

### 3.5 Analiza zaokrožitvenih napak pri LU razcepu

**Definicija 3.30.** *Pivotna rast* je

$$g(A) := \frac{\max_{i,j \in \{1, \dots, n\}} |u_{ij}|}{\max_{i,j \in \{1, \dots, n\}} |a_{ij}|}$$

TODO

### 3.6 Sistemi posebne oblike

Recimo, da rešujemo sistem  $Ax = b$ , kjer ima  $A$  posebno obliko, npr. diagonalna, trikotna itn.

#### Simetrična pozitivno definitna matrika

**Izrek 3.31.** *Naj bo  $A \in \mathbb{R}^{n \times n}$ . Tedaj*

1. *Če je  $A$  s.p.d., so vse vodilne podmatrike  $A_k = A(1 : k, 1 : k)$  tudi s.p.d.*
2. *Če je  $A$  s.p.d., obstaja enoličen LU razcep  $A = LU$ , kjer je  $u_{ii} > 0$  za  $i \in [n]$ .*
3.  *$A$  je s.p.d. natanko tedaj, ko obstaja enolična spodnja trikotna matrika  $V$ ,  $v_{ii} > 0$  za  $i \in [n]$ , da je*

$$A = V \cdot V^T.$$

*To je razcep Choleskega. Matriko  $V$  pa imenujemo faktor Choleskega.*

*Dokaz.* TODO

□

Kako rešimo sistem s pomočjo razcepa Choleskega?

**Postopek.**

1.  $A = VV^T$ ,
2.  $Vy = b$ ,
3.  $V^T x = y$ .

---

#### Algorithm 6: Razcep Choleskega

---

**Data:**  $A \in \mathbb{C}^{n \times n}$

**Result:**  $V \in \mathbb{C}^{n \times n}$

**for**  $i = 1, \dots, n$  **do**

$v_{ii} \leftarrow (a_{ii} - \sum_{k=1}^{i-1} v_{ik}^2)^{1/2}$   
**for**  $j = i + 1, \dots, n$  **do**  
 $v_{ji} \leftarrow (1/v_{ii}) \cdot (a_{ij} - \sum_{k=1}^{i-1} v_{ik} v_{jk})$   
**end**

**end**

---



## 4 Sistemi nelinearnih enačb

Naj bo  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  oz.  $F : \mathbb{C}^n \rightarrow \mathbb{C}^n$  preslikava. Iščemo vektorji  $x \in \mathbb{R}^n$ , ki rešijo sistem

$$F(x) = 0.$$

Podobno kot pri  $n = 1$  lahko uporabimo navadno iteracijo:

1. Prepišemo enačbo  $F(x) = 0$  v obliko  $x = G(x)$ .
2. Izberimo  $x^{(0)} \in \mathbb{R}^n$ .
3. Računamo rekurzivno:  $x^{(r+1)} = G(x^{(r)})$ .

Za konvergenco potrebujemo, da je  $G$  skrčitev na nekem zaprtem območju  $\Omega \subseteq \mathbb{R}^n$ , tj.

- $\forall x \in \Omega. G(x) \in \Omega$  in
- $\exists m \in [0, 1). \forall x, y \in \Omega. \|G(x) - G(y)\| \leq m \cdot \|x - y\|$ .

**Izrek 4.1.** Naj bo  $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$  zvezno odvedljiva na zaprtem območju  $\Omega \subseteq \mathbb{R}^n$ . Recimo, da velja

- $\forall x \in \Omega. G(x) \in \Omega$  in
- $\exists m \in [0, 1). \forall x \in \Omega. \rho(J_G(x)) \leq m$ ,

kjer

$$J_G = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix} \text{ je Jacobijeva matrika preslikave } G,$$

$$\rho(A) = \max\{|\lambda| \mid \lambda \text{ je lastna vrednost } A\} \text{ je spektralni radij.}$$

Potem za vsak  $x^{(0)} \in \Omega$  zaporedje  $x^{(r+1)} = G(x^{(r)})$  konvergira k negibni točki funkcije  $G$ , ki je edina negibna točka  $G$  na  $\Omega$ .

**Opomba 4.2.** TODO Za vsak  $\varepsilon > 0$  obstaja matrična norma (za dano matriko  $A$ ), da je

$$\rho(A) \leq A \leq \rho(A) + \varepsilon.$$

Če je  $\alpha = G(\alpha)$  in  $\rho(J_G(\alpha)) < 1$ , je  $\alpha$  privlačna negibna točka. Posledično za  $x^{(0)}$  dovolj blizu  $\alpha$  bo  $\lim_{r \rightarrow \infty} x^{(r)} = \alpha$  za  $x^{(r+1)} = G(x^{(r)})$ .

### 4.1 Newtonova metoda

**Ideja.** Naj bo  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  dvakrat zvezno odvedljiva v okolici  $\alpha$ , kjer  $F(\alpha) = 0$ . Naj bo  $x^{(r)}$  približek za  $\alpha$ . Iščemo popravek  $\Delta x^{(r)}$ , da bo

$$F(x^{(r)} + \Delta x^{(r)}) = 0.$$

Računamo

$$\begin{aligned} 0 &= F(x^{(r)} + \Delta x^{(r)}) = F(x^{(r)}) + J_F(x^{(r)}) \cdot \Delta x^{(r)} + \underbrace{o(\|\Delta x^{(r)}\|^2)}_{\text{zanemarimo}} \\ &\Rightarrow J_F(x^{(r)}) \Delta x^{(r)} \approx -F(x^{(r)}) \\ &\Rightarrow x^{(r+1)} = x^{(r)} - J_F^{-1}(x^{(r)}) \cdot F(x^{(r)}) \end{aligned}$$

Torej iteracijska funkcija je  $G(x) = x - J_F^{-1}(x^{(r)}) \cdot F(x^{(r)})$ .

