Jan Grošelj

UVOD V NUMERIČNE METODE

Zbirka nalog z rešitvami

Foreword

This is a partial English translation of the exercise workbook that is evolving in the process of exercise lectures for the course Introduction to Numerical Methods (Uvod v numerične metode) at the Faculty of Mathematics and Physics in Ljubljana, Slovenia.

For the time being, the text of exercise instructions and theory excerpts is translated. The solutions to exercises are in Slovene. Address any questions or remarks to `jan.groselj@fmf.uni-lj.si`. Note that this is a work in progress and may contain errors. The current English version of the workbook is published at the web address

`https://www.fmf.uni-lj.si/~groseljj/gradiva/unm_zbirka_nalog_eng.pdf`,

and the original Slovene version is available at

`https://www.fmf.uni-lj.si/~groseljj/gradiva/unm_zbirka_nalog.pdf`.

Last change: 5. januar 2025

Kazalo

1. Numerical Computing

Numerical mathematics deals with the implementation of mathematical procedures from analysis and algebra on computational machines. Since these are limited to a finite set of numbers with which they can operate, we have to be careful with the implementation of the procedures and adjust them in a way that the negative effect of the finite representation is as small as possible.

1.1. Number Representation

A representable number x from $P(b,t,L,U)$ is of the form $x = \pm m \cdot b^e$, where $b \in \mathbb{N}$ is a basis, $e \in \mathbb{Z}$ is an exponent in the limits $L \leq e \leq U$ for $L, U \in \mathbb{Z}$, and

$$m = 0.c_1c_2 \ldots c_t, \qquad 0 \leq c_i \leq b-1, \quad i = 1, 2, \ldots, t,$$

is a mantissa of length $t \in \mathbb{N}$. We require $c_1 \neq 0$, except for $e = L$. The representable numbers with $c_1 \neq 0$ are called normalized, the rest are denormalized.

Exercise 1.1. Write all normalized numbers from the set $P(2,3,-1,3)$. Which of them lie on the interval $(0,1)$? Count denormalized numbers.

Solution. Normalizirana števila iz množice $P(2,3,-1,3)$ so

$$\pm 0.100_2 \cdot 2^e \quad \pm 0.101_2 \cdot 2^e, \quad \pm 0.110_2 \cdot 2^e, \quad \pm 0.111_2 \cdot 2^e$$

za $e \in \{-1, 0, 1, 2, 3\}$ oziroma

± 0.2500,	± 0.5000,	± 1.0000,	± 2.0000,	± 4.0000,
± 0.3125,	± 0.6250,	± 1.2500,	± 2.5000,	± 5.0000,
± 0.3750,	± 0.7500,	± 1.5000,	± 3.0000,	± 6.0000,
± 0.4375,	± 0.8750,	± 1.7500,	± 3.5000,	± 7.0000

v desetiškem zapisu. Na intervalu $(0,1)$ ležijo števila s pozitivnim predznakom pri $e = -1$ in $e = 0$. Denormalizirana števila so določena z mantisami 0.001_2, 0.010_2 in 0.011_2 ter najmanjšim eksponentom $e = -1$. Torej jih je vsega skupaj šest (tri pozitivna in tri negativna): ± 0.0625, ± 0.1250 in ± 0.1875.

Exercise 1.2. Let $t \in \mathbb{N}$ be the length of mantissa, and let $L, U \in \mathbb{Z}$, $L < 0 < U$, be boundaries for exponents that describe a set of representable numbers $P(\cdot, t, L, U)$. Prove that the inclusions hold:

$$P(2, t, L, U) \subseteq P(4, t, L, U) \subseteq P(2, 2t, 2L, 2U).$$

Is any of the inclusion relations an equality relation?

Solution. Število $x \in P(2, t, L, U)$ lahko predstavimo v obliki

$$x = \pm \left(c_1 \cdot 2^{-1} + c_2 \cdot 2^{-2} + \ldots + c_t \cdot 2^{-t}\right) \cdot 2^e$$

za $c_i \in \{0, 1\}$, $i = 1, 2, \ldots, t$, in $e \in \{L, L+1, \ldots, U\}$. Naj bosta t_1 in t_0 taki števili, da je

$$t = 2t_1 + t_0, \quad t_1 \leq \tfrac{1}{2}t, \quad t_0 \in \{0, 1\},$$

ter e_1 in e_0 taki števili, da je

$$e = 2e_1 + e_0, \quad |e_1| \leq \tfrac{1}{2}|e|, \quad |e_0| \in \{0, 1\}.$$

Potem je $t_1 < t$ in $L < e_1 < U$. Število x lahko zapišemo kot

$$x = \pm \left((2c_1 + c_2) \cdot 4^{-1} + \ldots + (2c_{2t_1-1} + c_{2t_1}) \cdot 4^{-t_1} + 2t_0 c_t \cdot 4^{-(t_1+1)}\right) \cdot 4^{e_1} \cdot 2^{e_0}.$$

