

Numerične metode

Matej Filip

Fakulteta za elektrotehniko
Univerza v Ljubljani

Viri

Viri v slovenščini:

- ▶ Bojan Orel, Osnove numerične matematike, Založba FE in FRI.
- ▶ Bor Plestenjak: Razširjen uvod v numerične metode, DMFA založništvo.

Tuji viri - numerična linearna algebra:

- ▶ G.H. Golub, C.F. Van Loan: Matrix Computations, 3rd edition, Johns Hopkins Univ. Press, Baltimore, 1996.
- ▶ L.N. Trefethen, D. Bau: Numerical Linear Algebra, SIAM, Philadelphia, 1997.
- ▶ J.W. Demmel: Applied Numerical Linear Algebra, SIAM, 1997.

Tuji viri - numerična analiza:

- ▶ K. Atkinson, W. Han: Elementary Numerical Analysis, 3rd edition, John Wiley & Sons, Inc., New Jersey, 2003.
- ▶ R.L. Burden, J.D. Faires, A.M. Burden: Numerical Analysis, 10th edition, Cengage Learning, Boston, 2016.
- ▶ D.R. Kincaid, E.W. Cheney: Numerical Analysis, Mathematics of Scientific Computing, 3rd edition, Brooks/Cole, Pacific Grove, 2002.

Obveznosti

- ▶ Predavanja: 2 ure na teden
- ▶ Vaje: 2 uri na teden
- ▶ Pisni izpit: 50% ocene
- ▶ Izpit iz teorije: 50% ocene

Vsebina predmeta

1. Računanje in vloga napak pri numerični matematiki
2. Reševanje sistemov linearnih enačb
3. Reševanje nelinearnih enačb
4. Numerično odvajanje in integriranje
5. Numerično reševanje diferencialnih enačb

Prvo poglavje:

Uvod v numerično računanje

- ▶ Numerično računanje
- ▶ Predstavljiva števila
- ▶ Zaokrožitvene napake
- ▶ Katastrofalno seštevanje/odštevanje
- ▶ Primeri (ne)stabilnega računanja

Numerično in simbolno računanje

Numerično računanje:

- ▶ Takoj v formulo vstavljamo **števila**
- ▶ Pridemo do numeričnega rezultata - **numerične rešitve**

Simbolno računanje:

- ▶ **simboli** predstavljajo števila
- ▶ izraz preoblikujemo s simbolnim računanjem do novega simbolnega izraza - **analitična rešitev**

Primer

- ▶ *Numerično:*

$$\frac{(17.36)^2 - 1}{17.36 + 1} = 16.36; \quad 0.25, 0.33333 \dots (?), 3.14159 \dots (?)$$

- ▶ *Simbolno:*

$$\frac{x^2 - 1}{x + 1} = x - 1; \quad \frac{1}{4}, \frac{1}{3}, \pi, \tan 83$$

Numerično in simbolno računanje

Primer

```
1 >> x=rand; (x^2-1)/(x+1) - (x-1)
2
3 ans=1.387778780781446e-17
```

Analitično bi rezultat moral biti 0, vendar zaradi numeričnih napak dobimo majhno napako.

Kaj zanima numerično matematiko?

Metoda... matematična konstrukcija, s katero rešujemo problem

Algoritem... koraki metode

Implementacija... zapis algoritma v izbranem jeziku

Kaj pomeni 'biti numerično dober'?

majhna sprememba podatkov $\Rightarrow$ majhna napaka rezultata

Tipična vprašanja numerične matematike:

- ▶ Ali je problem občutljiv?
- ▶ Ali je metoda 'dobra'?
- ▶ Ali je algoritem robusten - deluje na širokem spektru problemov?
- ▶ Ali je implementacija hitra - časovna in prostorska zahtevnost?

Občutljivih problemov NM ne more rešiti

Problem je občutljiv, če se ob majhni spremembi začetnih podatkov točen rezultat zelo spremeni.

Občutljivost je odvisna le od narave problema in ne od izbrane numerične metode.

Primer (presečišča premic)

Sistem in njegova perturbacija

$$x + y = 2 \quad \rightarrow \quad x + y = 1.9999$$

$$x - y = 0 \quad \rightarrow \quad x - y = 0.0002$$

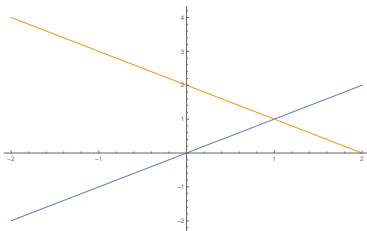
ima rešitvi $x = y = 1$ oz. $x = 1.00005$ in $y = 0.99985$. Problem je neobčutljiv, saj je šlo za spremembo za isti velikostni razred.

Sistem in njegova perturbacija

$$x + 0.99y = 1.99 \quad \rightarrow \quad x + 0.99y = 1.9899$$

$$0.99x + 0.98y = 1.97 \quad \rightarrow \quad 0.99x + 0.98y = 1.9701$$

ima rešitvi $x = y = 1$ oz. $x = 2.97$ in $y = -0.99$. Problem je občutljiv, saj je majhna sprememba začetnih podatkov povzročila veliko spremembo rezultata.



Na čem temeljijo numerične metode?

- ▶ **Matrike nadomestimo z enostavnejšimi** (upoštevamo samo diagonalni ali zgornjetrikotni del).
- ▶ **Nelinearne probleme nadomestimo z linearnimi** (linearna aproksimacija v točki).
- ▶ **Neskončne procese nadomestimo s končnimi** (uporabimo Taylorjev polinom) .
- ▶ **Neskončno razsežne prostore nadomestimo s končno razsežnimi** (funkcije nadomestimo s polinomi).
- ▶ **Diferencialne enačbe nadomestimo z algebraičnimi** (znebimo se vseh parcialnih odvodov iz enačb).

Zakaj sploh potrebujemo numerično matematiko?

Znanost, ki temelji na matematičnih izračunih, je neposredno odvisna od NM.

Nekatere katastrofe so se zgodile zaradi slabega numeričnega računanja (<http://www-users.math.umn.edu/~arnold//disasters/>):

- *Nesreča Misije Patriot, Zalivska vojna 1991, Savdska Arabija, 28 žrtev: slaba analiza zaokrožitvenih napak.*

Čas zadetka iraške rakete, usmerjene na Savdsko Arabijo, je bil računan na vsako desetino sekunde v 24-bitnem sistemu. Ker velja

$$\frac{1}{10} = 2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} + 2^{-12} + 2^{-13} + 2^{-16} + 2^{-17} + 2^{-20} + 2^{-21} +$$
$$\underbrace{+ 2^{-24} + 2^{-25} + 2^{-28} + \dots}_{\text{zanemarimo}},$$

je vsako desetinko sekunde napaka $9.5 \cdot 10^{-8}$ s. Po 100 urah računanja je bila napaka $9.5 \cdot 10^{-8}$ s $\cdot 100 \cdot 60 \cdot 60 \cdot 10 = 0.34$ s. Ker je hitrost rakete 1.676 m/s, je bila pozicija rakete za več kot 500 m napačno predvidena in je ta ušla radarjem.

- ▶ *Eksplozija rakete Ariana 5, Francoska Gvajana, 1996:*
posledica prekoračitve obsega števil.

https://www.youtube.com/watch?v=PK_yguLapgA

<https://www.youtube.com/watch?v=W3YJeoYgozw>

Ob prenovi rakete so 'pozabili' nadgraditi uporabljen številski sistem, ki je horizontalno hitrost meril v 16-bitnem sistemu (1 bit porabimo za predznak). Največja hitrost v tem sistemu je

$$2^0 + 2^1 + \dots + 2^{13} + 2^{14} = \frac{2^{15} - 1}{2 - 1} = 32767.$$

Ker je prenovljena raketa po 37 sekundah presegla to hitrost, je prišlo do zaustavitve motorjev...

- ▶ *Potop naftne ploščadi Sleipner A, Stavanger, Norveška, 1991, milijarda dolarjev škode: nenatančna obdelava obremenitev pri reševanju PDE-jev.*

<https://www.youtube.com/watch?v=eGdiPs4THW8>

Ponovitev predstavljivih števil

Števila shranjujemo v obliki

$$x = \pm 0.d_1 d_2 d_3 \dots d_m \times \beta^e,$$

kjer je

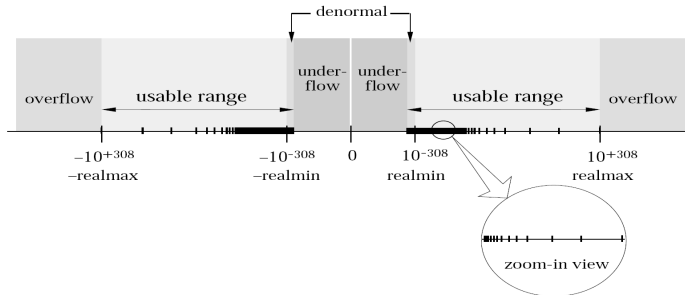
- ▶ β naravno število (v računalništvu $\beta = 2$),
- ▶ $d_1 d_2 d_3 \dots d_m$ mantisa, e eksponent.

Primer (baza 10)

- ▶ *1000.12345* zapišemo kot $+(0.100012345)_{10} \times 10^4$.
- ▶ *0.000812345* zapišemo kot $+(0.812345)_{10} \times 10^{-3}$.

Prekoračitev in podkoračitev

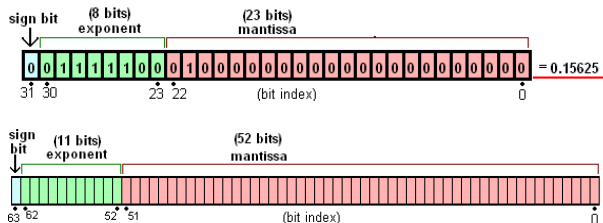
Floating Point Number Line



- ▶ izračuni preblizu 0 lahko povzročijo **podkoračitev**
- ▶ preveliki izračuni lahko povzročijo **prekoračitev**
- ▶ prekoračitev je v splošnem hujši problem

IEEE standard

- ▶ IEEE Enojna natančnost: števila so predstavljena z 32 biti.
- ▶ IEEE Dvojna natančnost: števila so predstavljena z 64 biti.



Kaj so zaokrožitvene napake?

- ▶ Večine realnih števil ne moremo predstaviti v strojni aritmetiki $\Rightarrow$ **zaokrožujemo** in delamo **zaokrožitvene napake**.
- ▶ IEEE standard... **zaokroži x do najbližjega predstavljivega števila $\text{fl}(x)$** . Naj bosta

$$x_- \leq x \leq x_+$$

najbližji predstavljivi števili števila x . Potem je

$$\text{fl}(x) = \begin{cases} x_-, & \text{če je } x \text{ bližje } x_-, \\ x_+, & \text{če je } x \text{ bližje } x_+. \end{cases}$$

- ▶ Kako velika je napaka? Recimo, da je x bližje x_- :

$$\begin{aligned} x &= (0.b_1b_2b_3 \dots b_mb_{m+1})_2 \times 2^e, \\ x_- &= (0.b_1b_2b_3 \dots b_m)_2 \times 2^e, \\ x_+ &= ((0.b_1b_2b_3 \dots b_m)_2 + 2^{-m}) \times 2^e, \end{aligned}$$

$$\text{fl}(x) = x(1 + \delta), |\delta| < 2^{-m}$$

Absolutna napaka:

$$x - x_- \leq \frac{x_+ - x_-}{2} = 2^{e-m-1}.$$

Relativna napaka:

$$\frac{x - x_-}{x} \leq \frac{2^{e-m-1}}{1/2 \times 2^e} \leq \underbrace{2^{-m}}_u \dots \text{osnovna zaokrožitvena napaka}$$

Torej je

$$x_- = x_- - x + x \geq -ux + x = x(1 - u).$$

Podobno

$$x_+ \leq x(1 + u).$$

Sledi

$$\boxed{\text{fl}(x) = x(1 + \delta)}, \quad \text{kjer je } |\delta| < u.$$

Kako računamo s predstavljenimi števili?

Za **predstavljeni** števili x, y in katerokoli od osnovnih operacij $\odot \in \{+, -, \cdot, :\}$ število $x \odot y$ ni nujno predstavljlivo. Po zgornjem pa velja

$$\boxed{\text{fl}(x \odot y) = (x \odot y)(1 + \delta)}, \quad \text{kjer je } |\delta| \leq u.$$

Seštevanje numerično **ni asociativna operacija**, tj.

$$\boxed{(a + b) + c \neq a + (b + c)} :$$

Primer

```
1 >> a=rand;b=rand;c=rand;((a+b)+c)-(a+(b+c))
2
3 ans=-2.220446049250313e-16
```

Seštevamo od manjših k večjim številom

$$\begin{aligned}(a + b) + c &= \text{fl}(\text{fl}(a + b) + c) = \text{fl}((a + b)(1 + \delta_1) + c) \\&= [(a + b)(1 + \delta_1) + c](1 + \delta_2) \\&= [(a + b + c) + (a + b)\delta_1](1 + \delta_2) \\&= (a + b + c) \left[1 + \frac{a + b}{a + b + c} \delta_1(1 + \delta_2) + \delta_2 \right]\end{aligned}$$

Podobno

$$a + (b + c) = (a + b + c) \left[1 + \frac{b + c}{a + b + c} \delta_3(1 + \delta_4) + \delta_4 \right].$$

Če pozabimo na člena $\delta_1\delta_2$ in $\delta_3\delta_4$ (Zakaj to lahko naredimo?), dobimo

$$(a + b) + c = (a + b + c)(1 + \epsilon_3) \quad \text{kjer je} \quad \epsilon_3 \approx \frac{a + b}{a + b + c} \delta_1 + \delta_2,$$

$$a + (b + c) = (a + b + c)(1 + \epsilon_4) \quad \text{kjer je} \quad \epsilon_4 \approx \frac{b + c}{a + b + c} \delta_3 + \delta_4.$$

Sklep: Ko seštevamo števila, je za čim manjšo napako najbolje začeti z najmanjšim in prištevati večje.

Napake pri numeričnem računanju

- ▶ Neodstranljiva napaka $D_n \dots$ nenatančni začetni podatki.
- ▶ Napaka metode $D_m \dots$ npr. neskončni proces aproksimiramo s končnim.
- ▶ Zaokrožitvena napaka $D_z \dots$ računanje s približki in zaokroževanje.

Celotna napaka D je

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

Primer ($\sin \frac{\pi}{10}$ računamo v desetiškem sistemu z $m = 4$)

- ▶ D_n : $fl(\frac{\pi}{10}) = 0.3142 \cdot 10^0$. Ocenimo: $|D_n| \approx \sin'(\frac{\pi}{10})|x - fl(x)| \leq \frac{1}{2} \cdot 10^{-4}$.
- ▶ D_m : $\sin x \approx x - x^3/6$. Ocenimo: $|D_m| \leq x^5/120 = 2.6 \cdot 10^{-5}$.
- ▶ D_z : $fl(x - fl(fl(fl(x \cdot x) \cdot x)/6))$. Ocenimo: $|D_z| \leq 3.0 \cdot 10^{-5}$.

Stabilnost meri kakovost metode

Stabilnost metode preverimo z **analizo zaokrožitvenih napak**.

Vrste napak (x naj bo točna vrednost, $\bar{x}$ pa približek zanjo):

▶ Prva delitev:

▶ **Absolutna napaka**: $\bar{x} - x$.

▶ **Relativna napaka**: $\frac{\bar{x} - x}{x}$.

▶ Druga delitev:

▶ **Direktna napaka**: Numerična napaka rezultata.

▶ **Obratna napaka**: Koliko je potrebno spremeniti začetne podatke, da dobimo izračunan rezultat.

Velja

$$|\text{direktna napaka}| \approx \text{občutljivost} \times |\text{obratna napaka}|.$$

Izračunana vrednost je blizu pravi, če rešujemo neobčutljiv problem z obratno stabilno metodo.

Odštevanje in seštevanje sta lahko 'katastrofalni'

odštevanje dveh približno enakih števil

seštevanje dveh približno nasprotnih števil

$$a = x.xxxx\ xxxx\ xxx1 \overbrace{ssss \dots}^{\text{izguba}}$$

$$b = x.xxxx\ xxxx\ xxx0 \overbrace{tttt \dots}^{\text{izguba}}$$

$$\begin{array}{rcl} & \text{končna natančnost} & \\ & \overbrace{x.xxx\ xxxx\ xxx1} & \\ \text{Potem} \quad & - \quad x.xxx\ xxxx\ xxx0 & \\ \hline & = \quad 0.000\ 0000\ 0001 & \quad \quad \quad \text{???? ????} \\ & = \quad 1. \quad \underbrace{\text{???? ????}}_{\text{izguba natančnosti}} \cdot \beta^{-m} & \end{array}$$

S ponavljanjem se napake seštevajo.