**Opomba 4.3.** Če je  $\alpha$  enostavna ničla, je  $\det(J_F(\alpha)) \neq 0$ .

---

**Algorithm 7:** Newtonova metoda

---

**Data:**  $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $x_0 \in \Omega$ ,

**Result:**  $c \in \Omega$ ,  $F(c) = 0$

**for**  $r = 1$  **to**  $\infty$  **do**

solve  $J_F(x^{(r)}) \cdot \Delta x^{(r)} = -F(x^{(r)})$   
 $x^{(r+1)} \leftarrow x^{(r)} + \Delta x^{(r)}$

**end**

---

**Opomba 4.4.**

- Ne množimo z inverzom, ampak rešimo sistem linearnih enačb.
- Metoda ima kvadratično konvergenco v bližini enostavnih ničel.

Težava z Newtonovo metodo je, da lahko zelo zahtevna, saj

- V vsakem korak potrebujemo  $n \times n$  matriko  $J_F(x^{(r)})$  in
- vsakič nov LU razcep za  $J_F(x^{(r)})$ .

## 4.2 Kvazi Newtonove metode

**Ideja.** Namesto matrike  $J_F(x^{(r)})$  uporabimo njen približek.

---

**Algorithm 8:** Kvazi Newtonova metoda

---

**Data:**  $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $B_0 \in \mathbb{R}^{n \times n}$ ,  $x_0 \in \Omega$

**Result:**  $c \in \Omega$ ,  $F(c) = 0$

**for**  $r = 1$  **to**  $\infty$  **do**

solve  $B_r \cdot \Delta x^{(r)} = -F(x^{(r)})$   
 $x^{(r+1)} \leftarrow x^{(r)} + \Delta x^{(r)}$   
 update  $B_r$  to  $B_{r+1}$

**end**

---

### Broydenova metoda

Najbolj znana je *Broydenova metoda*. Pri tej metodi določimo  $B_{r+1}$  tako, da zadošča, t.i. *sekantnemu pogoju*:

$$B_{r+1} (x^{(r+1)} - x^{(r)}) = F(x^{(r+1)}) - F(x^{(r)}).$$

Pri čemer je  $B_{r+1} = B_r + \Delta B_r$  in ima med vsemi možnimi  $\Delta B_r$  minimalno normo  $\|\Delta B_r\|_2$ .

**Opomba 4.5.** Še bolj pomembno je, da ima  $\Delta B_r$  minimalen rang 1.

Dobimo, da

$$\begin{aligned} (B_r + \Delta B_r) \Delta x^{(r)} &= F(x^{(r+1)}) - F(x^{(r)}), \quad B_r \Delta x^{(r)} = -F(x^{(r)}) \\ \Rightarrow \Delta B_r \Delta x^{(r)} &= F(x^{(r+1)}). \end{aligned}$$

**Lema 4.6.** Za dana neničelna vektorja  $u, v \in \mathbb{R}^n$  je matrika  $A$  z minimalno normo  $\|\cdot\|_2$ , za katero velja  $Au = v$ , enaka

$$A = \frac{vu^T}{\|u\|_2^2}.$$

Dokaz. **TODO**

□

---

**Algorithm 9:** Broydenova metoda

---

**Data:**  $F : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $x_0 \in \Omega$

**Result:**  $c \in \Omega$ ,  $F(c) = 0$

$B_0 \leftarrow J_F(x^{(0)})$

**for**  $r = 1$  **to**  $\infty$  **do**

    solve  $B_r \cdot \Delta x^{(r)} = -F(x^{(r)})$

$x^{(r+1)} \leftarrow x^{(r)} + \Delta x^{(r)}$

$B_{r+1} = B_r + \frac{F(x^{(r+1)}) \cdot (\Delta x^{(r)})^T}{\|\Delta x^{(r)}\|_2^2}$

**end**

---

**Trditev 4.7** (Sherman-Morrisonova formula). **TODO**

$$(A + uv^T)^{-1} = A^{-1} - \frac{(A^{-1}u)(v^T A^{-1})}{1 + v^T A^{-1}u}$$

**Opomba 4.8.** Sistem  $B_r \cdot \Delta x^{(r)} = -F(x^{(r)})$  lahko rešimo v  $O(n^2)$  z uporabo Sherman-Morrisonove formule. Sicer ta ni numerično stabilna.

## 5 Linearni problemi najmanjših kvadratov

### 5.1 Uvod

Imamo matriko  $A \in \mathbb{R}^{m \times n}$ ,  $m > n$ ,  $b \in \mathbb{R}^m$ . Iščemo  $x \in \mathbb{R}^n$ , ki minimizira  $\|Ax - b\|_2$ .

Če je  $m > n$ , je sistem  $Ax = b$  *predoločen* (več enačb kot neznank). Razen izjemoma (če je  $b \in \text{im}A$ ) tak sistem nima rešitve.