Če je $e_0 = 0$, je iz tega že razvidno, da je $x \in P(4, t, L, U)$. Če je $e_0 \in \{-1, 1\}$, pa lahko zapis x preoblikujemo v

$$x = \pm \left(c_1 \cdot 4^{-1} + (2c_2 + c_3) \cdot 4^{-2} + \ldots + (2c_{2t_1} + t_0 c_t) \cdot 4^{-(t_1+1)}\right) \cdot 4^{e_1 + \frac{1}{2}(e_0+1)},$$

kar dokazuje, da je tudi v teh dveh primerih $x \in P(4, t, L, U)$.

Število $x \in P(4, t, L, U)$ lahko zapišemo v obliki

$$x = \pm \left(c_1 \cdot 4^{-1} + c_2 \cdot 4^{-2} + \ldots + c_t \cdot 4^{-t}\right) \cdot 4^e$$

za $c_i \in \{0, 1, 2, 3\}$, $i = 1, 2, \ldots, t$, in $e \in \{L, L+1, \ldots, U\}$. Za vsak i naj bosta $c_{i,1}, c_{i,0} \in \{0, 1\}$ taki števili, da je $c_i = 2c_{i,1} + c_{i,0}$. Potem je

$$x = \pm \left(c_{1,1} \cdot 2^{-1} + c_{1,0} \cdot 2^{-2} + \ldots + c_{t,1} \cdot 2^{-2t+1} + c_{t,0} \cdot 2^{-2t}\right) \cdot 2^{2e},$$

kar dokazuje, da je $x \in P(2, 2t, 2L, 2U)$.

Število $4^{-t} \cdot 4^L$ je vsebovano v $P(4, t, L, U)$, ni pa vsebovano v $P(2, t, L, U)$. Število $\sum_{i=1}^{2t+1} 2^{-i}$ je vsebovano v $P(2, 2t, 2L, 2U)$, ni pa vsebovano v $P(4, t, L, U)$. Torej pri nobeni izmed relacij vsebovanosti ne velja enakost.

Exercise 1.3. Use Matlab to generate all representable numbers from the set $P(5, 4, -5, 5)$ and order them from smallest to largest. Then, find answers to the following questions.

1. What is the share of the denormalized numbers?

2. How many normalized numbers are smaller than π?

3. What is the average distance between the subsequent representable numbers with the absolute distance less than 1 from π?

Solution. Najprej sestavimo program, ki izračuna seznam predstavljivih (`X`), normaliziranih (`Xn`) in denormaliziranih (`Xdn`) števil v danem sistemu.

```
b = 5; t = 4; L = -5; U = 5;

% mantise
c = 0:b-1;
M = zeros(b^t,1);
i = 1;
for c1 = c
    for c2 = c
        for c3 = c
            for c4 = c
                M(i) = (b.^-(1:t))*[c1; c2; c3; c4];
                i = i+1;
            end
        end
    end
end

% normalizirana števila
d = U-L+1;
bm = b^(t-1);
Xpn = zeros((b-1)*bm, d);
for j = 1:d
    Xpn(:,j) = M(bm+1:end) * b^(L+j-1);
end
Xpn = Xpn(:);
Xn = [-Xpn(end:-1:1); Xpn];

% denormalizirana števila
Xpdn = M(2:bm) * b^L;
Xdn = [-Xpdn(end:-1:1); Xpdn];

% predstavljiva števila (brez 0, Inf, -Inf in NaN)
X = [Xn(1:end/2); Xdn(1:end/2); Xpdn; Xpn];
```

1. Delež denormaliziranih števil izračunamo tako, da njihovo število delimo s številom vseh predstavljivih števil. Rezultat je približno 2.2%.

2. Normalizirana števila, manjša od π, lahko preštejemo z ukazom `sum(Xn<pi)`. Dobimo 8768.

3. Da dobimo predstavljiva števila, ki se od π razlikujejo za manj kot ena, uporabimo ukaz `S = X(abs(X-pi)<1)`. Nato uporabimo vgrajeni funkciji `mean` in `diff`, da izračunamo povprečni razmik `mean(diff(S))`, ki je enak 0.008.

Poskusite začetni del programa nadgraditi tako, da izračuna vse mantise splošne dolžine `t`. Nato poiščite odgovore na zastavljena vprašanja še pri kaki drugi izbiri za `t`.