Primer katastrofalnega odštevanja

Iščemo rešitve kvadratne enačbe

$$x^2 + 2ax + b = 0, \quad \text{kjer je } a > 0 \text{ in } a^2 > b.$$

Rešitev z manjšo absolutno vrednostjo je

$$x_2 = \frac{-2a + \sqrt{4a^2 - 4b}}{2} = -a + \sqrt{a^2 - b}.$$

$$1 \quad k_1 := a^2$$

$$2 \quad k_2 := k_1 - b$$

$$3 \quad k_3 := \sqrt{k_2}$$

$$4 \quad k_4 := -a + k_3$$

Če je a^2 veliko večji od b , potem ima lahko korak 4 veliko napako. Možna rešitev:

$$x_2 = (-a + \sqrt{a^2 - b}) \cdot \frac{a + \sqrt{a^2 - b}}{a + \sqrt{a^2 - b}} = \frac{-b}{a + \sqrt{a^2 - b}}.$$

```
1  $k_1 := a^2$   
2  $k_2 := k_1 - b$   
3  $k_3 := \sqrt{k_2}$   
4  $k_4 := a + k_3$   
5  $k_5 := \frac{-b}{k_4}$ 
```

```
1 >> a = 10000;  
2 >> b = -1;  
3 >> x = -a+sqrt(a^2 - b)  
4 x = 5.0000000055588316e-05  
5  
6 >> x^2 + 2 * a * x + b  
7 ans = 1.361766321927860e-08  
8  
9 >> x = -b/(a+sqrt(a^2-b))  
10 x = 4.999999987500000e-05  
11  
12 >> x^2 + 2 * a * x + b  
13 ans = -1.110223024625157e-16
```

Računanje s stabilnejšo obliko

- ▶ Izračun vrednosti funkcije

$$f(x) = x(\sqrt{x+1} - \sqrt{x})$$

ni stabilen za velike x , ker je $\sqrt{x+1} \approx \sqrt{x}$. Tej težavi se lahko izognemo:

$$f(x) = f(x) \cdot \frac{\sqrt{x+1} + \sqrt{x}}{\sqrt{x+1} + \sqrt{x}} = \frac{x}{\sqrt{x+1} + \sqrt{x}}.$$

- ▶ Vrsto

$$\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \dots + \frac{1}{n(n+1)},$$

ki se sešteje v $\frac{n}{n+1}$ (dokaz: indukcija), je bolje numerično računati vzvratno kot

$$\frac{1}{n \cdot (n+1)} + \frac{1}{(n-1) \cdot n} + \dots + \frac{1}{1 \cdot 2}.$$

Seštevanje in odštevanje v splošnem nista relativno direktno stabilni operaciji

$x, y \in \mathbb{R}$. Računamo približek $\bar{p}$ za $p = x + y$.

$$\begin{aligned}\bar{p} &= \text{fl}(\text{fl}(x) + \text{fl}(y)) = \text{fl}(x(1 + \delta_1) + y(1 + \delta_2)) \\ &= (x(1 + \delta_1) + y(1 + \delta_2))(1 + \delta_3) \\ &= x(1 + \delta_1)(1 + \delta_3) + y(1 + \delta_2)(1 + \delta_3) \\ &= x + y + x(\delta_1 + \delta_3 + \delta_1\delta_3) + y(\delta_2 + \delta_3 + \delta_2\delta_3)\end{aligned}$$

kjer je $|\delta_i| \leq u$. Relativna napaka je

$$\boxed{\frac{|\bar{p} - p|}{|p|} \leq \frac{|x(\delta_1 + \delta_3 + \delta_1\delta_3) + y(\delta_2 + \delta_3 + \delta_2\delta_3)|}{|x + y|}}.$$

Torej:

Če je $x + y$ blizu 0, potem je $\frac{|\bar{p} - p|}{|p|}$ veliko.

Množenje (in deljenje) je relativno direktno stabilna operacija

$x, y \in \mathbb{R}$. Računamo približek $\bar{p}$ za $p = x \cdot y$.

$$\begin{aligned}\bar{p} &= \text{fl}(\text{fl}(x) \cdot \text{fl}(y)) = \text{fl}(x(1 + \delta_1) \cdot y(1 + \delta_2)) \\ &= x(1 + \delta_1) \cdot y(1 + \delta_2)(1 + \delta_3) \\ &= xy(1 + \delta_1 + \delta_2 + \delta_3 + \text{produkti več } \delta),\end{aligned}$$

kjer je $|\delta_i| \leq u$. Relativna napaka je

$$\boxed{\frac{|\bar{p} - p|}{|p|} \leq \frac{|xy||\delta_1 + \delta_2 + \delta_3 + \mathcal{O}(u^2)|}{|xy|} = |\delta_1 + \delta_2 + \delta_3 + \mathcal{O}(u^2)|}.$$

Torej:

Relativna napaka $\frac{|\bar{p} - p|}{|p|}$ ni odvisna od velikosti produkta xy .

Večina numeričnih metod ni relativno direktno stabilnih

Vse numerične metode, kjer sta vključeni

operaciji $+$ / $-$

in kot rezultat lahko dobimo npr. vrednost 0 ali nekje po poti kot vmesno vrednost skoraj singularno matriko, **niso relativno direktno stabilne**, tj. v rezultatu je lahko veliko relativna napaka.

Zato moramo vedno premisliti:

1. V katerih primerih so zgodi velika napaka?
2. Kako nestabilne primere preoblikovati v stabilne?

Primeri takih operacij:

- ▶ Računanje vrednosti polinoma.
- ▶ Računanje skalarnega produkta.
- ▶ Reševanje linearnega sistema.
- ▶ $\vdots$

Drugo poglavje:

Linearni sistemi

$$Ax = b$$

- ▶ Vektorske in matrične norme
- ▶ Pogojenostno število $\kappa(A)$
- ▶ Direktne metode za reševanje
 - ▶ *LU* razcep
 - ▶ Pivotna rast $\rho(A)$
 - ▶ Razcep Choleskega
- ▶ Predoločeni sistemi
 - ▶ *QR* razcep
 - ▶ Householderjeva zrcaljenja
 - ▶ *SVD* razcep

Vektorska norma je preslikava $\|\cdot\| : \mathbb{C}^n \rightarrow \mathbb{R}$, ki zadošča:

1. Pozitivna definitnost: $\|x\| \geq 0$ za vsak $x \in \mathbb{C}^n$ in $\|x\| = 0 \Leftrightarrow x = 0$.
2. Homogenost: $\|\alpha x\| = |\alpha| \|x\|$ za vsaka $\alpha \in \mathbb{C}$ in $x \in \mathbb{C}^n$
3. Trikotniška neenakost: $\|x + y\| \leq \|x\| + \|y\|$ za vsaka $x, y \in \mathbb{C}^n$.

Primer

Naj bo $x = (x_1, \dots, x_n) \in \mathbb{C}^n$.

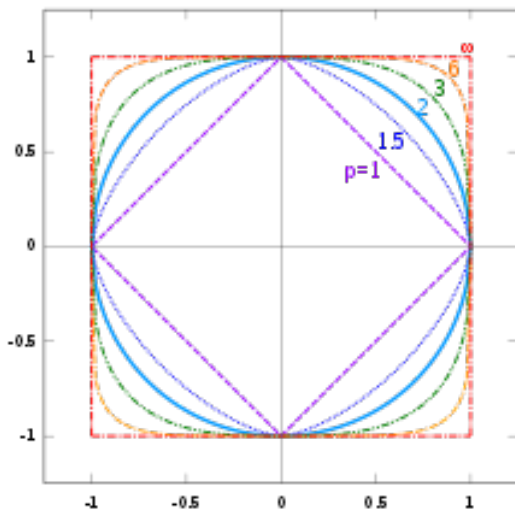
► *p*-norma, $p \in \mathbb{N}$:

$$\|x\|_p := (|x_1|^p + \dots + |x_n|^p)^{1/p}.$$

► *Supremum norma*:

$$\|x\|_\infty = \max(|x_1|, \dots, |x_n|).$$

Enotske krožnice v različnih normah



Matrična norma je preslikava $\|\cdot\| : \mathbb{C}^{n \times n} \rightarrow \mathbb{R}$, ki zadošča:

1. **Pozitivna definitnost:** $\|A\| \geq 0$ za vsak $A \in \mathbb{C}^{n \times n}$ in $\|A\| = 0 \Leftrightarrow A = 0$.
2. **Homogenost:** $\|\alpha A\| = |\alpha| \|A\|$ za vsaka $\alpha \in \mathbb{C}$ in $A \in \mathbb{C}^{n \times n}$.
3. **Trikotniška neenakost:** $\|A + B\| \leq \|A\| + \|B\|$ za vsaka $A, B \in \mathbb{C}^{n \times n}$.
4. **Submultiplikativnost:** $\|AB\| \leq \|A\| \|B\|$ za vsaka $A, B \in \mathbb{C}^{n \times n}$.

Trditev

Naj bo $\|\cdot\|_*$ vektorska norma na $\mathbb{C}^n$. Potem predpis

$$\|A\|_* := \max_{\|x\|=1} \|Ax\|_* = \max_{x \neq 0} \frac{\|Ax\|_*}{\|x\|_*}.$$

določa matrično normo na $\mathbb{C}^{n \times n}$.

Dokaz

Naj bo $A = [a_{ij}]_{i,j=1,\dots,n} \in \mathbb{C}^{n \times n}$ matrika. Nekaj matričnih norm:

1. **1-norma:** $\|x\|_1 = \sum_{i=1}^n |x_i|$ (1-norma),

$$\|A\|_1 = \max_{j=1,\dots,n} \left(\sum_{i=1}^n |a_{ij}| \right). \quad \text{Dokaz}$$

2. **Spektralna norma:** Tu $\lambda_j(X)$ označuje j -to lastno vrednost matrike X .

$$\|A\|_2 = \sqrt{\max_{j=1,\dots,n} \lambda_j(A^T A)}.$$

3. **Frobeniusova norma:**

$$\|A\|_F = \sqrt{\sum_{i,j=1}^n |a_{ij}|^2}.$$

4. **Supremum norma:**

$$\|A\|_\infty = \max_{i=1,\dots,n} \left(\sum_{j=1}^n |a_{ij}| \right).$$

Zakaj imeti več matričnih norm?

Nekatere norme je bistveno zahtevneje izračunati od ostalih. Zahtevno je npr. določanje spektralne norme $\|\cdot\|_2$, saj je računanje lastnih vrednosti zahtevna naloga. Poceni pa je izračunati 1–normo, ∞ –normo in F –normo. Iz različnih ocen, kot so

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

pa lahko dobro ocenimo $\|A\|_2$.

Občutljivost sistema $Ax = b$

Zanima nas, kako na spremembo rešitve x vpliva napaka v začetnih podatkih, tj. napaka v A in b .

Zanima nas torej, kako velik je Δx v primeru majhnih perturbacij ΔA in Δb v rešitvi

$$(A + \Delta A)(x + \Delta x) = b + \Delta b. \quad (1)$$

Radi bi ocenili relativno napako $\frac{\|\Delta x\|}{\|x\|}$, kjer je $\|\cdot\|$ neka vektorska norma.

Izberimo vektorsko normo $\|\cdot\|$. Definirajmo **občutljivost** oz. **pogojenostno število** obrnljive matrike A v normi $\|\cdot\|$:

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

$\kappa(A)$ meri občutljivost sistema $Ax = b$

Izrek

Naj bo A v (1) obrnljiva matrika.

1. Privzemimo, da je $\Delta A = 0$. Potem velja:

$$\frac{\|\Delta x\|}{\|x\|} \leq \kappa(A) \frac{\|\Delta b\|}{\|b\|}.$$

2. Naj bo $\Delta A \neq 0$, naj za identično matriko I velja $\|I\| = 1$ in naj bo še $\|A^{-1}\| \|\Delta A\| < 1$. Potem velja:

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

Primer

1. Če se spomnimo primera računanja prečišča dveh premic iz prvih predavanj, lahko vidimo, da je vprašanje občutljivosti glede na začetne podatke v resnici vprašanje občutljivosti matrik

$$A_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{in} \quad A_2 = \begin{pmatrix} 1.00 & 0.99 \\ 0.99 & 0.98 \end{pmatrix}.$$

Za njiju velja:

$$\kappa_1(A_1) = \kappa_F(A_1) = \kappa_\infty(A_1) = 2, \quad \kappa_2(A_1) = 1,$$

$$\kappa_1(A_2) = \kappa_\infty(A_2) = 3.96 \cdot 10^4, \quad \kappa_2(A_2) = 2, \quad \kappa_F(A_2) = 3.92 \cdot 10^4,$$

kjer κ_* označuje občutljivost v matrični normi $*$. Kot smo se geometrijsko prepričali, je drugi sistem res občutljiv, prvi pa ne.

2. Primer zelo občutljive matrike je Hilbertova matrika

$$H_n = \left[\frac{1}{i+j-1} \right]_{i,j} \in \mathbb{R}^{n \times n}.$$

Ta se pojavi pri iskanju polinoma, ki se v normi $\|f\| = \sqrt{\int_0^1 f^2 dx}$ najboljše prilega dani funkciji, saj je $\int_0^1 x^{i+j} dx = \frac{1}{i+j+1}$. Velja $\kappa_2(H_5) \approx 4.8 \cdot 10^5$, za naključno 5×5 matriko pa velja $\kappa_2 \approx 100$.

Primer

1. Če z Matlabom z ukazom `\` rešimo sistem $H_{15}x = v$, kjer je

$$v = H_{15} \cdot [1, \dots, 1]^T,$$

bi morali dobiti za rezultat $x = [1, \dots, 1]$. Toda

$$\|x - [1, \dots, 1]^T\|_2 = 5.3 \cdot 10^{-3}.$$

Direktne metode

$$Ax = b$$

- ▶ Gaussova-eliminacija
- ▶ LU razcep
- ▶ Pivotiranje
- ▶ Razcep Choleskega

Reševanje kvadratnih linearnih sistemov

Linearni sistem n enačb z n neznankami $x_1, \dots, x_n$ je oblike

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2,$$

$$\vdots \quad \quad \quad \vdots$$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n,$$

kjer so a_{ij}, b_j realna števila.

V matrični obliki ga zapišemo kot

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \dots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_x = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_b.$$

Geometrijski pomen sistema $Ax = b$

Naj bodo $a_{(1)}, a_{(2)}, \dots, a_{(n)}$ stolpci matrike A , tj.,

$$a_{(i)} := \begin{bmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{ni} \end{bmatrix} \in \mathbb{R}^n$$

Linearna kombinacija vektorjev $a_{(1)}, a_{(2)}, \dots, a_{(n)}$ je vsak vektor oblike

$$x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{n2} \end{bmatrix} + \dots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{nn} \end{bmatrix}, \quad (2)$$

kjer so $x_i \in \mathbb{R}$ realna števila.

Zanima nas, ali obstaja linearna kombinacija (2), ki je enaka vektorju b .

Sistem $Ax = b$ z vidika numerične matematike

- ▶ Kako **drago** je reševanje sistema $Ax = b$?
cena=število osnovnih računskih operacij (+, −, ·, :).
- ▶ Kateri **problemi** in **napake** se pojavijo med reševanjem $Ax = b$?
Ali obstajajo slabe matrike? Kako take matrike identificirati?
Vemo že, da so slabe matrike z velikim $\kappa(A)$.
Kaj pa, če $\kappa(A)$ ni velik?
- ▶ Za katere matrike se da **enostavno** in **poceni** rešiti tak sistem?

Ponovitev Gaussove eliminacije (GE)

Cilj je pretvoriti sistem v zgornjetrikotnega, nato pa ga rešiti z obratno substitucijo.

Primer

Rešujemo $Ax = b$, kjer sta

$$A = \begin{bmatrix} -3 & 2 & -1 \\ 6 & -6 & 7 \\ 3 & -4 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} -1 \\ -7 \\ -6 \end{bmatrix}.$$

Tvorimo *razširjen sistem*

$$\tilde{A} = [A \mid b] = \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 6 & -6 & 7 & -7 \\ 3 & -4 & 4 & -6 \end{array} \right]$$

Prištejemo 2-kratnik prve vrstice drugi in 1-kratnik prve vrstice tretji.

$$\tilde{A}_{(1)} = \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & -2 & 3 & -7 \end{array} \right]$$

Primer

Odštejemo 1-kratnik druge vrstice od tretje

$$\tilde{A}_{(2)} = \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & 0 & -2 & 2 \end{array} \right]$$

Rešimo z *obratno substitucijo*

$$x_3 = \frac{2}{-2} = -1,$$

$$x_2 = \frac{1}{-2} (-9 - 5x_3) = 2,$$

$$x_1 = \frac{1}{-3} (-1 - 2x_2 + x_3) = 2.$$

V nadaljevanju bomo:

1. Prešteli število potrebnih računskih operacij za Gaussovo eliminacijo (GE).
2. GE bomo zapisali s pomočjo matričnih množenj.
3. Ukvarjali se bomo s stabilnostjo GE.

Algoritem GE in cena GE

```
1  - $n \times n$  matrika  $A = [a_{ij}]_{ij}$  in  $n \times 1$  vektor  $b = [b_i]_i$ 
2  -preoblikujemo  $[A|b]$  v zgornjetrikotno z GE
3
4  for  $k = 1 \dots n - 1$ 
5      for  $i = k + 1 \dots n$ 
6           $xmult = a_{ik} / a_{kk}$ 
7           $a_{ik} = 0$ 
8          for  $j = k + 1 \dots n$ 
9               $a_{ij} = a_{ij} - (xmult) a_{kj}$ 
10         end
11          $b_i = b_i - (xmult) b_k$ 
12     end
13 end
```

Izrek

Število računskih operacij (+, −, ·, :) za prevedbo matrike A in razširjene matrike $[A|b]$ v zgornjetrikotno obliko je

$$\frac{2}{3}n^3 + \mathcal{O}(n^2). \quad \text{Dokaz}$$

Obratna substitucija in število operacij

```
1  -zgornjetrikotna  $n \times n$  matrika  $U = [u_{ij}]_{i,j}$ , vektor  
    $c = [c_i]_i$   
2  -resimo sistem  $Ux = c$   
3  
4   $x_n = c_n / u_{nn}$   
5  for  $i = n - 1 \dots 1$   
6      $s = c_i$   
7     for  $j = i + 1 \dots n$   
8          $s = s - u_{ij}x_j$   
9     end  
10     $x_i = s / u_{ii}$   
11 end
```

Izrek

Število računskih operacij $(+, -, \cdot, :)$ za rešitev sistem $Ux = c$ je

$$n^2. \quad \text{Dokaz}$$

Motivacija za zapis GE v matrični obliki

Videli smo, da je cena pretvorba matrike A oz. sistema $[A|b]$ v zgornjetrikotno obliko bistveno dražja kot pa obratna substitucija.