**Izrek 5.1.** Če je  $A \in \mathbb{R}^{m \times n}$ ,  $m \geq n$ ,  $\text{rang}A = n$ , potem je  $x \in \mathbb{R}^n$ , ki za dani  $b \in \mathbb{R}^m$  minimizira  $\|Ax - b\|_2$ , rešitev normalnega sistema  $A^T Ax = A^T b$ .

Matrika  $A^T A$  je *Grammova matrika*.

Predoločen sistem lahko rešimo:

1.  $B = A^T A$ ,  $c = A^T b$ ,
2.  $B = VV^T$ ,
3.  $Vy = c$
4.  $V^T x = y$

Matrika

$$\begin{bmatrix} 1 & x_1 & \dots & x_1^{n-1} \\ \vdots & & & \\ 1 & x_n & \dots & x_n^{n-1} \end{bmatrix}$$

je *Vandermondova matrika*.

### 5.2 QR razcep

**Izrek 5.2.** Naj bo  $A \in \mathbb{R}^{m \times n}$ ,  $m \geq n$ ,  $\text{rang}A = n$ . Potem obstaja enoličen razcep  $A = QR$ , kjer je  $Q \in \mathbb{R}^{m \times n}$  matrika z ortonormiranimi stolpci ( $Q^T Q = I_n$ ) in  $R \in \mathbb{R}^{n \times n}$  zgornja trikotna matrika s pozitivnimi diagonalnimi elementi ( $r_{ii} > 0$ ) za  $i \in [n]$ .

---

#### Algorithm 10: QR razcep

---

**Data:**  $A \in \mathbb{R}^{m \times n}$

**Result:**  $Q \in \mathbb{R}^{m \times n}$ ,  $R \in \mathbb{R}^{n \times n}$

```

for  $k = 1 : n$  do
     $q_k \leftarrow a_k$ 
    for  $i = 1 : k - 1$  do
         $r_{ik} \leftarrow q_i^T a_k$ 
         $q_k \leftarrow q_k - r_{ik} q_i$ 
    end
     $r_{kk} \leftarrow \|q_k\|_2$ 
     $q_k \leftarrow q_k / r_{kk}$ 
end

```

---

To je *klasična Gram-Schmidtova ortogonalizacija*.

Obstaja tudi *modificirana Gram-Schmidtova ortogonalizacija*:

---

**Algorithm 11:** QR razcep (MGS)
 

---

**Data:**  $A \in \mathbb{R}^{m \times n}$

**Result:**  $Q \in \mathbb{R}^{m \times n}$ ,  $R \in \mathbb{R}^{n \times n}$

**for**  $k = 1 : n$  **do**

$q_k \leftarrow a_k$

**for**  $i = 1 : k - 1$  **do**

$r_{ik} \leftarrow q_i^T q_k$

$q_k \leftarrow q_k - r_{ik} q_i$

**end**

$r_{kk} \leftarrow \|q_k\|_2$

$q_k \leftarrow q_k / r_{kk}$

**end**

---

Preprost način reševanja:

1.  $[Q, R] = \text{mgs}(A)$ ;

2. Reši  $Rx = Q^T b$ .

Boljši način reševanja:

1. Izračunamo QR razcep matrike  $\begin{bmatrix} A & b \end{bmatrix}$ :

$$\begin{bmatrix} A & b \end{bmatrix} = \begin{bmatrix} Q & q_{m+1} \end{bmatrix} \cdot \begin{bmatrix} R & z \\ 0 & \rho \end{bmatrix};$$

2. Reši  $Rx = z$ .

### 5.3 Givensove rotacije

Naj bo  $A \in \mathbb{R}^{m \times n}$ ,  $m \geq n$ ,  $\text{rang} A = n$ . Vemo, da obstaja QR razcep  $A = QR$ .

Poznamo tudi razširjeni QR razcep:  $A = \tilde{Q}\tilde{R}$ , kjer je  $\tilde{Q} \in \mathbb{R}^{m \times m}$  ortogonalna,  $\tilde{R} \in \mathbb{R}^{m \times n}$  zgornja trapezna (vsi elementi pod glavno diagonalo so enaki 0).

Prvih  $n$  stolpcev matrike  $\tilde{Q}$  in zgornji kvadrat matrike  $\tilde{R}$  tvori QR razcep matrike  $A$ . Minimum bo dosežen, ko  $Rx = Q^T b$ .

Navadno rotacijo v ravnini posplošimo na rotacijo v ravnini  $(i, k)$  v  $\mathbb{R}^n$ . Označimo z  $c = \cos \varphi$  in  $s = \sin \varphi$ . Dobimo ortogonalno matriko, ki je enaka identiteti povsod razen v  $i$ -ti in  $k$ -ti vrstici, kjer je

$$R_{ik}^T([i \ k], [i \ k]) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}.$$

Matriko  $R_{ik}^T$  imenujemo *Givensova rotacija*.

Algoritem za splošno matriko velikosti  $m \times n$ ,  $m \geq n$ , je zapisan v algoritmu 12. Če QR razcep računamo zato, da bomo rešili predoločen sistem  $Ax = b$ , matrike  $Q$  ni potrebno izračunati. Namesto tega v vsakem koraku z rotacijo  $R_{jk}^T$  pomnožimo vektor  $b$ , na koncu iz produkta  $\tilde{Q}^T b$  pobereemo prvih  $n$  elementov in rešimo sistem  $Rx = Q^T b$ .