The number x not contained in the selected set $P(b,t,L,U)$ is unrepresentable. We substitute it with a number $\mathrm{fl}(x)$ that is either the greatest representable number smaller than x or the smallest representable number greater than x. In rounding we choose the one that is closest to x. In this way, under the assumption that $|x|$ lies on the interval between the smallest and the largest representable number, we ensure that $\mathrm{fl}(x) = x(1+\delta)$, where δ is such a number that the absolute value $|\delta|$ is smaller than the unit roundoff $u = b^{1-t}/2$.

Exercise 1.4. Which is the greatest number from $P(5,4,-5,5)$ smaller than π, and which is the smallest number greater than π? Which of these two numbers is $\mathrm{fl}(\pi)$?

Solution. S pomočjo programa iz naloge 1.3 lahko odgovore na vprašanja poiščemo s spodnjimi ukazi.

```
x = pi;                              % 3.1416

xl = X(find(X<pi,1,'last'));         % 3.1360
xf = X(find(X>pi,1,'first'));        % 3.1440

if xf-x < x-xl
    flx = xl;
else
    flx = xf;                        % 3.1440
end
```

Pri tem je treba poudariti, da izbira $\mathrm{fl}(x)$ po zgornjem postopku za splošen x ni korektna. V primeru, da je x enako oddaljen od obeh najbližjih predstavljivih števil, se za $\mathrm{fl}(x)$ vzame tistega, ki ima sodo zadnjo števko v mantisi.

Exercise 1.5. Express the number $x = 47.712$ in the binary representation and by rounding find its closest representable number $\mathrm{fl}(x)$ from $P(2,9,-10,10)$. Check that the relative error $|\mathrm{fl}(x) - x| \,/\, |x|$ is smaller than the unit roundoff.

Solution. Dvojiški zapis celega oziroma decimalnega dela x dobimo z deljenjem

oziroma z množenjem z 2:

$$\begin{aligned} 47 &= 23 \cdot 2 + 1, & 0.712 \cdot 2 &= 0.424 + 1, \\ 23 &= 11 \cdot 2 + 1, & 0.424 \cdot 2 &= 0.848 + 0, \\ 11 &= 5 \cdot 2 + 1, & 0.848 \cdot 2 &= 0.696 + 1, \\ 5 &= 2 \cdot 2 + 1, & 0.696 \cdot 2 &= 0.392 + 1, \\ 2 &= 1 \cdot 2 + 0, & 0.392 \cdot 2 &= 0.784 + 0, \\ 1 &= 0 \cdot 2 + 1, & 0.784 \cdot 2 &= 0.568 + 1, \ldots . \end{aligned}$$

Od tod sledi $47 = 101111_2$ (ostanke v levem stolpcu prepišemo od spodaj navzgor) in $0.712 = 0.101101\ldots_2$ (celi del v desnem stolpcu prepišemo od zgoraj navzdol). Torej je

$$x = 0.101111101101\ldots_2 \cdot 2^6 \quad \text{in} \quad \mathrm{fl}(x) = 0.101111110_2 \cdot 2^6.$$

Ker je

$$\mathrm{fl}(x) - x = \left((0.101111101_2 + 2^{-9}) - (0.101111101_2 + 1.01\ldots_2 \cdot 2^{-10})\right) \cdot 2^6,$$

velja ocena

$$|\mathrm{fl}(x) - x| < \left|2^{-9} - 2^{-10}\right| \cdot 2^6 = 2^{-4}.$$

To pomeni, da je relativna napaka $|\mathrm{fl}(x) - x| \,/\, |x|$ manjša od 0.0014, kar je manj od osnovne zaokrožitvene napake $2^{1-9}/2 \approx 0.0020$.

We say that a number x is given in single precision if it is represented by $\mathrm{fl}(x)$ from the set $P(2, 24, -125, 128)$. In computer memory, such a number is saved in 32 bits. If it is normalized, it takes the form

$$\mathrm{fl}(x) = (-1)^s (1 + f) \cdot 2^{\tilde{e}-127},$$

where $s \in \{0, 1\}$ is the sign (1 bit), $\tilde{e} \in \left\{1, 2, \ldots, 2^8 - 1\right\}$ the exponent (8 bits) and $f = 0.c_2\, c_3 \ldots c_{24}$ a part of mantissa (23 bits). Similarly, 64 bits are used to save the numbers from $P(2, 53, -1021, 1024)$ which specifies double precision.

Exercise 1.6. Prove that

$$0.1 = \sum_{i=1}^{\infty} \left(2^{-4i} + 2^{-4i-1}\right),$$

and find $\mathrm{fl}(0.1)$ for 0.1 in single precision. How is that number represented in computer memory?