Če bomo v nekem postoku reševali sisteme $Ax = b$ pri **fiksni matriki A** , **vektor b pa se bo spreminjal**, bi bilo iz računskega vidika bistveno učinkoviteje preoblikovanje matrike A v zgornjetrikotno obliko narediti samo enkrat.

Ključno v tem procesu je ugotoviti, **kako moramo preblikovati vektor b** , ne da bi delali GE na razširjenem sistemu.

LU razcep matrike A

```
1  -Vhod:  $A = [a_{ij}]_{i,j}$   $n \times n$  matrika.
2  -Izhod: Spodnja trikotna matrika  $L$  in zgornja
    trikotna matrika  $U$ , da je  $A = LU$ 
3  - $\ell_{ik}$  v spodnjem algoritmu so elementi pod
    diagonalo v  $L$ , na diagonali so same 1
4  -preostali elementi  $a_{ij}$  v zgornjem trikotniku so
    elementi matrike  $U$ 
5
6  for  $k = 1, \dots, n-1$ 
7      for  $i = k+1, \dots, n$ 
8           $\ell_{ik} = a_{ik}/a_{kk}$ 
9          for  $j = k+1, \dots, n$ 
10              $a_{ij} = a_{ij} - \ell_{ik} a_{kj}$ 
11         end
12     end
13 end
```

Izrek

Število računskih operacij $(+, -, \cdot, :)$ za izračun LU razcepa matrike A je $\frac{2}{3}n^3 + \mathcal{O}(n^2)$.

Prema substitucija in število operacij

```
1  -Vhod: spodnja trikotna  $n \times n$  matrika  $L = [\ell_{ij}]_{i,j}$  in  
   vektor  $b = [b_i]_i$   
2  -Izhod: resitev  $y$  sistema  $Ly = b$   
3  
4   $y_1 = b_1 / \ell_{11}$   
5  for  $i = 2 \dots n$   
6       $s = b_i$   
7      for  $j = 1 \dots i - 1$   
8           $s = s - \ell_{ij} y_j$   
9      end  
10      $y_i = s / \ell_{ii}$   
11 end
```

Izrek

Število računskih operacij $(+, -, \cdot, :)$ za rešitev sistem $Ly = b$ je

$$n^2.$$

Reševanje sistema $Ax = b$ prek LU razcepa:

1. Izračunamo $A = LU$. Cena: $\frac{2}{3}n^3 + \mathcal{O}(n^2)$.
2. Rešimo $Ly = b$ s premo substitucijo, tj. od y_1 proti y_n .
Cena: $n^2 - n$.
3. Rešimo $Ux = y$ z obratno substitucijo, t. od x_n proti x_1 .
Cena: n^2 .

Cena preme substitucije je za n operacij manjša kot cena obratne substitucije, saj imamo ne diagonalni L same enice in prihranimo v vsaki spremenljivki eno deljenje.

Reševanje sistema $Ax = b$ prek LU razcepa

Primer

$$A = \begin{pmatrix} 2 & 1 & 3 & -4 \\ -4 & -1 & -4 & 7 \\ 2 & 3 & 5 & -3 \\ -2 & -2 & -7 & 9 \end{pmatrix}, \quad b = \begin{pmatrix} 8 \\ -14 \\ 7 \\ -16 \end{pmatrix}.$$

1. $L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ -1 & -1 & 1 & 1 \end{pmatrix}, U = \begin{pmatrix} 2 & 1 & 3 & -4 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & -2 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$

2. Rešimo $Ly = b$ in dobimo $y = (8 \quad 2 \quad -5 \quad -1)^T.$

3. Rešimo $Ux = y$ in dobimo $x = (1 \quad -1 \quad 1 \quad -1)^T.$

Obstoj LU razcepa matrike

V nadaljevanju se bomo ukvarjali z **obstojem** in **stabilnostjo LU razcepa**.

Problematična sta npr. matriki

$$A = \begin{bmatrix} 0 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad B = \begin{bmatrix} 10^{-17} & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix},$$

saj je 10^{-17} pod strojnimi ϵ . Da se natančno povedati, kdaj LU razcep obstaja.

Podmatriki matrike $A \in \mathbb{R}^{n \times n}$, zožene na prvih k vrstic in stolpcih, pravimo **k -ta glavna vodilna podmatrika**.

Izrek (Obstoj LU razcepa)

Za $n \times n$ matriko A sta naslednji trditvi ekvivalentni:

- 1. LU razcep matrike A obstaja in je enoličen.*
- 2. k -ta glavna vodilna podmatrika matrike A je obrnljiva za vsak $k = 1, \dots, n$.*

LU razcep z delnim pivotiranjem

Pri **delnem pivotiranju** pred eliminacijo v j -tem stolpcu primerjamo elemente

$$a_{jj}, a_{j+1,j}, \dots, a_{nj},$$

nato pa **zamenjamo j -to vrstico** s tisto, ki vsebuje element z **največjo absolutno vrednostjo**.

Menjava j -te in k -te vrstice pa je **množenje z leve s permutacijsko matriko** P_{jk} , ki se od identitete razlikuje le v j -ti in k -ti vrstici, ki sta zamenjani:

$$P_{jk} = I_n - E_{jj} - E_{kk} + E_{jk} + E_{kj}.$$

Tu so E_{ij} standardne koordinatne matrike (1 v i -ti vrstici in j -tem stolpcu in 0 drugje).

LU razcep z delnim pivotiranjem - algoritem

```
1  -Vhod:  $A = [a_{ij}]_{i,j}$   $n \times n$  matrika
2  -Izhod: permutacijska matrika  $P$ , spodnja in
      zgornja trikotna matrika  $L$  in  $U$ , da je
       $PA = LU$ 
3
4   $P$  in  $L$  identicni  $n \times n$  matriki
5  for  $k = 1, \dots, n-1$ 
6      poisci  $q$ -to in  $k$ -to vrstico, ki zadosca
           $|a_{qk}| = \max_{k \leq p \leq n} |a_{pk}|$ 
7      zamenjaj  $q$ -to in  $k$ -to vrstico v matrikah  $A, P$ 
          in strogem spodnjem trikotniku  $L$ 
8      for  $i = k+1, \dots, n$ 
9           $\ell_{ik} = a_{ik}/a_{kk}$ 
10         for  $j = k+1, \dots, n$ 
11              $a_{ij} = a_{ij} - \ell_{ik} a_{kj}$ 
12         end
13     end
14 end
```

LU razcep z delnim pivotiranjem

Izrek ($\mathcal{O}$ računski zahtevnosti LU razcep z delnim pivotiranjem)

Število računskih operacij $(+, -, \cdot, :)$ za izračun LU razcepa z delnim pivotiranjem je $\frac{2}{3}n^3 + \mathcal{O}(n^2)$.

Dodatno delo pri LU razcepu z delnim pivotiranjem je $\mathcal{O}(n^2)$ primerjanj in menjav.

Reševanje $Ax = b$ prek LU razcepa z delnim pivotiranjem:

1. Izračunamo $PA = LU$. Cena: $\frac{2}{3}n^3 + \mathcal{O}(n^2)$.
2. Rešimo $Ly = Pb$ s premo substitucijo. Cena: $n^2 - n$.
3. Rešimo $Ux = y$ z obratno substitucijo. Cena: n^2 .

Izrek (Obstoj LU razcepa z delnim pivotiranjem)

Za $n \times n$ matriko A sta naslednji trditvi ekvivalentni:

1. LU razcep matrike A z delnim pivotiranjem obstaja.
2. Matrika A je obrnljiva.

$Ax = b$ prek LU razcepa z delnim pivotiranjem

Primer.

$$A = \begin{pmatrix} 2 & 1 & 3 & -4 \\ -4 & -1 & -4 & 7 \\ 2 & 3 & 5 & -3 \\ -2 & -2 & -7 & 9 \end{pmatrix}, \quad b = \begin{pmatrix} 8 \\ -14 \\ 7 \\ -16 \end{pmatrix}.$$

$$1. \quad L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & -\frac{3}{5} & 1 & 0 \\ -\frac{1}{2} & \frac{1}{5} & -\frac{1}{8} & 1 \end{pmatrix}, \quad U = \begin{pmatrix} -4 & -1 & -4 & 7 \\ 0 & \frac{5}{2} & 3 & \frac{1}{2} \\ 0 & 0 & -\frac{16}{5} & \frac{58}{10} \\ 0 & 0 & 0 & \frac{1}{8} \end{pmatrix},$$
$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

2. Rešimo $Ly = Pb$ in dobimo $y = (-14 \quad 0 \quad -9 \quad -\frac{1}{8})^T$.

3. Rešimo $Ux = y$ in dobimo $x = (1 \quad -1 \quad 1 \quad -1)^T$.

LU s kompletnim pivotiranjem

Pri kompletnem pivotiranju pred eliminacijo v j -tem stolpcu poiščemo element z največjo absolutno vrednostjo v podmatriki $A(j : n, j : n)$ in nato izvedemo ustrezni menjavi vrstic in stolpcev.

Dodatno delo pri LU razcepu s **kompletnim pivotiranjem** je $\mathcal{O}(n^3)$ primerjanj in menjav. Torej je skupna cena **precej dražja** od LU razcepa z delnim pivotiranjem. Ker bomo videli, da je LU razcep z delnim pivotiranjem statistično numerično stabilen, se v praksi kompletno pivotiranje **redko uporablja**.

Stabilnost LU razcepa matrike A

Sistem $Ax = b$ smo rešili prek LU razcepa in dobili približek $\hat{x}$. Računali smo v treh korakih:

1. *Izračun LU razcepa:* $A + E = \hat{L}\hat{U}$.
2. *Prema substitucija:* $\hat{L}\hat{y} = b$.
3. *Obratna substitucija:* $\hat{U}\hat{x} = \hat{y}$.

Izkaže se, da je (teoretično) nestabilen samo prvi korak.

Spomnimo se, da z u označujemo osnovno zaokrožitveno napako 2^{-m} kjer je m dolžina mantise. Z $|A| = [|a_{ij}|]_{i,j}$ označimo matriko absolutnih vrednosti vhodov matrike $A = [a_{ij}]_{i,j}$

Izrek (Ocena absolutne napake pri izračunu LU razcepa)

Naj bo $A \in \mathbb{R}^{n \times n}$ obrnljiva matrika, pri kateri se izvede LU razcep brez pivotiranja. Za izračunani matriki $\hat{L}, \hat{U}$ velja $A = \hat{L}\hat{U} + E$, kjer je

$$|E| \leq 3(n-1)u \left(|A| + |\hat{L}||\hat{U}| \right) + \mathcal{O}(u^2). \quad \text{Dokaz}$$

Stabilnost LU razcepa matrike A z delnim pivotiranjem

Iz zgornjega izreka sledi

$$\begin{aligned}\|E\|_{\infty} &\leq 3(n-1)u \cdot \left(\|A\|_{\infty} + (\|\hat{L}\| \|\hat{U}\|)_{\infty} \right) + \mathcal{O}(u^2) \\ &\leq 3(n-1)u \cdot \left(\|A\|_{\infty} + \|\hat{L}\|_{\infty} \|\hat{U}\|_{\infty} \right) + \mathcal{O}(u^2) \\ &\leq 3(n-1)u \|A\|_{\infty} + 3(n-1)n \|\hat{U}\|_{\infty} + \mathcal{O}(u^2),\end{aligned}$$

kjer smo v drugi neenakosti upoštevali submultiplikativnost, v tretji neenakosti pa to, da so pri LU razcepu z delnim pivotiranjem vsi elementi matrike L navzgor omejeni z 1. Zato velja $\|L\|_{\infty} \leq n$. Torej je relativna napaka v supremum normi navzgor omejena z

$$\frac{\|E\|_{\infty}}{\|A\|_{\infty}} \leq 3(n-1)u + 3(n-1)nu \cdot \frac{\|\hat{U}\|_{\infty}}{\|A\|_{\infty}} + \mathcal{O}(u^2).$$

Izrek (Ocena relativne napake pri izračunu LU razcepa)

Pri LU razcepu z delnim pivotiranjem velja ocena relativne napake:

$$\frac{\|E\|_{\infty}}{\|A\|_{\infty}} \leq 3(n-1)u + 3(n-1)nu \cdot \frac{\|\hat{U}\|_{\infty}}{\|A\|_{\infty}} + \mathcal{O}(u^2).$$

LDL razcep simetrične matrike A

Trditev

Naj bo

$$A = A^T$$

$n \times n$ simetrična matrika in $A = LU$ njen LU razcep. Če je D diagonalna matrika, katere diagonalna se ujema z diagonalno U -ja, potem je $U = DL^T$ in

$$A = LDL^T.$$

Dokaz. Velja

$$LU = A = A^T = (LU)^T = U^T L^T.$$

Z množenjem te verige enakosti z leve z L^{-1} in z desne z $(L^T)^{-1}$ dobimo

$$U(L^T)^{-1} = L^{-1}U^T =: D.$$

Ker je leva stran zgornja trikotna, desna pa spodnja trikotna, je D diagonalna matrika. Torej velja

$$A = LU = LDL^T.$$

Razcep Choleskega pozitivno definitne matrike A

Izrek (Razcep Choleskega)

Naj bo A simetrična in pozitivno definitna matrika, tj. za vsak $x \neq 0$ velja $x^T A x > 0$. (ekvivalentno, vsaka lastna vrednost je > 0) Potem obstaja spodnjetrikotna matrika V , da velja

$$A = VV^T.$$

Temu razcepu pravimo **razcep Choleskega** matrike A . Matrika V je enaka $V := LD^{1/2}$, kjer sta L in D matriki iz LDL razcepa matrike A .

Dokaz. Za obstoj je potrebno preveriti samo to, da $D^{1/2}$ res lahko izračunamo, tj. da so diagonalni elementi matrike D pozitivni. Da dobimo i -ti element na diagonalni D , izračunamo $x^T A x$ za vektor $x = (L^T)^{-1} e_i$, kjer je e_i i -ti stolpec identitete. Ker je A pozitivno definitna, je $x^T A x > 0$.

Razcep Choleskega - algoritem

```
1   $A = [a_{ij}]_{i,j}$  je dana  $n \times n$  matrika
2  ce se razcep v celoti izvede, je rezultat
   spodnjetrikotna matrika  $V$  iz  $A = VV^T$ 
3
4  for  $k = 1, \dots, n$ 
5       $v_{kk} = \sqrt{a_{kk} - \sum_{i=1}^{k-1} v_{ki}^2}$ 
6      for  $j = k + 1, \dots, n$ 
7          for  $i = 1, \dots, k - 1$ 
8               $a_{jk} = a_{jk} - v_{ji}v_{ki}$ 
9          end
10          $v_{jk} = a_{jk}/v_{kk}$ 
11     end
12 end
```

Razcep Choleskega

Izrek (Cena razcepa Choleskega)

Število računskih operacij (+, −, ·, :) za izračun razcepa Choleskega pozitivno definitne matrike A je $\frac{n^3}{3} + \mathcal{O}(n^2)$.

Dokaz

Razcep Choleskega tako zahteva samo **pol toliko operacij** kot LU razcep in je **najcenejši** numerični način za ugotavljanje **pozitivne definitnosti** simetrične matrike.

Reševanje sistema $Ax = b$ prek razcepa Choleskega:

1. Izračunamo $A = VV^T$. Cena: $\frac{1}{3}n^3 + \mathcal{O}(n^2)$.
2. Rešimo $Vy = b$ s premo substitucijo. Cena: $n^2 + \mathcal{O}(n)$.
3. Rešimo $V^Tx = y$ z obratno substitucijo. Cena: $n^2 + \mathcal{O}(n)$.

Izrek (Stabilnost računanja razcepa Choleskega)

Računanje razcepa Choleskega je *numerično stabilna metoda*.

Predoločeni sistemi

$$Ax = b$$

- ▶ Normalni sistem
- ▶ QR razcep prek Gram-Schmidtove ortogonalizacije
- ▶ QR razcep prek Householderjevih zrcaljenj

Predoločeni sistemi

Za matriko $A \in \mathbb{R}^{n \times m}$, $n \geq m$, in vektor $b \in \mathbb{R}^n$, iščemo vektor $x \in \mathbb{R}^m$, ki zadošča:

$$Ax = b. \quad (3)$$

Kadar je $n > m$, sistemu (3) pravimo predoločen, točna rešitev pa najverjetneje ne obstaja. Zato nas navadno zanima rešitev, ki v izbrani vektorski normi $\|\cdot\|$ minimizira ostanek, tj. želimo, da je $\|Ax - b\|$ čim manjše. Če za $\|\cdot\|$ izberemo $\|\cdot\|_2$, potem se problemu reče **linearni problem najmanjših kvadratov**:

$$\text{Poišči } x \in \mathbb{R}^m, \text{ ki minimizira } \|Ax - b\|_2. \quad (4)$$

Primeri uporabe so:

- ▶ prilagajanje krivulj,
- ▶ statistično modeliranje podatkov,
- ▶ geodetsko modeliranje.

Primer (Kubična interpolacija)

Dane so točke $(x_1, y_1), \dots, (x_n, y_n)$. Iščemo pa kubični polinom

$$p(x) = a_3x^3 + a_2x^2 + a_1x + a_0,$$

ki se podatkom najboljše prilega po metodi najmanjših kvadratov. Torej iščemo koeficiente $a_0, \dots, a_3 \in \mathbb{R}$, tako da je vsota

$$\sum_{i=1}^n (a_3x_i^3 + a_2x_i^2 + a_1x_i + a_0 - y_i)^2$$

najmanjša možna.