**Opomba 5.3.** Algoritem 12 matriko  $A$  prepiše v matriko  $\tilde{R}$ .

---

**Algorithm 12:** QR razcep (Givensove rotacije)

---

**Data:**  $A \in \mathbb{R}^{m \times n}$

**Result:**  $\tilde{Q} \in \mathbb{R}^{m \times m}$ ,  $\tilde{R} \in \mathbb{R}^{n \times n}$

```

 $\tilde{Q} \leftarrow I_m$  /* če potrebujemo matriko  $\tilde{Q}$  */
for  $j = 1 : n$  do
    for  $k = j + 1 : m$  do
         $r \leftarrow \sqrt{a_{jj}^2 + a_{kj}^2}$ 
        if  $r > 0$  then
             $c \leftarrow a_{jj}/r$ 
             $s \leftarrow a_{kj}/r$ 
             $A([j \ k], j : n) \leftarrow \begin{bmatrix} c & s \\ -s & c \end{bmatrix} A([j \ k], j : n)$ 
             $b([j \ k]) \leftarrow \begin{bmatrix} c & s \\ -s & c \end{bmatrix} b([j \ k])$  /* če rešujemo sistem  $Ax = b$  */
             $\tilde{Q}(1 : m, [j \ k]) \leftarrow \tilde{Q}(1 : m, [j \ k]) \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$  /* če potrebujemo  $\tilde{Q}$  */
        end
    end
end
end

```

---

## 5.4 Householderjeva zrcaljenja

Za neničelen vektor  $w \in \mathbb{R}^n$  definiramo matriko

$$P = I - \frac{2}{w^T w} w w^T.$$

Matriko  $P$  imenujemo *Householderjevo zrcaljenje*.

Množenje vektorja  $x$  z zrcaljenjem  $P$  lahko izvedemo tako, da izračunamo

$$Px = x - \frac{1}{\mu} (w^T x) w,$$

kjer je  $\mu = \frac{1}{2} \|w\|_2^2$ .

---

### Algorithm 13: QR razcep (Householderjeva zrcaljenja)

---

**Data:**  $A \in \mathbb{R}^{m \times n}$

**Result:**  $\tilde{Q} \in \mathbb{R}^{m \times m}$ ,  $\tilde{R} \in \mathbb{R}^{n \times n}$

$\tilde{Q} \leftarrow I_m$  /\* če potrebujemo matriko  $\tilde{Q}$  \*/

**for**  $i = 1 : \min(n, m - 1)$  **do**

določi  $w_i \in \mathbb{R}^{m-i+1}$ , ki prezrcali  $A(i : m, i)$  v  $\pm k e_1 \in \mathbb{R}^{m-i+1}$

$A(i : m, i : n) \leftarrow P_i A(i : m, i : n)$

$b(i : m) \leftarrow P_i b(i : m)$

/\* če rešujemo sistem  $Ax = b$  \*/

$\tilde{Q}(1 : m, i : m) = \tilde{Q}(1 : m, i : m) P_i$

/\* če potrebujemo matriko  $\tilde{Q}$  \*/

**end**

---

## 6 Problemi lastnih vrednosti

Naj bo  $A \in \mathbb{R}^{n \times n}$ . Iščemo lastne vrednosti in lastne vektorje.

**Izrek 6.1.** Za vsako matriko  $A \in \mathbb{R}^{n \times n}$  obstajata unitarna matrika  $U \in \mathbb{C}^{n \times n}$  in zgornja trikotna  $S \in \mathbb{C}^{n \times n}$ , da je

$$A = USU^H.$$

To je *Schurova forma*.

**Izrek 6.2.** Če je  $A \in \mathbb{R}^{n \times n}$ , obstajata ortogonalna matrika  $Q \in \mathbb{R}^{n \times n}$  in kvazi zgornje trikotna (na diagonalni so lahko  $2 \times 2$  bloki) matrika  $R \in \mathbb{R}^{n \times n}$ , da je  $A = QRQ^T$ .

### 6.1 Potenčna metoda

Naj bo  $A \in \mathbb{C}^{n \times n}$ ,  $z_0 \in \mathbb{C}^n$ . Definiramo zaporedje

$$z_{k+1} = \frac{Az_k}{\|Az_k\|}. \quad (1)$$

**Izrek 6.3.** Naj bo  $\lambda_1$  dominantna lastna vrednost matrike  $A \in \mathbb{C}^{n \times n}$ , torej

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|.$$

Za naključno izbran normiran vektor  $z_0 \in \mathbb{C}^n$  zaporedje vektorjev  $z_k$  izračunanih po predpisu 1 po smeri konvergira proti lastnemu vektorju za  $\lambda_1$ .

Kaj je ustrezen zaustavitveni kriterij?

Denimo, da imamo približek  $x$  za lastni vektor in iščemo lastno vrednost. Najboljši približek je  $\lambda \in \mathbb{C}$ , ki minimizira

$$\|Ax - \lambda x\|_2.$$

Rešitev je (iščemo rešitev predločenega sistema  $x\lambda = Ax$  za  $\lambda$ ) *Rayleighov kvocient*

$$\rho(x, A) := \frac{x^H A x}{x^H x},$$

ki je definiran za  $x \neq 0$ .