Solution. Vrsto izračunamo s prevedbo na geometrijsko vrsto

$$\sum_{i=1}^{\infty} \left(2^{-4i} + 2^{-4i-1}\right) = \left(1 + \frac{1}{2}\right) \sum_{i=1}^{\infty} \left(2^{-4}\right)^i = \frac{3}{2} \cdot \frac{2^{-4}}{1 - 2^{-4}} = \frac{1}{10}$$

in s tem dokažemo, da lahko število 0.1 predstavimo v želeni obliki. Iz tega rezultata sledi, da je $0.1 = 0.0\overline{0011}_2$. Ker ima 0.1 v dvojiški bazi neskončen decimalni zapis, fl(0.1) dobimo z zaokroževanjem. Na podlagi

$$0.1 = 0.1100110011001100110011001\ldots_2 \cdot 2^{-3}$$

sklepamo, da je

$$\mathrm{fl}(0.1) = 0.11001100110011001100110 1_2 \cdot 2^{-3}$$

oziroma

$$\mathrm{fl}(0.1) = (-1)^0(1 + 0.10011001100110011001101_2) \cdot 2^{123-127}.$$

Število fl(0.1) torej opišemo z biti 0, 01111011 in 10011001100110011001101, ki po vrsti določajo s, $\tilde{e}$ in f. Rezultat v Matlabu preverimo s pomočjo ukaza `single(0.1)`, ki vrne fl(0.1) za enojno natančnost.

```
x = 0.1;
flx = [repmat([1 1 0 0],1,5) [1 1 0 1]] * 2.^-(4:27)';
double(single(x))-x      % 1.4901161e-09
double(single(x))-flx    % 0
```

Exercise 1.7. In Matlab implement the function that rounds a given in double precision representable number $x \in \mathbb{R}$ to the nearest number from $P(2, t, -1021, 1024)$ where the length of the mantissa $t \in \{1, 2, \ldots, 53\}$ is an input parameter. Use the function to compute the representable numbers from Exercises 1.5 and 1.6.

Solution. Neničelno število x lahko zapišemo v obliki $x = d \cdot 2^e$, kjer je $|d| \in [1/2, 1)$ in e neko celo število. Pri iskanju fl(x) si lahko zato pomagamo z dvojiškim logaritmom. Ker je $\log_2(|x|) = \log_2(|d|)+e$ in velja $-1 \leq \log_2(|d|) < 0$, po zaokroževanju vrednosti $\log_2(|x|)$ na najbližje celo število navzdol dobimo $e-1$. V Matlabu lahko torej števili e in d določimo z ukazoma `e = floor(log2(abs(x))) + 1` in `d = x/2^e`. Hitreje in zanesljiveje do teh dveh vrednosti pridemo z ukazom `[d,e] = log2(x)`.

Za izračun fl(x) je treba vrednost d, ki si jo predstavljamo kot

$$d = \pm \left(d_1 \cdot 2^{-1} + d_2 \cdot 2^{-2} + \ldots\right), \quad d_i \in \{0, 1\},\ i = 1, 2, \ldots,$$

ustrezno zaokrožiti. Ker je

$$d \cdot 2^t = d_1 \cdot 2^{t-1} + d_2 \cdot 2^{t-2} + \ldots + d_t \cdot 2^0 + d_{t+1} \cdot 2^{-1} + \ldots,$$

po zaokroževanju števila $d \cdot 2^t$ in deljenju z 2^t dobimo število m, ki je najbližje d med vsemi števili s t decimalkami v dvojiškem zapisu. Dodatno pazljivi moramo biti v primeru, ko d leži na sredini med dvema predstavljivima številoma. Takrat je $d \cdot 2^t$ oblike $c/2$ za $c \in \mathbb{Z}$ in po standardu IEEE ga zaokrožimo na najbližje sodo število. Za tako določeno vrednost m velja fl(x) $= m \cdot 2^e$. V Matlabu lahko m izračunamo z ukazom `m = round(d*2^t,'TieBreaker','even') / 2^t`, število fl(x) pa nato z ukazom `flx = m*2^e`. Tudi tu se lahko poslužimo hitrejšega in zanesljivejšega načina, in sicer z ukazom `flx = pow2(round(pow2(d,t),'TieBreaker','even'),e-t)`.

```
function flx = fl(x,t)
% Vrne najbližje predstavljivo število flx za x v
% dvojiški bazi in mantiso dolžine t.

[d,e] = log2(x);
flx = pow2(round(pow2(d,t),'TieBreaker','even'),e-t);

end
```