To lahko napišemo kot sistem

$$\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ \vdots & & & \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

ki ga rešujemo po metodi najmanjših kvadratov.

Reševanje prek normalnega sistema

Trditev

Naj bo $\text{rank}(A) = m$. Rešitev sistema (3) po metodi najmanjših kvadratov je $x \in \mathbb{R}^m$, ki reši t.i. **normalni sistem**:

$$A^T A x = A^T b. \quad (5)$$

Dokaz

Težava, ki se pojavi pri reševanju (5), je numerična stabilnost. Računanje skalarne produkta v splošnem ni relativno stabilna operacija. Vhodi matrike $A^T A$ pa so ravno skalarni produkti stolpcev A .

Primer - linearna regresija

Iščemo premico, ki se najbolje prilega podatkom

$$(x_1, y_1), \dots, (x_n, y_n)$$

po metodi najmanjših kvadratov. Premica je oblike

$$y = a + bx.$$

Torej sta spremenljivki a in b . Sistem lahko zapišemo v obliki

$$\underbrace{\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_n \end{bmatrix}}_A \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\vec{b}}. \quad (6)$$

Vemo, da je rešitev (6) enaka

$$\vec{x} = (A^T A)^{-1} A^T \vec{b} = \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix}.$$

Reševanje normalnega sistema - razcep Choleskega

Če je matrika $A \in \mathbb{R}^{n \times m}$ ranga m , potem je matrika $A^T A$ pozitivno definitna (kar pomeni, da so vse lastne vrednosti > 0) in sistem $A^T A x = A^T b$ lahko rešimo s pomočjo razcepa Choleskega:

1. Izračunaj $B = A^T A$ in $c = A^T b$.

Cena: $nm^2 + \mathcal{O}(mn)$. (Zadosti je računati zgornjetrikotni del AA^T .)

2. Izračunaj razcep Choleskega $B = VV^T$. Cena: $\frac{1}{3}m^3 + \mathcal{O}(m^2)$.

3. Reši $Vy = A^T b$. Cena: $\mathcal{O}(m^2)$.

4. Reši $V^T x = y$. Cena: $\mathcal{O}(m^2)$.

Skupna cena: $nm^2 + \frac{1}{3}m^3 + \mathcal{O}(nm)$.

QR razcep

QR razcep matrice $A \in \mathbb{R}^{n \times m}$ sta ortogonalna matrika $Q \in \mathbb{R}^{n \times m}$ ($Q^T Q = I_m$) in zgornjetrikotna matrika $R \in \mathbb{R}^{m \times m}$, ki zadoščata

$$A = QR. \quad (7)$$

Iz (7) sledi, da sta stolpična prostora matrik A in Q enaka. Pogoji ortogonalnosti matrice Q pa pomeni, da so njeni stolpci normirani (tj. dolžine 1) in paroma pravokotni. Če označimo z $a_1, \dots, a_m$ in $q_1, \dots, q_m$ stolpce matrik A in Q ter $R = [r_{ij}]_{i,j}$, potem veljajo zveze:

$$\begin{aligned} q_1 &= \frac{1}{r_{11}} a_1, \\ q_2 &= \frac{1}{r_{22}} (a_2 - r_{12} q_1), \\ &\vdots, \\ q_m &= \frac{1}{r_{mm}} (a_m - r_{1m} q_1 - \dots - r_{m-1,m} q_{m-1}). \end{aligned} \quad (8)$$

Iz (8) lahko izpeljemo enega od načinov za izračun QR razcepa, tj. z uporabo Gram-Schmidtove ortogonalizacije (GSO)

```
1   $A = [a_1, \dots, a_m]$  je  $n \times m$  matrika s stolpci  $a_1, \dots, a_m$ 
2  Rezultat sta matriki  $q = [q_1, \dots, q_m]$  in  $R = [r_{ij}]$ , da je
    $A = QR$ 
3
4   $r_{11} = \|a_1\|_2$ 
5   $q_1 = \frac{1}{r_{11}} a_1$ 
6  for  $j = 2, \dots, m$ 
7       $q_j = a_j$ 
8      for  $i = 1, \dots, j$ 
9           $r_{ij} = q_i^T a_j$ 
10          $q_j = q_j - r_{ij} q_i$ 
11     end
12      $r_{jj} = \|q_j\|_2$ 
13      $q_j = \frac{1}{r_{jj}} q_j$ 
14 end
```

Uporaba QR razcepa za reševanje sistema (3)

Izrek (Računska zahtevnost GSO)

Število računskih operacij (+, −, ·, :) za izračun QR razcepa z GSO je

$$\approx 2nm^2.$$

Dokaz

Trditev (Reševanje sistema prek QR razcepa)

Naj bo $A \in \mathbb{R}^{n \times m}$ in $\text{rank}(A) = m$. Rešitev sistema (3) po metodi najmanjših kvadratov je enaka rešitvi zgornjetrikotnega sistema

$$Rx = Q^T b, \tag{9}$$

kjer je $A = QR$ za ortogonalno matriko Q in zgornjetrikotno matriko R .

Dokaz

Householderjeva zrcaljenja (HZ)

Pri GS postopku izračunamo

$$v \leftarrow a_k - \sum_{i=1}^{k-1} (q_i^\top a_k) q_i.$$

Če je a_k skoraj v $\text{span}\{q_i\}$, potem

$$\sum_{i=1}^{k-1} (q_i^\top a_k) q_i \approx a_k,$$

zato je razlika majhna: odštevamo dve veliki, skoraj enaki števili $\Rightarrow$ katastrofalna izguba pomembnih števk.

Zato želimo QR razcep narediti na drugačen način z uporabo ortogonalnih matrik.

Zrcaljenje preko hiperravnine, ki je pravokotna na vektor $w \in \mathbb{R}^m$, imenovano **Householderjevo zrcaljenje (HZ)**, je predstavljeno z matriko

$$P_w = I - \frac{2}{\|w\|_2^2} ww^T,$$

kjer $I = I_m$.

Trditev (Lastnosti HZ)

- ▶ P_w je ortogonalna: $P_w^T = P_w$ in $P_w^2 = I$.
- ▶ Za $x = \alpha w + u$, kjer je $u \perp w$, velja $P_w x = -\alpha w + u$.

Matrike P_w ne izračunamo, ampak zadošča hraniti le vektor w , saj je

$$P_w x = x - \frac{2}{\|w\|_2^2} (w^T x) w.$$

Trditev

Naj bosta $x, y \in \mathbb{R}^m$ s $\|x\|_2 = \|y\|_2$. Za $w = x - y$ velja

$$P_w x = \left(I - 2 \frac{ww^\top}{\|w\|_2^2} \right) x = y.$$

Dokaz: Zapišimo

$$w^\top x = (x - y)^\top x = \|x\|_2^2 - x^\top y,$$

$$\|w\|_2^2 = \|x - y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 - 2x^\top y = 2(\|x\|_2^2 - x^\top y),$$

kjer smo uporabili $\|x\|_2 = \|y\|_2$. Zato je $\frac{w^\top x}{\|w\|_2^2} = \frac{1}{2}$ in

$$P_w x = x - 2 \cdot \frac{w^\top x}{\|w\|_2^2} w = x - w = y.$$

Lemma

Naj bo $x = [x_1 \ x_2 \ \cdots \ x_m]^\top \in \mathbb{R}^m$, $e_1 = [1 \ 0 \ \cdots \ 0]^\top$ in $\rho = \|x\|_2$. Definirajmo $w_\pm = x \mp \rho e_1$. Tedaj velja

$$\|w_\pm\|_2^2 = 2\rho(\rho \mp x_1) = 2\|x\|_2(\|x\|_2 \mp x_1).$$

Dokaz:

$$\|w_\pm\|_2^2 = \|x \mp \rho e_1\|_2^2 = (x_1 \mp \rho)^2 + \sum_{i=2}^m x_i^2.$$

Ker je $\rho^2 = \|x\|_2^2 = x_1^2 + \sum_{i=2}^m x_i^2$, dobimo

$$\|w_\pm\|_2^2 = x_1^2 \mp 2\rho x_1 + \rho^2 + \sum_{i=2}^m x_i^2 = \underbrace{(x_1^2 + \sum_{i=2}^m x_i^2)}_{=\rho^2} + \rho^2 \mp 2\rho x_1 = 2\rho^2 \mp 2\rho x_1$$

Za numerično stabilnost izberemo predznak, ki poveča $\rho \mp x_1$: če je $x_1 < 0$, vzamemo $w_- = x - \rho e_1$ (izraz $\rho - x_1 = \rho + |x_1|$ je večji); če je $x_1 \geq 0$, vzamemo $w_+ = x + \rho e_1$ (izraz $\rho + x_1$ je večji).

Izračun QR razcepa prek HZ:

Matriko A preoblikujemo v zgornjetrikotno R z množenjem z leve z $m - 1$ HZ-ji:

1. S H_1 preslikamo prvi stolpec A v večkratnik e_1 .
2. Drugi stolpec $H_1 A$ od diagonale navzdol v večkratnik e_2 z $\tilde{H}_2 = 1 \oplus H_2$.
3. Tretji stolpec $\tilde{H}_2 H_1 A$ od diagonale navzdol v večkratnik e_3 z $\tilde{H}_3 = I_2 \oplus H_2$.
4. Nadaljujemo ta postopek:

$$R = \tilde{H}_k \tilde{H}_{k-1} \cdots \tilde{H}_2 \tilde{H}_1 A, \quad Q = \tilde{H}_1^\top \tilde{H}_2^\top \cdots \tilde{H}_k^\top.$$

Za rešitev predločenega sistema preko metode najmanjših kvadratov $A \in \mathbb{R}^{n \times m}$ ($n \geq m$), rešen s Householderjevimi zrcaljenji ($A = QR$, $Q^\top b = y$, $Rx = y$), je cena $\approx 2nm^2 - \frac{2}{3}m^3$.

Izračun QR razcepa prek HZ - grafično

$$A = \begin{pmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{pmatrix} \xrightarrow{H_1} A_1 = \begin{pmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 \\ 0 & H_2 \end{pmatrix} \xrightarrow{\quad} A_2 = \begin{pmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & x & x & x \end{pmatrix} \xrightarrow{\quad} A_3 = \begin{pmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & x & x \end{pmatrix}$$

$$\begin{pmatrix} I_2 & 0 \\ 0 & H_3 \end{pmatrix}$$

$$\begin{pmatrix} I_3 & 0 \\ 0 & H_4 \end{pmatrix} \xrightarrow{\quad} A_4 = \begin{pmatrix} x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{pmatrix}.$$

Primerjava dveh :

Cholesky: $\approx nm^2 + \frac{1}{3}m^3$,

QR (Householder): $\approx 2nm^2 - \frac{2}{3}m^3$.

Zakaj Householder QR namesto Cholesky pri reševanju normalnega sistema?:

- ▶ $\kappa(A^\top A) = \kappa(A)^2 \Rightarrow$ relativna napaka $\sim \kappa(A)^2 \varepsilon$. Pri QR je $\sim \kappa(A) \varepsilon$.
- ▶ **Stabilnost:** za Householder QR velja $Q^\top Q = I + O(\varepsilon)$.
- ▶ **Izogremo se tvorjenju $A^\top A$:** manj kopičenja zaokrožitvenih napak

Če je $n = m$ dobimo pri QR (Householder) $\frac{4}{3}n^3$ (tukaj boljše uporabiti LU razcep z delnim pivotiranjem)

Reševanje nelinearnih enačb in optimizacija

$$* f(x) = 0$$

$$* f_i(x_1, x_2, \dots, x_n) = 0, \quad i = 1, \dots, n$$

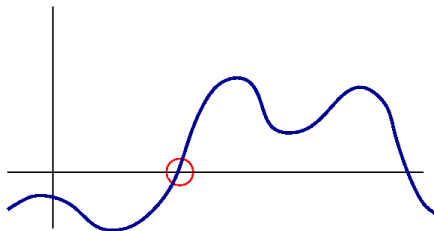
$$* \min\{f(x) : x \in K \subseteq \mathbb{R}^n\}$$

- ▶ Ena enačba v eni spremenljivki: Bisekcija, tangentna metoda, sekantna metoda, navadna iteracija
- ▶ Sistem n enačb v n spremenljivkah: Newtonova in Jacobijeva iteracija

Motivacija

Problem: Naj bo dana funkcija $f(x)$. Poišči x , ki zadošča

$$f(x) = 0.$$

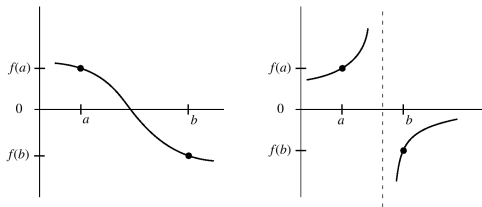


- ▶ **Nelinearni sistemi niso tako enostavno rešljivi** kot sistemi linearnih enačb.
- ▶ Ničel polinoma stopnje 5 ne moremo zapisati analitično.
- ▶ Kako reševati take probleme? Z **iterativnim postopkom**, pri čemer se rešitvam čim bolj približamo.

Osnovna strategija reševanja

1. Skiciraj funkcijo.

- ▶ Postavimo **začetno domnevo**, kaj je lahko ničla.
- ▶ **Ničla** x gotovo **obstaja** na intervalu $[a, b]$, če imata $f(a)$ in $f(b)$ **različna predznaka** in je funkcija f zvezna na $[a, b]$.
- ▶ Toda: **Sprememba predznaka funkcije ne pomeni vedno**, da je na tem intervalu ničle, kajti lahko imamo na intervalu singularnost:



- ## 2. Začnemo z **začetno domnevo** in uporabimo nek **iteracijski algoritem**.

Konvergenčni kriteriji za x

Zaustavitveni kriterij je odvisen od narave problema, ki ga rešujemo:

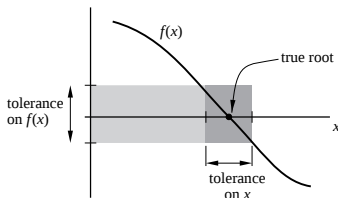
- ▶ Lahko nas zanima, kdaj velja

$$|x_k - x_{k-1}| < \text{toleranca}.$$

- ▶ Lahko pa nas zanima, kdaj velja

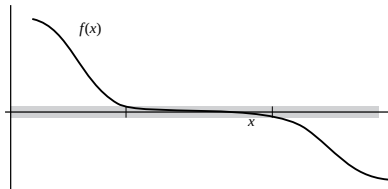
$$|f(x_k)| < \text{toleranca}.$$

- ▶ Še najboljše pa je zahtevati izpolnjenost **obeh pogojev** hkrati.

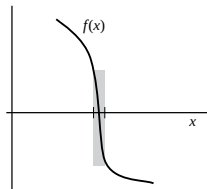


Primerjava obeh konvergenčnih kriterijev

Če je $f'(x)$ majhen v okolici ničle, je lažje zadostiti toleranci na funkcijsko vrednost.



Če je $f'(x)$ velik v bližini ničle, je možno zadostiti toleranci na dolžino intervala, četudi je $|f(x)|$ še vedno velik.



Povezava med obema kriterijama

Vprašanje: Kako sta kriterija na x in $f(x)$ povezana med sabo?

Ko x_a in x_b konvergirata proti x^* , gre razmerje

$$\frac{f(x_b) - f(x_a)}{x_b - x_a} \quad \text{proti} \quad f'(x^*)$$

Zato lahko pričakujemo, da velja

$$|f(x_b) - f(x_a)| \approx |f'(x^*)| |x_b - x_a|,$$

ko x_a in x_b konvergirata proti x^* .

Zaključek: $|f'(x^*)|$ določa povezavo med kriterijema.

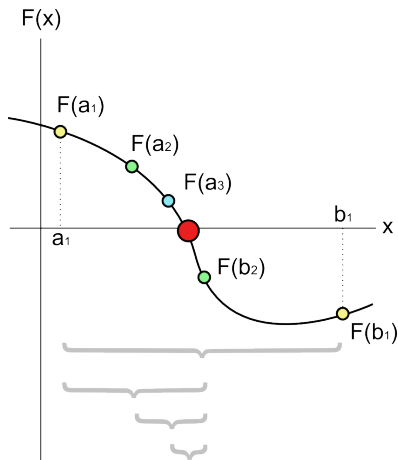
Bisekcija

Razpolovišče začetnega intervala $[a, b]$ je točka

$$x_m = \frac{1}{2}(a + b).$$

Postopek:

1. Poišči razpolovišče.
2. Izmed dveh možnih intervalov izberi tistega, kjer ima funkcija različno predznačeni krajišči.
3. Nadaljujemo s prvim korakom.
4. Ustavimo se, ko je interval krajši od naprej predpisane tolerance.