Primeren zaustavitveni kriterij za potenčno metodo je

$$\|Az_k - \rho(z_k, A)z_k\|_2 < \epsilon.$$

Denimo, da smo s potenčno metodo izračunali lastno vrednost  $\lambda_1$  s pripadajočim normiranim lastnim vektorjem  $x_1$ . Za izračun drugih lastnih parov lahko uporabimo potenčno metodo na reducirani matriki. Dve možnosti redukciji sta:

1. *Householderjeva redukcija*. Z uporabo Householderjeva zrcaljenja poiščemo unitarno matriko  $U$ , da je  $x_1 = Ue_1$ . Potem ima matrika  $B = U^H A U$  obliko

$$B = \begin{bmatrix} \lambda_1 & b^T \\ 0 & C \end{bmatrix}.$$

Preostale lastne vrednosti matrike  $A$  se ujemajo z lastnimi vrednostmi matrike  $C$ .



**Opomba 6.4.** V potenčni metodi dovolj, da znamo izračunati produkt  $Cw$  za vektor  $w \in \mathbb{C}^{n-1}$ . Pri tem si pomagamo z zvezo

$$U^H AU \begin{bmatrix} 0 \\ w \end{bmatrix} = \begin{bmatrix} b^T w \\ Cw \end{bmatrix}$$

2. *Hotellingova redukcija* v primeru simetrične matrike  $A$ . Definiramo

$$B = A - \lambda_1 x_1 x_1^T.$$

Če uporabimo potenčno metodo na matriki  $B$ , dobimo drugo dominantno lastno vrednost matrike  $A$ .

**Opomba 6.5.** Matriko  $B$  ni potrebno eksplicitno izračunati. Dovolj, da uporabimo zvezo

$$Bz = Az - \lambda_1 (x_1^T z) x_1.$$

---

**Algorithm 14:** Potenčna metoda

---

**Data:**  $A \in \mathbb{C}^{n \times n}$ ,  $z_0 \in \mathbb{C}^n$ ,  $|z_0| = 1$ ,  $\varepsilon > 0$

**Result:**  $\lambda_1$

$y_1 \leftarrow Az_0$

$\rho_0 \leftarrow z_0^H y_1$

$k \leftarrow 0$

**while**  $\|y_{k+1} - \rho_k z_k\|_2 \geq \varepsilon$  **do**

$k \leftarrow k + 1$

$z_k \leftarrow y_k / \|y_k\|_2$

$y_{k+1} \leftarrow Az_k$

$\rho_k \leftarrow z_k^H y_{k+1}$

**end**

---

Če iščemo po absolutni vrednosti najmanjšo lastno vrednost nesingularne matrike  $A$ , uporabimo potenčno metodo na  $A^{-1}$ . V algoritmu namesto produkta  $y_{k+1} = A^{-1}z_k$  rešimo sistem  $Ay_{k+1} = z_k$ .

**Opomba 6.6.** Sistem  $Ay_{k+1} = z_k$  lahko rešimo s pomočjo LU razcepa, ki ga dovolj izračunati le enkrat.

## 6.2 Inverzna iteracija

Denimo, da smo izračunali približek za lastno vrednost in potrebujemo še lastni vektor. Tu si lahko pomagamo z *inverzno iteracijo*:

---

**Algorithm 15:** Potenčna metoda

---

**Data:**  $A \in \mathbb{C}^{n \times n}$ , približek za lastno vrednost  $\sigma$ ,  $z_0 \in \mathbb{C}^n$ ,  $|z_0| = 1$

**Result:**  $z \in \mathbb{C}^n$ ,  $Az \approx \sigma z$

**for**  $k = 0 : \infty$  **do**

    reši sistem  $(A - \sigma I)y_{k+1} = z_k$   
     $z_{k+1} = y_{k+1} / \|y_{k+1}\|$

**end**

---

**Opomba 6.7.**

- Inverzna iteracija je potenčna metoda za matriko  $(A - \sigma I)^{-1}$ .
- V praksi potrebujemo le en do dva koraka inverzne iteracije, da iz poljubnega začetnega vektorja izračunamo pripadajoči lastni vektor.

## 6.3 Ortogonalna iteracija

**Definicija 6.8.** Podprostor  $U \leq \mathbb{C}^n$  je invarianten za matriko  $A \in \mathbb{C}^{n \times n}$ , če je

$$\forall x \in U. Ax \in U.$$

Naj bo  $X = \begin{bmatrix} X_1 & X_2 \end{bmatrix}$  nesingularna in  $B = X^{-1}AX = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$ .

Stolpci  $X$  razpenjajo invariantni podprostor za  $A \Leftrightarrow B_{21} = 0$ . Lastne vrednosti  $A$  so unija lastnih vrednosti  $B_{11}$  in  $B_{22}$ .

Naj za lastne vrednosti  $A$  velja:

$$|\lambda_1| \geq \dots \geq |\lambda_p| > |\lambda_{p+1}| \geq \dots \geq |\lambda_n|. \quad (2)$$

Invariantni podprostor  $X_1$  dimenzije  $p$ , ki ustreza lastnim vrednostim  $\lambda_1, \dots, \lambda_p$ , ke potem *dominanti* invariantni podprostor dimenzije  $p$ .