Algoritem za bisekcijo

```
1  Vhod: funkcija  $f$ , krajisci  $a$ ,  $b$ , toleranca  $tol$   
2  Izhod: priblizek  $x^*$  za  $f(x) = 0$  z  $|x - x^*| < tol$   
3  
4  for  $k = 1, 2, \dots$   
5       $x_m = a + (b - a)/2$   
6      if  $\text{sign}(f(x_m)) = \text{sign}(f(a))$   
7           $a = x_m$   
8      else  
9           $b = x_m$   
10     end  
11     if  $|b - a| < tol$ , stop  
12 end
```

Hitrost konvergence in računska zahtevnost

Naj bo δ_n velikost intervala po n -tem koraku bisekcije. Potem velja

$$\delta_0 = b-a, \quad \delta_1 = \frac{1}{2}\delta_0, \quad \delta_2 = \frac{1}{2}\delta_1 = \frac{1}{4}\delta_0, \quad \dots, \quad \delta_n = \left(\frac{1}{2}\right)^n \delta_0$$

$$\Rightarrow \quad \frac{\delta_n}{\delta_0} = \left(\frac{1}{2}\right)^n = 2^{-n} \quad \text{ali} \quad n = \log_2 \left(\frac{\delta_n}{\delta_0}\right)$$

n	$\frac{\delta_n}{\delta_0}$	število izračunov funkcijskih vrednosti
5	3.1×10^{-2}	7
10	9.8×10^{-4}	12
20	9.5×10^{-7}	22
30	9.3×10^{-10}	32
40	9.1×10^{-13}	42
50	8.9×10^{-16}	52

Navadna iteracija

Pri metodi navadne iteracije osnovno enačbo

$$f(x) = 0$$

preoblikujemo v ekvivalentno

$$x = g(x)$$

in izvajamo iteracijo

$$x_{r+1} = g(x_r)$$

pri izbranem začetnem približku x_0 .

Nekatere možne izbire za iteracijsko funkcijo g so denimo

$$g(x) = x - f(x)$$

$$g(x) = x - Cf(x), \quad C \neq 0$$

$$g(x) = x - h(x)f(x), \quad h(x) \neq 0.$$

Konvergenco navadne iteracije opisuje naslednji izrek.

Izrek

Naj iteracijska funkcija g na intervalu $I = [\alpha - \delta, \alpha + \delta]$ zadošča pogoju

$$|g(x) - g(y)| \leq m|x - y|, \quad x, y \in I, \quad 0 \leq m < 1.$$

Potem za vsak $x_0 \in I$ zaporedje

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

konvergira k α . Velja

$$|x_r - \alpha| \leq m^r |x_0 - \alpha|$$

in

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

Posledica

Naj bo $g(\alpha) = \alpha$ in g zvezno odvedljiva pri α . Če je $|g'(\alpha)| < 1$, potem obstaja taka okolica I za α , da za vsak $x_0 \in I$ zaporedje

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

konvergira k α . O hitrosti konvergence v bližini α odloča število $g'(\alpha)$.

Definicija

Naj zaporedje $(x_r)_{r=0}^{\infty}$ konvergira k α . Pravimo, da je red konvergence enak p , če obstajata konstanti $C_1, C_2 > 0$, da za dovolj pozne člene zaporedja velja

$$C_1|x_r - \alpha|^p \leq |x_{r+1} - \alpha| \leq C_2|x_r - \alpha|^p.$$

Lema

Naj bo iterativna funkcija g v okolici negibne točke $\alpha = g(\alpha)$ p -krat zvezno odvedljiva in naj bo

$$|g^{(k)}(\alpha)| = 0 \quad \text{za } k = 1, 2, \dots, p-1, \quad \text{ter} \quad g^{(p)}(\alpha) \neq 0.$$

Potem ima iterativna metoda

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

v bližini rešitve α red konvergence p .

Posebni primeri konvergence so:

- ▶ $p = 1$: linearna (na vsakem koraku pridobimo konstantno mnogo novih točnih decimal),
- ▶ $p = 2$: kvadratična (na vsakem koraku se število točnih decimal podvoji),
- ▶ $p = 3$: kubična (na vsakem koraku se število točnih decimal potroji),
- ▶ $1 < p < 2$: superlinearna (hitrejša od linearne in počasnejša od kvadratične).

Primer

Ena od možnih iteracijskih funkcij za računanje $\sqrt{a}$, $a > 0$, je

$$g(x) = \frac{x^2 + a}{2x}.$$

Očitno je $g(\sqrt{a}) = \sqrt{a}$, $g'(\sqrt{a}) = 0$ in $g''(\sqrt{a}) \neq 0$. Iterativna metoda

$$x_{r+1} = g(x_r)$$

ima torej v bližini $\sqrt{a}$ kvadratično konvergenco.

Izberimo $a = 10$ in $x_0 = 3$. Potem je

r	x_r
0	3.00000000
1	3.16666667
2	3.16228070
3	3.16227766

V zadnjem stolpcu so točne decimalke, ki se na vsakem koraku približno podvojijo, kar potrjuje kvadratično konvergenco.

Tangentna metoda

Ideja za tangentno metodo je preprosta: nov približek je presečišče tangente v prejšnjem približku z abscisno osjo. Če torej rešujemo enačbo

$$f(x) = 0,$$

se zaporedje približkov glasi

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)}, \quad r = 0, 1, \dots$$

Očitno je to poseben primer navadne iteracije z iteracijsko funkcijo

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

Za konvergenco tangentne metode velja:

Če je ničla α funkcije f enostavna, je konvergenca vsaj kvadratična. Kvadratična je, če je $f''(\alpha) \neq 0$, sicer je vsaj kubična.

Izrek

Naj bo α enostavna ničla dvakrat zvezno odvedljive funkcije f . Potem obstajata okolica I točke α in konstanta C , da tangentna metoda konvergira za vsak $x_0 \in I$ in približki x_r zadoščajo oceni

$$|x_{r+1} - \alpha| \leq C(x_r - \alpha)^2.$$

Izrek

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

Sekantna metoda

Pri tangentni metodi poleg vrednosti potrebujemo tudi odvod. Če ta ni na voljo ali ga težko računamo, ga aproksimiramo z diferencialnim kvocientom

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

Tako dobimo sekantno metodo

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

Izkaže se, da je red konvergence zanjo približno $p \approx 1.62$ (superlinearna).

Primerjava metod: zahteve

▶ **Bisekcija**

- ▶ Začetni interval $[a, b]$ z $f(a) \cdot f(b) < 0$.
- ▶ Ne potrebuje odvoda.

▶ **Tangentna metoda**

- ▶ Potrebuje začetni približek x_0 .
- ▶ Potrebuje odvod $f'(x)$.

▶ **Sekantna metoda**

- ▶ Potrebuje dve začetni točki x_0, x_1 .
- ▶ Odvod nadomesti s sekanto.

Primerjava metod: Hitrost konvergence in robustnost

Hitrost konvergence

- ▶ **Bisekcija:** linearna, počasna, dolžina intervala se prepolovi v vsakem koraku.
- ▶ **Tangentna:** tipično kvadratna konvergenca (zelo hitra, če je x_0 blizu ničle).
- ▶ **Sekantna:** nad-linearna konvergenca (red $\approx 1,618$), hitrejša od bisekcije, nekoliko počasnejša od tangentne metode.

Robustnost

- ▶ **Bisekcija:** zelo robustna, konvergira, če je f zvezna in $f(a) \cdot f(b) < 0$.
- ▶ **Tangentna:** občutljiva na slab x_0 ali majhen $f'(x)$, lahko divergira.
- ▶ **Sekantna:** manj robustna kot bisekcija, bolj kot tangentna; možne težave, če je $f(x_k) - f(x_{k-1})$ zelo majhen.

Sistemi nelinearnih enačb

Rešujemo sistem nelinearnih enačb:

$$f_1(x_1, \dots, x_n) = 0,$$

$$f_2(x_1, \dots, x_n) = 0,$$

$$\vdots$$

$$f_n(x_1, \dots, x_n) = 0.$$

Če definiramo

$$\underline{f} := (f_1, \dots, f_n) : \mathbb{R}^n \rightarrow \mathbb{R}^n,$$

potem lahko sistem na kratko zapišemo kot

$$\underline{f}(\underline{x}) = 0.$$

Newtonova metoda: posplošitev tangentne metode,
Jacobijeva iteracija: posplošitev navadne iteracije.

Newtonova iteracija

Pri Newtonovi iteraciji tvorimo zaporedje približkov

$$\underline{x}^{(r+1)} = \underline{x}^{(r)} - J_{\underline{f}}(\underline{x}^{(r)})^{-1} \underline{f}(\underline{x}^{(r)}),$$

kjer je $J_{\underline{f}}(\underline{x}^{(r)})$ matrika prvih odvodov preslikave $\underline{f}$, ki ji pravimo **Jacobijeva matrika**:

$$J_{\underline{f}}(\underline{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} (\underline{x}).$$

V praksi pa ne računamo inverza $J_{\underline{f}}(\underline{x}^{(r)})^{-1}$, ampak namesto tega rešimo sistem

$$\begin{aligned} J_{\underline{f}}(\underline{x}^{(r)}) \Delta \underline{x}^{(r)} &= -\underline{f}(\underline{x}^{(r)}), \\ \underline{x}^{(r+1)} &= \underline{x}^{(r)} + \Delta \underline{x}^{(r)}. \end{aligned}$$

Izpeljava:

1. Funkcije f_i razvijemo v Taylorjevo vrsto:

$$f_i(\underline{x} + \Delta \underline{x}) = f_i(\underline{x}) + \sum_{k=1}^n \frac{\partial f_i}{\partial x_k}(\underline{x}) \Delta x_k + \dots, \quad i = 1, \dots, n.$$

2. Zanemarimo člene višjega reda in enačimo $f_i(\underline{x} + \Delta \underline{x}) = 0$.
3. Dobimo zgornji sistem.

Jacobijeva iteracija

1. Sistem $\underline{f}(\underline{x}) = 0$ preoblikujemo v ekvivalentno obliko

$$\underline{g}(\underline{x}) = \underline{x},$$

kjer je $\underline{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n$.

2. Izberemo začetni približek

$$\underline{x}^{(0)} \in \mathbb{R}^n.$$

3. Računamo zaporedje približkov

$$\underline{x}^{(r+1)} = \underline{g}(\underline{x}^{(r)}).$$

Izrek (Prvi konvergenčni izrek Jacobijeve iteracije)

Naj $\underline{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ na nekem območju $\Omega \subseteq \mathbb{R}^n$ zadošča:

1. $\underline{g}(\Omega) \subseteq \Omega$.
2. $\|\underline{g}(\underline{x}) - \underline{g}(\underline{y})\| \leq m \|\underline{x} - \underline{y}\|$ za vsaka $\underline{x}, \underline{y} \in \Omega$ in nek $0 \leq m < 1$.

Enačba

$$\underline{g}(\underline{x}) = \underline{x}$$

ima na območju Ω eno samo rešitev $\underline{\xi}$ in zaporedje $\underline{x}^{(r+1)}$ konvergira proti $\underline{\xi}$ za poljuben začetni približek $\underline{x}^{(0)} \in \Omega$. Velja še

$$\|\underline{x}^{(r+1)} - \underline{\xi}\| \leq \frac{m^r}{1-m} \|\underline{x}^{(1)} - \underline{x}^{(0)}\|.$$

Izrek (Drugi konvergenčni izrek Jacobijeve iteracije)

Naj $\underline{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ zvezno odvedljiva v negibni točki $\underline{\xi}$ in naj bo $\|\underline{J}_{\underline{g}}(\underline{\xi})\| < 1$.

Potem obstaja zaprta okolica $\Omega \subseteq \mathbb{R}^n$ fiksne točke $\underline{\xi}$, tako da zaporedje $\underline{x}^{(r+1)}$ konvergira proti $\underline{\xi}$ za poljuben začetni približek $\underline{x}^{(0)} \in \Omega$.

Čeprav obstajajo izreki o konvergenci Newtonove metode (denimo Kantorovičev), je njihove predpostavke v praksi težko preveriti. Konvergenco ponavadi zagotovi že dober začetni približek.

Kadar je sistem enačb velik, je z Newtonovo metodo veliko dela. Pohitrimo jo lahko tako, da Jacobijevo matriko na novo računamo samo na vsakih nekaj korakov. Taki metodi rečemo kvazi-Newtonova metoda.

Znane so tudi variante, ko Jacobijevo matriko ocenimo brez poznavanja parcialnih odvodov (Broydenova metoda).

Polinomska interpolacija in aproksimacija

* Poišči polinom p , da je
 $p(x_i) = y_i, \quad i = 0, 1, \dots, n.$

* Poišči polinom p stopnje k , da je
 $\sum_{i=0}^n \|f(x_i) - p(x_i)\|^2$ minimalno.

- ▶ Interpolacija v standardni bazi
- ▶ Interpolacija v Lagrangeovi bazi
- ▶ Interpolacija v Newtonovi bazi
- ▶ Polinomska aproksimacija, ortogonalni polinomi

Uvod v interpolacijo in aproksimacijo

Cilj: Aproksimirati želimo funkcijo $f(x)$ z lažjo funkcijo $g(x)$.

Tipi aproksimativnih funkcij: Polinomi, odsekoma polinomske funkcije, racionalne funkcije, trigonometrične funkcije, eksponentna funkcija, itd.

Vprašanje: Kako aproksimirati $f(x)$ z $g(x)$? V kakšnem smislu je aproksimacija dobra? Imamo več kriterijev:

1. **Interpolacija:** $g(x)$ mora imeti iste vrednosti kot $f(x)$ na dani množici točk.
2. **Metoda najmanjših kvadratov:** $g(x)$ se mora čim bolj prilegati $f(x)$ v smislu 2-norme, tj.

$$\int_a^b |f(t) - g(t)|^2 dt \quad \text{mora biti čim manjše.}$$

3. **Aproksimacija Čebiševa:** $g(x)$ se mora čim bolj prilegati $f(x)$ v smislu supremum norme, tj.

$$\text{minimizirati želimo} \quad \max_{t \in [a, b]} |f(t) - g(t)|.$$

Interpolacijski polinom v standardni bazi

Dani so naslednji podatki:

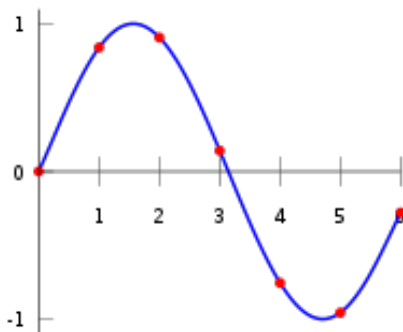
$n + 1$ točk $x_0, \dots, x_n$ in vrednosti $y_0, \dots, y_n$.

Iščemo polinom

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

stopnje n , ki zadošča

$$p(x_0) = y_0, \quad p(x_1) = y_1, \quad \dots, \quad p(x_n) = y_n. \quad (10)$$



Dobimo sistem

$$\begin{aligned}a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n &= y_0, \\a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n &= y_1, \\&\vdots \\a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n &= y_n.\end{aligned}\tag{11}$$

Polinomu $p(x)$ pravimo **interpolacijski polinom**.

V matrični obliki lahko sistem (11) zapišemo kot

$$Ax = b,$$

kjer je

$$A = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & & & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix}, \quad x = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \quad b = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}.$$

Matriki A pravimo **Vandermondova matrika** na točkah $x_0, \dots, x_n$. Velja

$$\det(A) = \prod_{0 \leq j < i \leq n} (x_i - x_j)$$

Posledica (O obstoju in enoličnosti interpolacijskega polinoma)

- ▶ Če so točke x_i , $i = 0, \dots, n$, paroma različne, ima sistem enolično rešitev.
- ▶ Polinom stopnje največ n skozi $n + 1$ točk je en sam.

Vprašanje:

1. Kako računsko zahtevno je reševanje sistema (11)?
2. Ali je sistem (11) numerično občutljiv?

Odgovor:

1. Računanje interpolacijskega polinoma s pomočjo Vandermondove matrike ni poceni ($\frac{2}{3}n^3 + \mathcal{O}(n^2)$ operacij).
2. Sistem je lahko že pri majhnem številu točk (npr. 10) zelo občutljiv za numerične napake.

Interpolacijski polinom: Lagrangeova in Newtonova baza

Namesto uporabe **standardne baze**

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

je bolje uporabiti eno od naslednjih baz:

► **Lagrangeova baza:**

$$\frac{(x-x_1)\cdots(x-x_n)}{(x_0-x_1)\cdots(x_0-x_n)}, \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)}, \dots, \frac{(x-x_0)\cdots(x-x_{n-1})}{(x_n-x_0)\cdots(x_n-x_{n-1})}.$$

► **Newtonova baza:**

$$1, x - x_0, (x - x_0)(x - x_1), \dots, (x - x_0) \cdots (x - x_{n-1}).$$

Obe zgornji bazi sta stabilni, Newtonova pa je cenejša za računanje v primeru dodajanja novih interpolacijskih točk.

Interpolacijski polinom v Lagrangeovi bazi

Primer

Poišči polinom najnižje stopnje, ki interpolira naslednji točki:

$$\begin{array}{c|cc} x & 1.4 & 1.25 \\ \hline y & 3.7 & 3.9 \end{array}.$$

Dobimo

$$p_1(x) = \left(\frac{x - 1.25}{1.4 - 1.25} \right) 3.7 + \left(\frac{x - 1.4}{1.25 - 1.4} \right) 3.9 = 3.7 - \frac{4}{3}(x - 1.4)$$

Kaj smo naredili? Zapisali smo $p(x)$ v obliki

$$p(x) = \underbrace{\left(\frac{x - x_1}{x_0 - x_1} \right)}_{\substack{\ell_0(x), \\ \ell_0(x_0)=1, \ell_0(x_1)=0}} y_0 + \underbrace{\left(\frac{x - x_0}{x_1 - x_0} \right)}_{\substack{\ell_1(x), \\ \ell_1(x_0)=0, \ell_1(x_1)=1}} y_1$$

Danih imamo $n + 1$ točk

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n).$$

Cilj je najti **Lagrangeove bazne polinome** stopnje največ n , ki zadoščajo

$$\ell_i(x_j) = \begin{cases} 0, & j \neq i, \\ 1, & j = i. \end{cases}$$

Torej je

$$\ell_i(x) = \underbrace{C_i}_{\text{konstanta}} \cdot \prod_{j \neq i} (x - x_j), \quad i = 0, \dots, n.$$

i -ti Lagrangeov bazni polinom je

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

Interpolacijski polinom v Lagrangeovi obliki je

$$p(x) = \sum_{i=0}^n \ell_i(x) y_i$$

Primer

Poišči enačbo parabole v Lagrangeovi obliki, ki gre skozi točke

$$(1, 6), (-1, 0), (2, 12).$$

$$\begin{aligned} \ell_0(x) &= \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x+1)(x-2)}{(2)(-1)} \\ \ell_1(x) &= \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-1)(x-2)}{(-2)(-3)} \\ \ell_2(x) &= \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x-1)(x+1)}{(1)(3)} \end{aligned}$$

$$\begin{aligned} p_2(x) &= y_0 \ell_0(x) + y_1 \ell_1(x) + y_2 \ell_2(x) \\ &= -3(x+1)(x-2) + 0(x-1)(x-2) + 4(x-1)(x+1). \end{aligned}$$

Interpolacijski polinom v Newtonovi bazi

Newtonov interpolacijski polinom na točkah $x_0, x_1, x_2, \dots, x_n$ je oblike

$$p_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + \dots \\ + c_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}).$$

Newtonovi bazni polinomi so

$$1, x - x_0, (x - x_0)(x - x_1), \dots, \prod_{i=0}^{n-1} (x - x_i).$$

Newtonova baza proti Lagrangeovi bazi:

Prednost Newtonove baze pred Lagrangeovo je v tem, da se z dodajanjem novih točk $x_{n+1}, \dots, x_{n+m}$ vsi že izračunani koeficienti $c_0, \dots, c_n$ ne spremenijo.

V primeru **zlepkov**, ko imamo v naprej določen n , so Lagrangeovi polinomi primernejši, saj imamo koeficiente že dane.

Interpolirajmo podatke (x_0, y_0) , (x_1, y_1) , (x_2, y_2) v **Newtonovi obliki**.

Poiskati moramo **koeficiente** c_0 , c_1 in c_2 v polinomu

$$p_2(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1).$$

Iz n podatkov dobimo sistem **n linearnih enačb** v neznanih koeficientih:

$$x_0 : \quad y_0 = c_0 + 0 + 0$$

$$x_1 : \quad y_1 = c_0 + c_1(x_1 - x_0) + 0$$

$$x_2 : \quad y_2 = c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1)$$

Ali v matrični obliki:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & x_1 - x_0 & 0 \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

Ker je matrika **spodnje trikotna**, potrebujemo samo $\mathcal{O}(n^2)$ operacij:

$$\begin{aligned}
 c_0 &= y_0 = f(x_0), \\
 c_1 &= \frac{y_1 - c_0}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}, \\
 c_2 &= \frac{y_2 - c_0 - (x_2 - x_0)c_1}{(x_2 - x_1)(x_2 - x_0)} \\
 &= \frac{f(x_2) - f(x_0) - (x_2 - x_0)\frac{f(x_1) - f(x_0)}{x_1 - x_0}}{(x_2 - x_1)(x_2 - x_0)} \\
 &= \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}.
 \end{aligned}$$

Deljena diferenca $f[x_0, \dots, x_k]$

Iz zgornjega primera opazimo naslednji vzorec. Pojavljajo se izrazi oblike:

$$\frac{f(x_j) - f(x_i)}{x_j - x_i}. \quad (12)$$

Če izraz (12) označimo z oglatimi oklepaji kot $f[x_i, x_j]$, potem bi na našem primeru dobili:

$$c_0 = f(x_0), \quad c_1 = f[x_0, x_1], \quad c_2 = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}.$$

To se da posplošiti do **rekurzivnega računanja** polinomov v Newtonovi obliki.

Deljena diferenca $f[x_0, \dots, x_k]$ je vodilni koeficient (pri x^k) interpolacijskega polinoma stopnje največ k , ki se z f ujema v točkah $x_0, \dots, x_k$.

Izrek (O koeficientih Newtonovega interpolacijskega polinoma)

1. *Koeficienti Newtonovega interpolacijskega polinom p_n stopnje največ n , ki se z f ujema v točkah $x_0, \dots, x_n$, so enaki*

$$c_i = f[x_0, x_1, \dots, x_i], \quad i = 0, \dots, n.$$

2. *Deljene difference povezuje formula*

$$f[x_0, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}.$$

Dokaz

Primer. Konstruirajmo deljene difference za podatke $(1, 3)$, $(\frac{3}{2}, \frac{13}{4})$, $(0, 3)$, $(2, \frac{5}{3})$.

Iz tabele deljenih diferenc preberimo interpolacijski polinom.

x	$f[\cdot]$	$f[\cdot, \cdot]$	$f[\cdot, \cdot, \cdot]$	$f[\cdot, \cdot, \cdot, \cdot]$
1	3			
$\frac{3}{2}$	$\frac{13}{4}$	$\frac{1}{2}$	$\frac{1}{3}$	
0	3	$\frac{1}{6}$	$-\frac{5}{3}$	-2
2	$\frac{5}{3}$	$-\frac{2}{3}$		

Interpolacijski polinom je tako

$$p_2(x) = 3 + \frac{1}{2}(x-1) + \frac{1}{3}(x-1)\left(x - \frac{3}{2}\right) - 2(x-1)\left(x - \frac{3}{2}\right)x.$$

Če uporabimo spodnjo stranico trikotnika, pa dobimo $p_2(x)$ izražen v drugi Newtonovi bazi:

$$p_2(x) = \frac{5}{3} - \frac{2}{3}(x-2) - \frac{5}{3}(x-2)x - 2(x-2)x\left(x - \frac{3}{2}\right).$$

Višanje stopnje aproksimacije

... ne izboljša vedno aproksimacije funkcije s polinomom.

Znan je **Rungejev primer**, ko funkcijo

$$f(x) = \frac{1}{1+x^2}$$

interpoliramo na intervalu $[-5, 5]$ z ekvidistantnimi točkami, tj.

$$x_0 = -5, x_1 = -5 + 10 \cdot \frac{1}{n}, \dots, x_{n-1} = -5 + 10 \cdot \frac{n-1}{n}, x_n = 5.$$

Pričakujemo, da se bo interpolacijski polinom **vse bolj prilegal** naši funkciji. Izkaže pa se, da temu ni tako. Če interpoliramo v točkah Čebiševa

$$x_i = 5 \cos \left(\frac{\pi}{2(i-1)(n+1)} \right), \quad i = 0, \dots, n$$

pa z višanjem stopnje res dobimo **boljše prileganje**.

Napaka polinomske interpolacije

Ponavadi nas zanima razlika med vrednostjo funkcije f in vrednostjo interpolacijskega polinoma p_n v neki točki t :

$$e_n(t) = p_n(t) - f(t).$$

Naj bo q_{n+1} interpolacijski polinom funkcije f skozi točke $x_0, \dots, x_n$ in t :

$$q_{n+1}(x) = p_n(x) + f[x_0, x_1, \dots, x_n, t] \cdot \prod_{i=0}^n (x - x_i).$$

Iz enakosti $f(t) = q_{n+1}(t)$, sledi

$$e_n(t) = p_n(t) - f(t) = -f[x_0, x_1, \dots, x_n, t] \prod_{i=0}^n (t - x_i).$$

Za oceno napako moramo oceniti še vrednost $f[x_0, x_1, \dots, x_n, t]$.

Izrek (O deljenih diferencah)

$$f[x_0, x_1, \dots, x_n, t] = \frac{f^{(n+1)}(\xi)}{(n+1)!} \quad \text{za nek } \xi \in [a, b].$$

Dokaz

Izrek (Napaka polinomske interpolacije)

Naj bo f vsaj $(n+1)$ -krat zvezno odvedljiva na intervalu $[a, b]$ in naj bo p_n interpolacijski polinom stopnje največ n skozi točke x_i , $i = 0, \dots, n$, ki vse ležijo na intervalu $[a, b]$. Potem je za vsak $x \in [a, b]$

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0) \cdots (x - x_n),$$

kjer ξ leži na intervalu $[a, b]$.

Če znamo odvod $f^{(n+1)}$ na intervalu, ki nas zanima, omejiti, lahko dobimo uporabno oceno.

Aproksimacija po metodi najmanjših kvadratov

Za funkcijo, podano v n točkah

$$(x_0, y_0), \dots, (x_n, y_n),$$

iščemo polinom p_k stopnje $k \leq n$, za katerega ima izraz

$$E_{LSQ} = \sqrt{\sum_{i=0}^n (p_k(x_i) - y_i)^2}$$

najmanjšo vrednost. Če zapišemo na dolgo:

$$E_{LSQ} = \sqrt{\sum_{i=0}^n \underbrace{(a_0 + a_1 x_i + \dots + a_k x_i^k)}_{p_k(x_i)} - y_i)^2}.$$

Torej iščemo ekstrem funkcije več spremenljivk. Iz analize vemo, da je potreben pogoj za ekstrem

$$\frac{\partial E_{LSQ}}{\partial a_0} = \frac{\partial E_{LSQ}}{\partial a_1} = \dots = \frac{\partial E_{LSQ}}{\partial a_k} = 0.$$

Naj bo

$$s_1 = x_0 + \dots + x_n, \quad s_2 = x_0^2 + \dots + x_n^2, \quad \dots, \quad s_{2k} = x_0^{2k} + \dots + x_n^{2k}.$$

Dobimo **normalni sistem**:

$$\begin{bmatrix} n & s_1 & s_2 & \dots & s_k \\ s_1 & s_2 & s_3 & \dots & s_{k+1} \\ s_2 & s_3 & s_4 & \dots & s_{k+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s_k & s_{k+1} & s_{k+2} & \dots & s_{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n y_i x_i \\ \sum_{i=0}^n y_i x_i^2 \\ \vdots \\ \sum_{i=0}^n y_i x_i^k \end{bmatrix},$$

ki pa je pri velikem številu točk lahko **numerično slabo pogojen**.

Aproksimacija po metodi najmanjših kvadratov

Do ekvivalentnega sistema bi prišli tudi z zapisamo prvotnega sistema v obliki $Ax = b$ in mu priredili **normalni sistem** $A^T Ax = A^T b$:

$$\begin{aligned}a_0 + a_1 \sum_{i=1}^m x_i + \dots + a_n \sum_{i=1}^m x_i^n &= \sum_{i=1}^m f(x_i), \\a_0 \sum_{i=1}^m x_i + a_1 \sum_{i=1}^m x_i^2 + \dots + a_n \sum_{i=1}^m x_i^{n+1} &= \sum_{i=1}^m f(x_i)x_i, \\a_0 \sum_{i=1}^m x_i^2 + a_1 \sum_{i=1}^m x_i^3 + \dots + a_n \sum_{i=1}^m x_i^{n+2} &= \sum_{i=1}^m f(x_i)x_i^2, \\&\vdots \\a_0 \sum_{i=1}^m x_i^n + a_1 \sum_{i=1}^m x_i^{n+1} + \dots + a_k \sum_{i=1}^m x_i^{2n} &= \sum_{i=1}^m f(x_i)x_i^n.\end{aligned}$$

Ta sistem lahko rešimo z Gaussovo eliminacijo, vendar je lahko pri veliko točkah **slabo pogojen**. Cilj je problem preoblikovati v ekvivalentnega, vendar **bolje pogojenega**. Rešitev leži v uporabi baze **ortogonalnih polinomov**.

Zamenjava baze prostora polinomov

Če bi namesto baze $\{1, x, \dots, x^n\}$ vzeli novo bazo polinomov

$$\{g_0(x), g_1(x), \dots, g_n(x)\},$$

bi dobili sistem linearnih enačb:

$$a_0 \sum_{i=1}^m g_0^2(x_i) + a_1 \sum_{i=1}^m g_0(x_i)g_1(x_i) + \dots + a_n \sum_{i=1}^m g_0(x_i)g_n(x_i) = \sum_{i=1}^m f(x_i)g_0(x_i),$$

$$a_0 \sum_{i=1}^m g_1(x_i)g_0(x_i) + a_1 \sum_{i=1}^m g_1^2(x_i) + \dots + a_n \sum_{i=1}^m g_1(x_i)g_n(x_i) = \sum_{i=1}^m f(x_i)g_1(x_i),$$

$$a_0 \sum_{i=1}^m g_2(x_i)g_0(x_i) + a_1 \sum_{i=1}^m g_2(x_i)g_1(x_i) + \dots + a_n \sum_{i=1}^m g_2(x_i)g_n(x_i) = \sum_{i=1}^m f(x_i)g_2(x_i),$$

$\vdots$

$$a_0 \sum_{i=1}^m g_n(x_i)g_0(x_i) + a_1 \sum_{i=1}^m g_n(x_i)g_1(x_i) + \dots + a_n \sum_{i=1}^m g_n^2(x_i) = \sum_{i=1}^m f(x_i)g_n(x_i).$$

Želeli bi izbrati bazo, v kateri bo ta **sistem diagonalen**. Iščemo torej polinome g_j , za katere velja

$$\sum_{i=1}^m g_j(x_i)g_k(x_i) = 0 \quad \text{za } j \neq k.$$

Zgornji sistem lahko zapišemo tudi v **matrični obliki**:

$$A = \begin{pmatrix} \langle g_0, g_0 \rangle & \langle g_0, g_1 \rangle & \dots & \langle g_0, g_n \rangle \\ \langle g_1, g_0 \rangle & \langle g_1, g_1 \rangle & \dots & \langle g_1, g_n \rangle \\ \vdots & & & \vdots \\ \langle g_n, g_0 \rangle & \langle g_n, g_1 \rangle & \dots & \langle g_n, g_n \rangle \end{pmatrix}, \quad x = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix}, \quad b = \begin{pmatrix} \langle f, g_0 \rangle \\ \langle f, g_1 \rangle \\ \vdots \\ \langle f, g_n \rangle \end{pmatrix},$$

kjer je $\langle f, g \rangle := \sum_{i=1}^m f(x_i)g(x_i)$. Iščemo torej polinome g_j , za katere velja

$$\langle g_j, g_k \rangle = 0 \quad \text{za } j \neq k. \quad (13)$$

Zaporedju polinomov $g_0, \dots, g_n$, za katere velja $\deg g_j = j$, vodilni koeficient g_j je 1 in velja (13), pravimo **zaporedje ortogonalnih polinomov (ZOP)**.

Rešitev našega problema je ZOP g_j polinom p :

$$p(x) = \sum_{i=0}^n a_i g_i(x), \quad a_i = \frac{\langle f, g_i \rangle}{\langle g_i, g_i \rangle}.$$

Numerična integracija

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

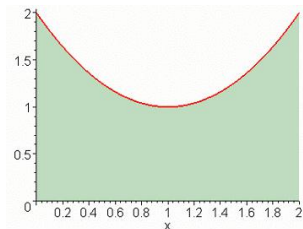
- ▶ Newton–Cotesova (NC) pravila: trapezno, Simpsonovo pravilo
- ▶ Izbira koraka v NC pravilih
- ▶ Adaptivna NC pravila
- ▶ Gaussove kvadraturene formule

Numerična integracija

Naš cilj je izračunati določen integral

$$\int_a^b f(x) dx = F(b) - F(a)$$

funkcije $f(x)$. Tu je F nedoločen integral funkcije f .

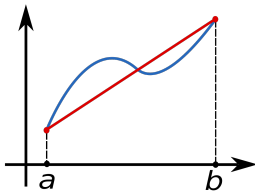


Če ne znamo izračunati nedoločenega integrala F , smo v težavah. Npr. za $f(x) = e^{-x^2}$, $g(x) = \frac{\sin x}{x}$, $h(x) = x \tan x$.

Prav tako ne moremo točno izračunati vrednosti integrala, če imamo funkcijo podano samo na neki množici točk.

Osnovno trapezno pravilo in napaka E

Integral $\int_a^{a+h} f(x) dx$ tako, da f aproksimiramo z linearno funkcijo in izračunamo ploščino pod linearno funkcijo oz. trapezom.



$$p(x) = f(a) + \frac{f(b) - f(a)}{b - a}(x - a)$$

Velja

$$\begin{aligned} \int_a^b f(x) dx &\approx \int_a^b p(x) dx = f(a)(b - a) + \frac{f(b) - f(a)}{b - a} \frac{(b - a)^2}{2} \\ &= \frac{(b - a)}{2} (f(a) + f(b)). \end{aligned}$$

Pri tem je napaka naslednja:

$$\begin{aligned} E &= \int_a^b (f(x) - p(x)) dx = \int_a^b f[a, b, x](x - b)(x - a) dx \\ &= \frac{f''(\eta)}{2} \cdot \int_a^b (x - b)(x - a) dx \\ &= \frac{f''(\eta)}{2} \left(-\frac{1}{6}(b - a)^3 \right) \\ &= \boxed{-\frac{(b - a)^3 f''(\eta)}{12}}, \end{aligned}$$

kjer je $\eta \in [a, b]$ in tretja enakost sledi po izreku o $f[a, b, \xi]$.

Sestavljeno trapezno pravilo

Če interval ni zelo kratek, potem očitna naivna linearna transformacija običajno ne da dobrega približka integrala.