---

**Algorithm 16:** Ortogonalna iteracija

---

**Data:**  $A \in \mathbb{C}^{n \times n}$ ,  $Z_0 \in \mathbb{C}^{n \times p}$ ,  $p \leq n$  z ON stolpci

**Result:**  $Z \in \mathbb{C}^{n \times p}$

**for**  $k = 0 : \infty$  **do**

$Y_{k+1} \leftarrow AZ_k$   
    izračunaj QR razcep  $Y_{k+1} = Q_{k+1}R_{k+1}$   
     $Z_{k+1} \leftarrow Q_{k+1}$

**end**

---

Z uporabo algoritma 16 na diagonali matrike  $Z$  dobimo  $p$  največjih lastnih vrednosti.

**Lema 6.9.** Če velja 2, potem stolpci  $Z_k$  za naključno izbrano matriko  $Z_0$  z ON stolpci konvergirajo proti ortogonalni bazi za dominantni invariantni podprostor dimenzije  $p$ .

Če velja še  $|\lambda_r| > |\lambda_{r+1}|$  za neka  $r < p$ , potem prvih  $r$  stolpcev  $Z_k$  konvergira proti ONB za dominantni invariantni podprostor dimenzije  $r$ .

**Posledica 6.10.** Če za lastne vrednosti matrike  $A \in \mathbb{C}^{n \times n}$  velja

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|,$$

potem za naključno matriko  $Z_0 \in \mathbb{C}^{n \times n}$  z ON stolpci matrika  $Z_k^T A Z_k$  iz ortogonalne iteracije konvergira k Schurovi formi.

## 6.4 QR iteracija

---

**Algorithm 17:** Osnovna verzija QR iteracije

---

**Data:**  $A \in \mathbb{C}^{n \times n}$

**Result:** Lastne vrednosti matrike  $A$

$A_0 \leftarrow A$

**for**  $k = 0 : \infty$  **do**

    izračunaj QR razcep  $A_k = Q_k R_k$

$A_{k+1} \leftarrow R_k Q_k$

**end**

---

**Izrek 6.11.** Matrika  $A_k$  iz QR iteracije je enaka matrike  $Z_k^T A Z_k$ , kjer je  $Z_k$  matrika iz ortogonalne iteracije za  $A$ , kjer vzamemo  $Z_0 = I$ .

Zahtevnost enega koraka zmanjšamo s predhodno redukcijo na zgornjo Hessenbergovo matriko.

### 6.4.1 Redukcija na Hessenbergovo obliko

Vsak korak osnovne QR iteracije zahteva  $O(n^3)$  operacij. Zahtevnost zmanjšamo, če matriko na začetku preoblikujemo v zgornjo Hessenbergovo obliko.

**Definicija 6.12.** Matrika  $A$  je zgornja Hessenbergova, če je  $a_{ij} = 0$  za  $i > j + 1$ .

**Trditev 6.13.** Če je  $A$  zgornja Hessenbergova matrika, se njena oblika med QR iteracijo ohranja.

Realno matriko  $A$  lahko z ortogonalno podobnostno transformacijo  $Q^T A Q = H$  preoblikujemo v zgornjo Hessenbergovo matriko. Za splošno matriko uporabimo Householderjeva zrcaljenja.

---

**Algorithm 18:** Redukcija na zgornjo Hessenbergovo obliko z uporabo zrcaljenj
 

---

**Data:**  $A \in \mathbb{C}^{n \times n}$ **Result:** Ortogonalna matrika  $Q$ , Hessenbergova matrika  $H$ , da  $A = Q^T H Q$  $Q \leftarrow I$  /\* če potrebujemo matriko  $Q$  \*/**for**  $i = 0 : n - 2$  **do**    določi  $w_i \in \mathbb{R}^{n-i}$  za H. zrcaljenje  $P_i$ , ki prezrcali  $A(i+1 : n, i)$  v  $\pm k e_1$      $A(i+1 : n, i : n) \leftarrow P_i A(i+1 : n, i : n)$      $A(1 : n, i+1 : n) \leftarrow A(1 : n, i+1 : n) P_i$      $Q(i+1 : n, 1 : n) \leftarrow P_i Q(i+1 : n, 1 : n)$  /\* če potrebujemo matriko  $Q$  \*/**end**


---

**Definicija 6.14.** Zgornja Hessenbergova matrika  $H$  velikosti  $n \times n$  je *nerazcepna*, če so vsi njeni poddiagonalni elementi  $h_{i+1,i}$  za  $i = 1, \dots, n-1$  neničelni.

Če je  $H$  razcepna, potem problem lastnih vrednosti razpade na dva ali več ločenih problemov. Zato lahko predpostavimo, da je  $H$  nerazcepna.

V praksi proglasimo  $h_{i+1,i}$  za dovolj majhnega, ko je

$$|h_{i,i-1}| < \varepsilon(|h_{i-1,i-1}| + |h_{ii}|).$$

#### 6.4.2 Premiki

Konvergenco lahko pospešimo z vpeljavo premikov, kot predstavljeno v algoritmu 19

---

**Algorithm 19:** QR iteracija s premiki
 

---

**Data:**  $A \in \mathbb{C}^{n \times n}$ **Result:** Premaknjena matrika  $A$  $A_0 \leftarrow A$ **for**  $k = 0 : \infty$  **do**    izberi premik  $\sigma_k$     izračunaj QR razcep  $A_k - \sigma_k I \leftarrow Q_k R_k$      $A_{k+1} \leftarrow R_k Q_k + \sigma_k I$ **end**


---

**Lema 6.15.** Matriki  $A_k$  in  $A_{k+1}$  pri QR iteraciji s premiki sta ortogonalno podobni.

Za hitro konvergenco moramo za premik  $\sigma_k$  izbrati čim boljši prebližek za lastno vrednost. Kaj če izberemo točno lastno vrednost?

**Lema 6.16.** Naj bo  $\sigma$  lastna vrednost nerazcepne zgornje Hessenbergove matrike  $A$ . Če je QR razcep  $A - \sigma I = QR$  in  $B = RQ + \sigma I$ , potem je  $b_{n,n-1} = 0$  in  $b_{nn} = \sigma$ .

Za premik izberemo čim boljši približek za lastno vrednost matrike  $A$ . Uporabljata se naslednji izbiri:

1. *Enojni premik:* za  $\sigma_k$  izberemo  $a_{nn}^{(k)} = \rho(e_n, A_k)$ . Primeren le za matrike z realnimi lastnimi vrednostmi.

2. *Dvojni oz. Francisov premik*: vzamemo podmatriko

$$A_k(n-1:n, n-1:n) = \begin{bmatrix} a_{n-1, n-1}^{(k)} & a_{n-1, n}^{(k)} \\ a_{n, n-1}^{(k)} & a_{nn}^{(k)} \end{bmatrix},$$

ki ima lastni vrednosti  $\sigma_1^{(k)}$  in  $\sigma_2^{(k)}$ . Sedaj naredimo dva premika v enem koraku:

---

**Algorithm 20:** QR iteracija z dvojni premiki

---

**Data:**  $A \in \mathbb{C}^{n \times n}$

**Result:** Premaknjena matrika  $A$

$A_0 \leftarrow A$

**for**  $k = 0 : \infty$  **do**

izračunaj QR razcep  $A_k - \sigma_1^{(k)} I \leftarrow Q_k R_k$

$A'_k \leftarrow R_k Q_k + \sigma_1^{(k)} I$

izračunaj QR razcep  $A'_k - \sigma_2^{(k)} I = Q'_k R'_k$

$A_{k+1} = R'_k Q'_k + \sigma_2^{(k)} I$

**end**

---

## 7 Polinomska interpolacija

### 7.1 Uvod

Dane so točke  $(x_0, y_0), \dots, (x_n, y_n)$ , kjer so  $x_0, \dots, x_n$  paroma različne. Iščemo funkcijo  $f$ , ki *interpolira* te točke, torej

$$\forall i \in \{0, 1, \dots, n\} \cdot f(x_i) = y_i.$$

### 7.2 Lagrangeva interpolacija

Pri polinomske interpolaciji iščemo polinom stopnje največ  $n$ , za katerega velja  $p(x_i) = y_i$  za vse  $i \in \{0, 1, \dots, n\}$ . Tak polinom je *interpolacijski polinom* v točkah  $(x_i, y_i)$ . Točke  $x_0, x_1, \dots, x_n$  so imenujemo *vozli*.

Če tak polinom iščemo v standardni bazi  $1, x, x^2, \dots, x^n$ , potem iščemo koeficiente kot rešitve sistema:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

To je *Vandermondova matrika*, ki je nesingularna za paroma različne točke  $x_0, x_1, \dots, x_n$ . Njena determinanta je  $\prod_{0 \leq i < j \leq n} (x_j - x_i)$ . Torej obstaja enolična rešitev, torej je interpolacijski polinom enoličen.

**Izrek 7.1.** Za paroma različne točke  $x_0, x_1, \dots, x_n$  in vrednosti  $y_0, y_1, \dots, y_n$  obstaja natanko en polinom  $p$  stopnje največ  $n$ , za katerega velja  $p(x_i) = y_i$  za vse  $i \in \{0, 1, \dots, n\}$ .

Lagrangevi bazni polinomi so polinomi:

$$l_{n,i} = \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}.$$

Te polinomi tvorijo razčlenitev enote in bazo za vse polinome stopnje  $n$  ali manj. Ker velja

$$l_{n,i}(x_j) = \delta_{ij},$$

interpolacijski polinom lahko definiramo na naslednji način:

$$p_n(x) := \sum_{k=0}^n y_k l_{n,k}(x).$$

**Izrek 7.2.** Naj bo  $f$   $n+1$ -krat zvezno odvedljiva. Če so  $x_0, \dots, x_n$  paroma različne točke in je  $p$  interpolacijski polinom za  $f$  na  $x_0, x_1, \dots, x_n$ , potem velja:

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

kjer je

$$\omega(x) = (x - x_0)(x - x_1) \cdots (x - x_n) \quad \text{in} \quad \min\{x_0, x_1, \dots, x_n\} < \xi < \max\{x_0, x_1, \dots, x_n\}$$

### 7.3 Deljene difference

**Definicija 7.3.** Za paroma različne točke  $x_0, x_1, \dots, x_k$  in funkcijo  $f$  je *delna difference*  $[x_0, x_1, \dots, x_n]f$  vodilni koeficient  $mx^k$  interpolacijskega polinoma za  $f$  na točkah  $x_0, x_1, \dots, x_k$ .