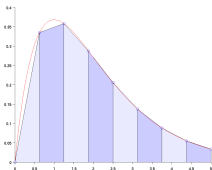
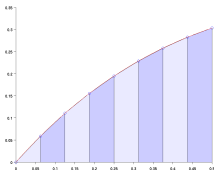
Če interval $[a, b]$ razdelimo z ekvidistantnimi točkami $x_0, x_1, \dots, x_n$, tj.

$$h := h_i = x_{i+1} - x_i$$

je konstanta in na vsakem intervalu uporabimo osnovno trapezno pravilo, dobimo:

$$\int_a^b f(x) dx \approx \frac{h}{2} \sum_{i=0}^{n-1} f(x_i) + f(x_{i+1})$$

$$= \frac{h}{2} (f(x_0) + 2f(x_1) + 2f(x_2) + \dots + 2f(x_{n-1}) + f(x_n))$$



Napaka E_i na intervalu $[x_i, x_{i+1}]$ je enaka

$$E_i = -\frac{h^3 \cdot f''(\eta_i)}{12} \quad \text{za nek } \eta_i \in [x_i, x_{i+1}].$$

Torej je skupna napaka

$$\begin{aligned} E &= \sum_{i=0}^{n-1} E_i = \sum_{i=0}^{n-1} -\frac{h^3 \cdot f''(\eta_i)}{12} = -\frac{h^3 \cdot n f''(\eta)}{12} \\ &= \boxed{-\frac{(b-a)h^2 \cdot f''(\eta)}{12}}, \end{aligned}$$

kjer je $\eta \in [a, b]$ in smo v tretji enakosti uporabili izrek o srednji vrednosti (f'' je zvezna) in v zadnji enakosti dejstvo, da je $b - a = nh$.

Primer - $\int_0^1 e^{-x^2} dx$

Koliko točk uporabiti, da bo sestavljeno trapezno pravilno natančno z napako omejeno z 10^{-6} ?

Želimo

$$\left| \frac{(b-a)h^2 f''(\eta)}{12} \right| \leq 10^{-6}$$

Kako velik je drugi odvod $f''(x)$?

$$f'(x) = -2xe^{-x^2}, \quad f''(x) = -2e^{-x^2} + 4x^2e^{-x^2}.$$

Ker je

$$f'''(x) = 12xe^{-x^2} - 8x^3e^{-x^2} = 4x(3 - 2x^2)e^{-x^2}$$

pozitiven na $[0, 1]$, je f'' monoton naraščajoč na $[0, 1]$ in zato $|f''|$ zavzame maksimum v krajišču: $|f''(0)| = 2$. Potem lahko omejimo

$$\frac{(b-a)2h^2}{12} \leq 10^{-6} \quad \Rightarrow \quad h^2 \leq 6 \cdot 10^{-6} \quad \Rightarrow \quad \underbrace{\sqrt{(1/6)10^3}}_{\approx 410} \leq n.$$

Trapezno pravilo s kontrolo koraka

Motivacija. Če uporabimo sestavljeno trapezno pravilo, moramo:

- ▶ Vnaprej določiti velikost h .
- ▶ Če želimo oceniti napako, moramo znati oceniti $f''(\eta)$ na intervalu $[a, b]$.

Obe težavi želimo rešiti, tj. radi bi, da funkcija samo zmanjšuje h , v kolikor napaka ni dovolj manjka. V ta namen moramo znati to napako oceniti.

Pridemo do **trapeznega pravila s kontrolo koraka**.

Naj bo $I = \int_a^b f(x)dx$ in $T(h)$ ocena za I z uporabo sestavljenega trapeznega pravila z velikostjo intervala h .

Spomnimo se, da pri sestavljenem trapeznem pravilu $T(h)$ za napako $E(h)$ velja:

$$E(h) := T(h) - I = \frac{b-a}{12} f''(\xi_h) h^2, \quad \text{kjer je } \xi_h \in (a, b).$$

Želimo se izogniti dejstvu, da moramo poznati f'' . Zapišimo napako še v primeru razpolovljenega koraka, tj. $\frac{h}{2}$:

$$E(h/2) := T(h/2) - I = \frac{b-a}{12} f''(\xi_{h/2}) \frac{h^2}{4}, \quad \text{kjer je } \xi_{h/2} \in (a, b).$$

Predpostavimo, da je $\frac{b-a}{12} f''(\xi_h)$ približno **enako C** za vsak h .

Dobimo:

$$I = T(h) - Ch^2 = T(h/2) - C\frac{h^2}{4}.$$

Sledi:

$$T(h) - T(h/2) = \frac{3}{4}Ch^2 + \mathcal{O}(h^4) \quad \text{oz.} \quad Ch^2 \approx \frac{4}{3}(T(h) - T(h/2)).$$

Tako sta

$$\frac{4}{3}(T(h) - T(h/2)), \quad \frac{1}{3}(T(h) - T(h/2))$$

približka za napaki $E(h)$ in $E(h/2)$. Velja

$$T(h/2) = \underbrace{\frac{T(h)}{2}}_{\text{razpolovimo } T(h)} + \underbrace{\frac{h}{2} \sum_{i=1}^n f(a + (i-1/2)h)}_{\text{računamo samo ta del}}, \quad n = (b-a)/h.$$

Algoritem:

1. Izračunamo $T(b-a) = (b-a) \frac{f(a)+f(b)}{2}$.
2. Izračunamo $T((b-a)/2) = \frac{T(b-a)}{2} + \frac{b-a}{2} f((a+b)/2)$.
3. Izračunamo $\frac{1}{3}(T(b-a) - T((b-a)/2))$. Če je to dovolj majhno po absolutni vrednosti, končamo, približek za integral pa je $T((b-a)/2)$. Sicer ponovimo postopek z razpolovljenim h .

Adaptivno trapezno pravilo

Motivacija: Če uporabimo trapezno pravilo s kontrolo koraka, potem dolžine koraka h ne rabimo sami določiti, vendar pa je h enak na celotnem integracijskem intervalu. **Želeli bi, da na nekaterih delih intervala uporabimo večje h , manjše pa le tam, kjer je to res potrebno.**

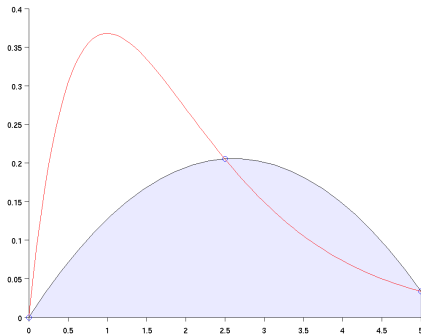
Zgornji cilj lahko dosežemo z uporabo **rekurzivnega računanja integrala**:

- ▶ Najprej izračunamo $T(b - a)$ in $T((b - a)/2)$.
- ▶ Če je podobno kot pri kontroli koraka zgoraj ocena napake $e := \frac{T(b-a)/2 - T(b-a)}{3}$ dovolj majhna, vrnemo $T((b - a)/2) + e$ in končamo.
- ▶ Če je e prevelik, ponovimo zgornji postopek ločeno za podintervala $[a, (a + b)/2]$ in $[(a + b)/2, b]$, pri čemer naj bo napaka na vsakem največ **polovica začetne tolerance**.
- ▶ **Rekurzivno nadaljujemo** zgornji postopek in dobimo oceno integrala, pri čemer delilne točke ne bodo enakomerno razporejene po intervalu $[a, b]$.

Enostavno Simpsonovo pravilo

Naj bo p_2 polinom stopnje 2, s katerim interpoliramo točke

$$(a, f(a)), \quad \left(\frac{a+b}{2}, f\left(\frac{a+b}{2}\right)\right), \quad (b, f(b)) :$$



$$p_2(x) = C_0 + C_1 \cdot (x - a) + C_2 \cdot (x - a) \left(x - \frac{a+b}{2} \right).$$

Označimo $h := \frac{b-a}{2}$. Rešujemo sistem:

$$p_2(a) = f(a), \quad p_2\left(\frac{a+b}{2}\right) = f\left(\frac{a+b}{2}\right), \quad p_2(b) = f(b).$$

Dobimo

$$C_0 = f[a] = f(a), \quad C_1 = f\left[a, \frac{a+b}{2}\right] = \frac{f(a+h) - f(a)}{h},$$

$$C_2 = f\left[a, \frac{a+b}{2}, b\right] = \frac{f(a+2h) - 2f(a+h) + f(a)}{2h^2}.$$

Računamo $\int_a^b p_2(x) dx$ (naredimo substitucijo $x = a + t$);

$$\begin{aligned} \int_a^{a+2h} p_2(x) dx &= \int_0^{2h} p_2(a+t) dt \\ &= f(a) \cdot 2h + \frac{f(a+h) - f(a)}{h} \cdot 2h^2 + \\ &+ \frac{f(a+2h) - 2f(a+h) + f(a)}{2h^2} \cdot \frac{2}{3}h^3 = \\ &\boxed{\frac{h}{3}(f(a) + 4f(a+h) + f(a+2h))}. \end{aligned}$$

V prejšnjem izračunu smo upoštevali

$$p_2(a+t) = C_0 + C_1 t + C_2 t(t-h).$$

Izkaže se, da je napaka približno:

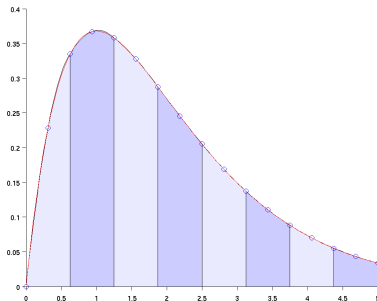
$$\boxed{-\frac{1}{90} h^5 f^{(4)}(\xi)}, \quad \xi \in [a, b].$$

Sestavljeno Simpsonovo pravilo in napaka

Vzemimo ekvidistantno particijo $P = \{x_0 = a < \dots < x_n = b\}$ intervala $[a, b]$ na sodo število enako dolgih intervalov in na zaporednih trojicah točk uporabimo osnovno Simpsonovo pravilo ($h = x_{i+1} - x_i$):

$$\int_a^b f(x) dx \approx \sum_{i=0}^{\frac{n}{2}-1} \frac{h}{3} [f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})]$$

$$= \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \dots + f(x_n)].$$



Napaka E_i na intervalu $[x_{2i}, x_{2i+2}]$ je enaka

$$E_i = -\frac{h^5 f^{(4)}(\eta_i)}{90}$$

za nek $\eta_i \in [x_{2i}, x_{2i+2}]$. Torej je skupna napaka

$$\begin{aligned} E &= \sum_{i=0}^{\frac{n}{2}-1} E_i = \sum_{i=0}^{\frac{n}{2}-1} -\frac{h^5 f^{(4)}(\eta_i)}{90} = -\frac{n}{2} \frac{h^5 f^{(4)}(\eta)}{90} \\ &= \boxed{-\frac{(b-a)h^4 f''(\eta)}{180}}, \end{aligned}$$

kjer je $\eta \in [a, b]$ in smo v tretji enakosti uporabili izrek o srednji vrednosti.

Adaptivno Simpsonovo pravilo

Motivacija. Ideja je povsem enaka kot pri adaptivnem trapeznem pravilu, tj. radi bi uporabili čim večji h povsod, kjer je to mogoče. Če s $S(h)$ označimo vrednost sestavljenega Simpsonovega pravila s korakom dolžine h , potem napako E ocenimo iz $S(h)$ in $S(h/2)$.

Postopek:

- ▶ Najprej izračunamo $S(b-a)$ in $S((b-a)/2)$.
- ▶ Iz $\int_a^b f(x)dx = S(h) + C_1 h^4 = S(h/2) + C_1 (\frac{h}{2})^4$ izrazimo

$$C_1 \left(\frac{h}{2}\right)^4 = \frac{S(b-a)/2 - S(b-a)}{15},$$

kar je naša ocena napake E . Če je E dovolj majhna, vrnemo $S((b-a)/2) + E$ in končamo.

- ▶ Če je E prevelik, ponovimo zgornji postopek ločeno za podintervala $[a, (a+b)/2]$ in $[(a+b)/2, b]$, pri čemer naj bo napaka na vsakem največ polovica začetne tolerance.
- ▶ **Rekurzivno nadaljujemo** zgornji postopek in dobimo oceno integrala, pri čemer delilne točke ne bodo enakomerno razporejene po intervalu $[a, b]$.

Gaussove kvadraturene formule

- ▶ NC pravila za integriranje so oblike

$$\int_a^b f(x) dx \approx \sum_{j=0}^n w_j f(x_j), \quad (14)$$

kjer so točke x_j enakomerno razporejene, **vozli** w_j pa **uteži**.

- ▶ Vemo pa že iz poglavja o interpolacijskih polinomih, da **ekvidistantne točke niso vedno najboljša izbira**.
- ▶ Rešili se bomo ekvidistantnih vozlov v kvadraturenih formulah.
- ▶ V formuli (14) bomo **izbirali vozle in koeficiente na optimalen način**, tako da **maksimiziramo stopnjo natančnosti**, tj. integracijsko pravilo bo točno za polinome najvišjih možnih stopenj.
- ▶ Imamo **$n + 1$ prostih točk** $x_j \in [a, b]$,

$$a \leq x_0 < x_1 < \cdots < x_{n-1} < x_n \leq b.$$

in **$n + 1$ realnih koeficientov** w_j , tj. skupaj **$2n + 2$ neznank**.

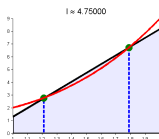
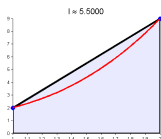
Primer najboljših vozlov za interval $[-1, 1]$

Oglejmo si primer $n = 1$ (tj. 2 točki) na primeru intervala $[-1, 1]$. Poiščimo w_0 , w_1 , x_0 , x_1 , tako da velja

$$\int_{-1}^1 f(x) dx \approx w_0 f(x_0) + w_1 f(x_1),$$

pri čemer je aproksimacija kar se da točna.

$$\int_1^2 (x^3 + 1) dx = \left[\frac{x^4}{4} + x \right]_1^2 = 4.75.$$



Cilj: poišči w_0 , w_1 , x_0 , x_1 tako da bi aproksimacija točna za polinome stopnje največ 3:

$$f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3.$$

To pomeni, da mora za vsak $c_0, c_1, c_2, c_3 \in \mathbb{R}$ veljati:

$$\begin{aligned} \int_{-1}^1 f(x) dx &= \int_{-1}^1 (c_0 + c_1 x + c_2 x^2 + c_3 x^3) dx \\ &= w_0 (c_0 + c_1 x_0 + c_2 x_0^2 + c_3 x_0^3) + w_1 (c_0 + c_1 x_1 + c_2 x_1^2 + c_3 x_1^3). \end{aligned}$$

Desno stran preuredimo na **konstantne, linearne, kvadratične in kubične člene**, ter dobimo, da je naslednji izraz

$$c_0 \left(w_0 + w_1 - \int_{-1}^1 1 \, dx \right) + c_1 \left(w_0 x_0 + w_1 x_1 - \int_{-1}^1 x \, dx \right) \\ + c_2 \left(w_0 x_0^2 + w_1 x_1^2 - \int_{-1}^1 x^2 \, dx \right) + c_3 \left(w_0 x_0^3 + w_1 x_1^3 - \int_{-1}^1 x^3 \, dx \right).$$

ničelen. Ker so koeficienti c_0, c_1, c_2 in c_3 poljubni, morajo biti koeficienti pri njih ničelni.

Od tod sledi:

$$\begin{aligned} w_0 + w_1 &= \int_{-1}^1 1 \, dx = 2 & w_0 x_0 + w_1 x_1 &= \int_{-1}^1 x \, dx = 0 \\ w_0 x_0^2 + w_1 x_1^2 &= \int_{-1}^1 x^2 \, dx = \frac{2}{3} & w_0 x_0^3 + w_1 x_1^3 &= \int_{-1}^1 x^3 \, dx = 0 \end{aligned}$$

Z nekaj algebre pridemo do:

$$w_0 = 1 \quad w_1 = 1 \quad x_0 = -\frac{\sqrt{3}}{3} \quad x_1 = \frac{\sqrt{3}}{3}$$

Zato:

$$\int_{-1}^1 f(x) \, dx \approx f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right).$$

Posplošitev na interval $[a, b]$

Z linearno substitucijo

$$t = a_0 + a_1 x, \quad t(a) = -1, \quad t(b) = 1,$$

preslikamo interval $[a, b]$ na $[-1, 1]$.

Velja $a_0 = -\frac{b+a}{b-a}$ in $a_1 = \frac{2}{b-a}$ ter

$$x = \frac{b-a}{2}t + \frac{b+a}{2}, \quad dx = \frac{b-a}{2}dt.$$

Sledi:

$$\int_a^b f(x) dx = \int_{-1}^1 f\left(\frac{(b-a)t + b+a}{2}\right) \frac{b-a}{2} dt$$

in lahko uporabimo kvadraturno formulo nad $[-1, 1]$.

Z uporabo dveh točk, $n = 1$, smo dobili točen integral za polinome stopnje največ $2 \cdot 1 + 1 = 3$.

Razširitev Gaussovih kvadraturnih formul

Sedaj je naš cilj **razširiti zgornje pravilo tako, da bo delovalo za polinome višje stopnje**, tj. z vsaki dodanim parom vozla in uteži želimo povečati točnost za dve stopnji.

Velja:

- ▶ Smiselno kvadraturno pravilo za integracijo nad intervalom $[-1, 1]$ na enem vozlu bi uporabilo $x = 0$. To pa je ničla funkcije $\phi(x) = x$.
- ▶ Kvadratura pravilo na dveh točkah $\pm \frac{1}{\sqrt{3}}$ smo dobili za ničli funkcije

$$\phi(x) = 3x^2 - 1.$$

- ▶ Kako nadaljevati?

Izrek (Gauss)

Naj bo $q(x)$ netrivialen polinom stopnje $n + 1$, tako da je

$$\int_a^b x^k q(x) dx = 0 \quad \text{za vsak } k = 0, 1, \dots, n$$

in naj bodo $x_0, x_1, \dots, x_n$ ničle funkcije $q(x)$. Potem velja

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

kjer je

$$A_i = \int_a^b \ell_i(x) dx \quad \text{za } i = 0, \dots, n,$$

pri čemer ℓ_i označuje i -ti Lagrangeov bazni polinom na točkah $x_0, \dots, x_n$, pravilo pa je točno za polinome stopnje največ $2n + 1$.

Reševanje diferencialnih enačb

$$y' = f(x, y), \quad y(x_0) = y_0$$

- ▶ Eulerjeva metoda
- ▶ Runge-Kutta metode

Diferencialna enačba

Diferencialna enačba (DE) je enačba oblike:

$$F(t, x, \dot{x}, \ddot{x}, \dots, x^{(n)}) = 0, \quad (16)$$

kjer je $x = x(t)$ **odvisna** spremenljivka, t **neodvisna** spremenljivka, $\dot{x}$ pa označuje odvod x po t .

Če je $y = y(x)$ **odvisna** spremenljivka, x pa **neodvisna**, potem je DE oblike

$$F(x, y', y'', \dots, y^{(n)}) = 0. \quad (17)$$

Ključna lastnost DE je ta, da poleg neodvisne spremenljivke t (oz. x) in odvisne spremenljivke x (oz. y) nastopajo še **odvodi** odvisne spremenljivke $\dot{x}, \dots, x^{(n)}$ (oz. $y', \dots, y^{(n)}$).

Rešitev DE je (dovoljkrat odvedljiva) funkcija, ki zadošča enačbi (16) oz. (17) na definicijskem območju $\mathcal{D}$ neodvisne spremenljivke.

Red DE je stopnja najvišjega odvoda, ki nastopa v DE.

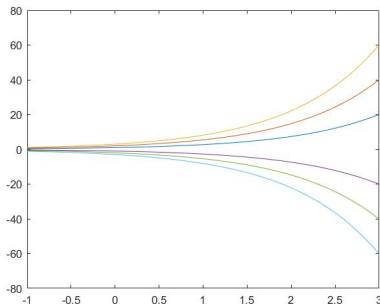
Primeri. $y' = y$, $y' + 5xy = 3x^2$, $\dot{x} + x = 0$, $\ddot{x} + a\dot{x} + bx = A \cos \omega t$.

Splošna rešitev diferencialne enačbe reda n je družina funkcij, odvisna od n parametrov, ki so vse rešitve diferencialne enačbe.

Primer

Rešimo DE $y' = y$.

$$\begin{aligned}\frac{dy}{dx} = y &\Rightarrow \frac{dy}{y} = dx \Rightarrow \int \frac{dy}{y} = \int dx \\ &\Rightarrow \log(|y|) = x + C, \quad C \in \mathbb{R} \\ &\Rightarrow y = Ke^x, \quad K \in \mathbb{R}\end{aligned}$$



Partikularna rešitev je posamezna rešitev iz te družine.

Določena je z n dodatnimi pogoji, na primer z začetnimi pogoji:

$$x(t_0) = a_0, \quad \dot{x}(t_0) = a_1, \dots, \quad x^{(n-1)}(t_0) = a_{n-1}$$

Zelo malo DE je analitično rešljivih. Mednje sodijo:

- ▶ DE z ločljivima spremenljivkama
- ▶ Linearne DE
- ▶ DE zelo posebne oblike

Večina DE ni analitično rešljivih. Te rešujemo numerično.

Diferencialna enačba 1. reda z ločljivima spremenljivkama

$$\dot{x} = f(t)g(x)$$

Enačbo rešimo tako, da vpeljemo $\dot{x} = \frac{dx}{dt}$ in ločimo spremenljivki:

$$\frac{dx}{dt} = f(t)g(x), \quad \frac{dx}{g(x)} = f(t)dt$$

in potem integriramo

$$\int \frac{dx}{g(x)} = \int f(t)dt.$$

Linearna diferencialna enačba

$$y' + f(x)y = g(x) \quad (18)$$

Pravimo, da je enačba **homogena**, če je $g(x) = 0$ in **nehomogena**, če je $g(x) \neq 0$.

1. Rešimo **homogeni del** $y' + f(x)y = 0$ s pomočjo ločitve spremenljivk. Dobimo rešitev

$$y = Ce^{-\int f(x)dx} = C z(x)$$

2. Metoda **variacije konstante**

- ▶ V (18) vstavimo $y = C(x) z(x)$ in rešimo na $C(x)$.
- ▶ Tako dobljeni C vstavimo v rešitev homogenega dela.

Numerično reševanje DE

Na intervalu $[a, b]$ rešujemo DE prvega reda

$$y' = f(x, y), \quad y(a) = y_0. \quad (19)$$

Interval $[a, b]$ razdelimo z zaporedjem točk

$$a = x_0 < x_1 < x_2 < \dots < x_n = b.$$

Z y_i označimo **približek** za rešitev (19) v točki x_i . Označimo **dolžino koraka** z $h_i := x_{i+1} - x_i$.

Razliko med približkom in točno rešitvijo v x_i pišemo z $g_i = y_i - y(x_i)$ in jo imenujemo **globalna napaka** v x_i .

Razliko med približkom in točno rešitvijo DE

$$z' = f(x, z), \quad z(x_{i-1}) = y_{i-1} \quad (20)$$

v x_i pišemo z $\ell_i = y_i - z(x_i)$ in jo imenujemo **lokalna napaka** v x_i .

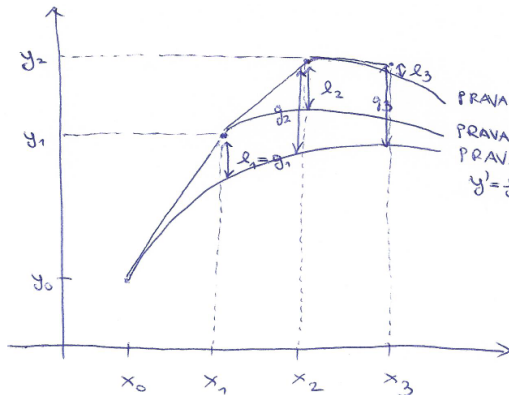
Red metode je število $p \in \mathbb{N}$, ki zadošča

$$\ell_i = Ch_i^{p+1} + \mathcal{O}(h_n^{p+2})$$

Eulerjeva metoda

Pri tej metodi v vsaki točki x_i uporabimo **linearno aproksimacijo funkcije**. Rešitev na intervalu $[x_i, x_{i+1}]$ nadomestimo z odsekom tangente na graf rešitve v točki x_i :

$$y_{i+1} = y_i + h_i \cdot f(x_i, y_i).$$



PRAVA REŠITEV $y' = f(x, y)$, $y(x_2) = y_2$

PRAVA REŠITEV $y' = f(x, y)$, $y(x_1) = y_1$

PRAVA REŠITEV
 $y' = f(x, y)$, $y(x_0) = y_0$

```

1
2   $y = y_0$ 
3   $x = x_0$ 
4   $h = (b - a)/n$ 
5  for  $i = 1, \dots, n - 1$ 
6       $y = y + h \cdot f(x, y)$ 
7       $x = x + h$ 
8  end

```

Ker je

$$y(x + h) = \underbrace{y(x) + hy'(x)}_{\text{upoštevamo}} + \underbrace{\frac{h^2}{2}y''(\xi)}_{\text{napaka}}, \quad \xi \in [x, x + h],$$

je red Eulerjeve metode 1.

Metode Runge-Kutta

Ideja teh metod je, da za aproksimacijo odvoda na intervalu $[x_n, x_{n+1}]$ ne upoštevamo odvoda le v točki x_n , temveč neko uteženo povprečje odvodov na $[x_n, x_{n+1}]$.

Primer (Metode Runge-Kutta (RK) reda 2)

Upoštevamo odvoda v točki x_n in $x_n + ch \in [x_n, x_{n+1}]$, kjer je $h = x_{n+1} - x_n$ in $c \in [0, 1]$. Približek y_{n+1} izračunamo tako, da se premaknemo za uteženo povprečje premikov po tangentah v točkah x_n in $x_n + ch$:

$$y_{n+1} = y_n + \underbrace{b_1}_{\text{utež}} \cdot \underbrace{(h \cdot f(x_n, y_n))}_{\text{tangenta v } x_n} + \underbrace{b_2}_{\text{utež}} \cdot \underbrace{(h \cdot f(x_n + ch, y(x_n + ch)))}_{\text{tangenta v } x_n + ch} \quad (21)$$

Upoštevamo

$$y(x_n + ch) \approx y_n + chy'(x_n) = y_n + chf(x_n, y_n) \approx y_n + ahf(x_n, y_n), \quad (22)$$

kjer je a postal prost parameter.

Primer (Metode Runge-Kutta (RK) reda 2)

Upoštevamo (22) v (21) in dobimo

$$y_{n+1} = y_n + b_1 \cdot \underbrace{(h \cdot f(x_n, y_n))}_{k_1} + b_2 \cdot \underbrace{(h \cdot f(x_n + ch, y_n + a \cdot k_1))}_{k_2}. \quad (23)$$

Z razvojem funkcij $y(x_n + h)$ in $f(x_n + ch, y_n + ak_1)$ v Taylorjevi vrsti in primerjavo koeficientov pri h in h^2 v (23) dobimo pogoja

$$\begin{aligned} 1 &= b_1 + b_2, \\ \frac{1}{2}(f_x + f_y f)_n &= b_2 c(f_x)_n + b_2 a(ff_y)_n, \end{aligned} \quad (24)$$

kjer $f_n, (f_x)_n, (f_y)_n$ pomenijo $f(x_n, y_n), f_x(x_n, y_n), f_y(x_n, y_n)$. Enačbi (24) imata veliko rešitev, npr.:

► $b_1 = b_2 = \frac{1}{2}$ in $c = a = 1$. RK metoda je:

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{2}(k_1 + k_2), \\ k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + h, y_n + k_1). \end{aligned}$$

Primer (Metode Runge-Kutta (RK) reda 2)

- $b_1 = 1, b_2 = 0$ in $c = a = \frac{1}{2}$. RK metoda je:

$$\begin{aligned}y_{n+1} &= y_n + k_2, \\k_1 &= hf(x_n, y_n), \\k_2 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1).\end{aligned}$$

Splošna RK metoda je oblike

$$\begin{aligned}y_{n+1} &= y_n + b_1 k_1 + b_2 k_2 + \dots + b_s k_s, \\k_1 &= hf(x_n, y_n), \\k_2 &= hf(x_n + c_2 h, y_n + a_{2,1} k_1), \\k_3 &= hf(x_n + c_3 h, y_n + a_{3,1} k_1 + a_{3,2} k_2), \\k_s &= hf(x_n + c_s h, y_n + a_{s,1} k_1 + \dots + a_{s,s-1} k_{s-1}).\end{aligned}\tag{25}$$

Butcherjeva tabela

RK metode (25) v kompaktni obliki shranjujemo v [Butcherjevi tabeli](#):

0	0					
c_2	$a_{2,1}$	0				
c_3	$a_{3,1}$	$a_{3,2}$	0			
$\vdots$	$\vdots$					
c_s	$a_{s,1}$	$a_{s,2}$	$a_{s,3}$	$\dots$	$a_{s,s-1}$	0
	b_1	b_2	b_3	$\dots$	b_{s-1}	b_s

kjer je še

$$c_2 = a_{2,1},$$

$$c_3 = a_{3,1} + a_{3,2},$$

$$\vdots$$

$$c_s = a_{s,1} + a_{s,2} + \dots + a_{s,s-1}.$$

Metoda Runge-Kutta reda 4

Butcherjeva tabela:

0	0			
1	1			
$\frac{1}{2}$	$\frac{1}{2}$	0		
$\frac{1}{2}$	0	$\frac{1}{2}$	0	
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Metoda je

$$y_{n+1} = y_n + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4,$$

$$k_1 = hf(x_n, y_n), \quad k_2 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right),$$

$$k_3 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right), \quad k_4 = hf(x_n + h, y_n + k_3).$$

Ocenjevanje napake in kontrola koraka

1. Pri računanju nas zanima velikost globalne napake.
2. Med izvajanjem metode ocenjujemo velikost lokalnih napak.
3. Na velikost lokalnih napak ključno vpliva izbira dolžine koraka.

Naj bo M metoda reda p , s katero izračunamo $y(x_{n+1})$ z dolžino koraka h . Približek označimo z $y_{n+1,h}$. Velja:

$$\ell_{n+1} := y_{n+1,h} - z(x_{n+1}) \approx Ch^{p+1}, \quad (26)$$

kjer je $z(x)$ rešitev začetnega problema

$$y' = f(x, y), \quad y(x_n) = y_n. \quad (27)$$

Podobno velja:

$$\ell_{n+1} = y_{n+1,h/2} - z(x_{n+1}) \approx C(h/2)^{p+1} + C(h/2)^{p+1} = 2^{-p}Ch^{p+1}, \quad (28)$$

saj smo pri koraku $h/2$ naredili dva koraka metode.

Odštejemo (28) od (26) in dobimo

$$y_{n+1,h} - y_{n+1,h/2} \approx Ch^{p+1}(1 - 2^{-p}). \quad (29)$$

Iz (29) izrazimo Ch^{p+1} in dobimo

$$Ch^{p+1} \approx \frac{y_{n+1,h} - y_{n+1,h/2}}{1 - 2^{-p}}. \quad (30)$$

1. Če je $|\ell_{n+1}| < \epsilon h$, potem $y_{n+1,h}$ sprejmemo.
V vsaki točki namreč omejimo napako na ϵ . Na celem intervalu integriramo torej napako največ ϵ in dobimo mejo ϵh .
2. Če je $|\ell_{n+1}| \geq \epsilon h$, potem ponovimo računanje približka $y(x_{n+1})$ s krajšim korakom.
3. Če je $|\ell_{n+1}|$ bistveno manjši od ϵh , lahko v nadaljevanju uporabimo daljši korak.

Globalna napaka

Lipschitzov pogoj. Funkcija f je Lipschitzova v y (na danem območju) s konstanto L , če za vse x in vse y_1, y_2 iz območja velja

$$|f(x, y_1) - f(x, y_2)| \leq L |y_1 - y_2|.$$

To je standarden (lokalno pogosto izpolnjen) pogoj: če je $\partial f / \partial y$ zvezna in omejena na območju, potem je f Lipschitzova.

(Povezava z globalno napako) Če je f Lipschitzova v y s konstanto L , potem velja

$$|g_i| \leq e^{L(x_i - a)} \sum_{k=1}^i |\ell_k| \leq e^{L(b-a)} \sum_{k=1}^i |\ell_k|.$$

Globalna napaka

Zakaj kriterij $|\ell_{n+1}| < \varepsilon h$? Če na vsakem koraku dosežemo $|\ell_k| \lesssim \varepsilon h_k$, potem

$$\sum_{k=1}^i |\ell_k| \lesssim \varepsilon \sum_{k=1}^i h_k = \varepsilon (x_i - a),$$

zato po zgornji oceni sledi

$$|g_i| \lesssim e^{L(b-a)} \sum_{k=1}^i |\ell_k| \lesssim e^{L(b-a)} \varepsilon (x_i - a) \leq e^{L(b-a)} \varepsilon (b - a).$$

(Torej εh pomeni približno konstantno napako na enoto dolžine; globalna napaka je nadzorovana do stabilnostnega faktorja.)

Sistemi diferencialnih enačb

Sistem DE je oblike:

$$\begin{aligned}y_1' &= f_1(x, y_1, \dots, y_m), \\y_2' &= f_2(x, y_1, \dots, y_m), \\&\vdots \\y_m' &= f_m(x, y_1, \dots, y_m),\end{aligned}\tag{31}$$

kjer so $y_1(x), \dots, y_m(x)$ neznane funkcije. Imamo še m začetnih pogojev $y_i(x_0) = y_{i,0}$ za $i = 1, \dots, m$. Sistem (31) lahko zapišemo v vektorski obliki:

$$\vec{y}' = \vec{f}(x, \vec{y}), \quad \vec{y}(x_0) = \vec{y}_0,\tag{32}$$

kjer so

$$\begin{aligned}\vec{y} &= (y_1, \dots, y_m), \quad \vec{f} = (f_1, \dots, f_m), \\ \vec{y}(x_0) &= (y_1(x_0), \dots, y_m(x_0)).\end{aligned}$$

Sistem (32) lahko rešujemo z Runge-Kutta metodami, le da vse funkcije podamo kot vektorske funkcije, točke pa kot vektorje.

Robni problem - strelska metoda

Robni sistem DE v dveh spremenljivkah je oblike:

$$\begin{aligned}y' &= f(x, y, z), \\z' &= g(x, y, z),\end{aligned}\tag{33}$$

kjer sta $y(x)$ in $z(x)$ neznani funkciji, $x \in [a, b]$, dana pa sta še pogoja

$$y(a) = y_a \in \mathbb{R}, \quad z(b) = z_b \in \mathbb{R}.$$

Sistem (33) na intervalu rešujemo s **strelsko metodo**, tako da ugotovimo vrednost $z(a) = \alpha_1$, rešimo začetni problem z eno od numeričnih metod in pogledamo, ali je v rešitvi res $z(b) = z_b$. To skoraj gotovo ne bo izpolnjeno.

Zato uvedemo **funkcijo napake**

$$F : \mathbb{R} \rightarrow \mathbb{R}, \quad \alpha \mapsto z_b - z(b).$$

Radi bi našli α , tako, da je $F(\alpha) = 0$. Iščemo torej ničlo funkcije F . Ker F ni eksplicitno podana, tangentne metode za iskanje ničle F ne moremo uporabiti. Lahko pa uporabimo sekantno metodo, pri čemer sprva izračunamo $F(\alpha_1)$ in $F(\alpha_2)$ za dva začetna približka α_1, α_2 .