**Izrek 7.4.** Za paroma različne točke  $x_0, x_1, \dots, x_n$  lahko interpolacijski polinom za  $f$  zapišemo v obliki:

$$p(x) = [x_0]f + (x - x_0)[x_0, x_1]f + (x - x_0)(x - x_1)[x_0, x_1, x_2]f + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})[x_0, x_1, \dots, x_n]$$

Uporabimo bazo  $1, (x - x_0), (x - x_0)(x - x_1), \dots, (x - x_0)(x - x_1) \dots (x - x_{n-1})$ . Tej obliki pravimo *Newtonova oblika interpolacijskega polinoma*.

**Lema 7.5.** Naj bodo  $x_0, x_1, \dots, x_n$  paroma različne točke. potem velja:

1.  $[x_0, x_1, \dots, x_n]f$  je simetrična funkcija glede na točke  $x_0, x_1, \dots, x_n$ , tj. vrstni red  $x_0, x_1, \dots, x_n$  ni pomemben.
2. Deljena difference je linearni funkcional, tj.

$$[x_0, x_1, \dots, x_n](\alpha f + \beta g) = \alpha[x_0, x_1, \dots, x_n]f + \beta[x_0, x_1, \dots, x_n]g.$$

3.  $[x_0, x_1, \dots, x_k]f = \frac{[x_0, x_1, \dots, x_k]f - [x_0, x_1, \dots, x_{k-1}]f}{x_k - x_0}$  za  $k > 0$ ,  $[x_0]f = f(x_0)$ .

Zdaj lahko izračunamo deljene difference po trikotni shemi:

$x_i$	$[x_i]f$	$[\cdot, \cdot]f$	$[\cdot, \cdot, \cdot]f$
$x_0$	$F(x_0)$		
$x_1$	$F(x_1)$	$[x_0, x_1]f$	
$x_2$	$F(x_2)$	$[x_0, x_2]f$	$[x_0, x_1, x_2]f$

**Opomba 7.6.** Velja:

- $[x_0]f = f(x_0)$ ;
- $[x_0, x_1]f = \frac{[x_1]f - [x_0]f}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$ . To je smerni koeficient premice skozi  $(x_0, f(x_0))$  in  $x_1, f(x_1)$ .

Opazimo, če je  $f$  zvezno odvedljiva, je

$$\lim_{x_1 \rightarrow x_0} [x_0, x_1]f = f'(x_0).$$

Interpolacijski polinom  $F(x_0) + \frac{F(x_1) - f(x_0)}{x_1 - x_0}(x - x_0)$  (sekanta) v limiti  $x_1 \rightarrow x_0$  postane tangenta  $f(x_0) + f'(x_0)(x - x_0)$ .

Z uporabo limit lahko definicijo interpolacijskega polinoma razširimo na primere z večkratnimi točkami. Če se točka  $x_i$  pojavi  $m$ -krat, potem se moreta polinom in  $f$  ujemati v  $x_i$  in še v prvih  $m - 1$  odvodih:

$$p(x_i) = f(x_i), p'(x_i) = f'(x_i), \dots, p^{(m-1)}(x_i) = f^{(m-1)}(x_i).$$

Če dopuščamo tudi ponavljanje točk, za deljene difference velja rekurzivna zveza:

$$[x_0, x_1, \dots, x_k]f = \begin{cases} \frac{f^{(k)}(x_0)}{k!}, & x_0 = x_1 = \dots = x_k, \\ \frac{[x_0, x_1, \dots, x_k]f - [x_0, x_1, \dots, x_{k-1}]f}{x_k - x_0}, & \text{sicer.} \end{cases}$$

**Izrek 7.7.** Za  $k$ -krat zvezno odvedljivo funkcijo  $f$  velja:

$$[x_0, x_1, \dots, x_k]f = \int_0^1 dt_1 \int_0^{t_1} dt_2 \dots \int_0^{t_{k+1}} f^{(k)}(\xi_k) dt_k,$$

kjer je

$$\xi_k = t_k(x_k - x_{k-1}) + t_{k-1}(x_{k-1} - x_{k-2}) + \dots + t_1(x_1 - x_0) + x_0.$$

**Posledica 7.8.** Za  $k$ -krat zvezno odvedljivo funkcijo  $f$  velja:

$$[x_0, x_1, \dots, x_k]f = \frac{f^{(k)}(\xi)}{k!},$$

ze nek  $\min\{x_0, x_1, \dots, x_k\} \leq \xi \leq \max\{x_0, x_1, \dots, x_k\}$ .

**Lema 7.9.** Za funkcijo  $f$  in točke  $x_0, x_1, \dots, x_n$  in interpolacijski polinom  $p$  za  $f$  na teh točkah velja:

$$f(x) = p(x) + [x_0, \dots, x_n, x]f \cdot (x - x_0) \dots (x - x_n).$$

**Posledica 7.10.** Če  $p$  interpolira  $(n+1)$ -krat zvezno odvedljivo funkcijo  $f$  na točkah  $x_0, x_1, \dots, x_n$ , potem velja:

$$f(x) = p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot \omega(x),$$

za neki  $\min\{x_0, x_1, \dots, x_n\} \leq \xi \leq \max\{x_0, x_1, \dots, x_n\}$ . Pri tem je  $\xi$  odvisen od  $x$ .