Z uporabo funkcije `fl` bi radi izračunali najbližje predstavljivo število za 47.712 pri mantisi dolžine 9 in za 0.1 pri mantisi dolžine 24. Rezultati spodnjih ukazov potrdijo, da s funkcijo `fl` dobimo enaki števili, kot sta izpeljani v nalogah 1.5 in 1.6.

```
flx = [1 0 1 1 1 1 1 1 0] * 2.^(5:-1:-3)';
flx - fl(47.712,9)    % 0

flx = double(single(0.1));
flx - fl(0.1,24)      % 0
```

Exercise 1.8. Let 1_- be the largest number in double precision that is smaller than 1, and let 1_+ be the smallest number in double precision that is larger than 1. Which number in double precision is the nearest to the number $1_- \cdot 1_+$?

Solution. Števila v dvojni natančnosti so predstavljena v dvojiški bazi z mantiso dolžine 53. Največje število, ki je manjše od 1, je

$$1_- = 0.\underbrace{11\ldots1}_{53}{}_2 \cdot 2^0 = 1 - 2^{-53},$$

najmanjše število, ki je večje od 1, pa

$$1_+ = 0.1\underbrace{00\ldots0}_{51}1_2 \cdot 2^1 = 1 + 2^{-52}.$$

Produkt teh dveh števil je

$$1_- \cdot 1_+ = (1 + 2^{-52}) - (2^{-53} + 2^{-105}) = 1 + (2^{-53} - 2^{-105}),$$

iz česar je razvidno, da je $1_- \cdot 1_+$ število med 1 in 1_+ ter da je bližje 1 kot 1_+.

Exercise 1.9. Let x and y be any two consecutive positive normalized numbers from the set $P(b,t,L,U)$. Prove that $b^{-t}x \leq |x-y| \leq b^{1-t}x$.

Solution. Pozitivno število x iz množice $P(b,t,L,U)$ je oblike

$$x = \left(c_1 b^{-1} + c_2 b^{-2} + \ldots + c_t b^{-t}\right) \cdot b^e$$

za $c_i \in \{0, 1, \ldots, b-1\}$, $i = 1, 2, \ldots, t$, in $e \in \{L, L+1, \ldots, U\}$. Predpostavljamo, da je x normalizirano število, zato velja $c_1 \neq 0$. Zapišemo ga kot $x = d \cdot b^{e-t}$, kjer za

$$d = c_1 b^{t-1} + c_2 b^{t-2} + \ldots + c_t$$

velja $b^{t-1} \leq d < b^t$.

Če x ni največje normalizirano število, je najmanjše normalizirano število y, ki je večje od x, enako $y = x + b^{e-t}$. Torej je

$$|x - y| = b^{e-t} = \frac{|x|}{d}$$

in oceni za d potrjujeta dokazovano spodnjo in zgornjo mejo za $|x - y|$.

Obravnavajmo še največje pozitivno normalizirano število y, ki je manjše od x. Če je $c_1 > 1$ ali če obstaja $j \in \{2, 3, \ldots, t\}$, da je $c_j \neq 0$, je $y = x - b^{e-t}$, zato tako kot prej velja $|x - y| = b^{e-t}$. Sicer je $x = b^{t-1} \cdot b^{e-t} = b^{e-1}$ in če x ni najmanjše pozitivno normalizirano število ($e > L$), velja

$$y = \big((b-1)b^{t-1} + (b-1)b^{t-2} + \ldots + (b-1)\big) \cdot b^{e-t-1} = x - b^{e-t-1}.$$

Torej je $b^{-t}x = b^{-t} \cdot b^{e-1} = b^{e-t-1} = |x - y|$, kar potrjuje spodnjo in zgornjo mejo za $|x - y|$.

1.2. Computational Errors

In numerical mathematics we face the errors in different phases of computing.

1. Usually an error ocurrs already in the preparation of input data at the beginning of computing. The error that is the difference between the computation with real and actual data is *unremovable error*.
2. When solving a specific problem, we are often forced to compute an estimate of the exact solution due to the difficulty of the problem or its computational complexity. Thus we are not solving the original problem, but a simpler related problem, and the error caused by that is *approximation error*.
3. Lastly, we have to consider the *rounding error*, which is a consequence of rounding on every step of the metod. We need to be aware of the fact that the result of every operation is rounded to the closest representable number.

The sum of all three errors is a computational error.

Exercise 1.10. The function f is defined as $f(x) = \sqrt{1 + x}$. By computing with the set $P(10, 5, -10, 10)$ determine the value of $f(x)$ for $x = 1/13$.

1. Estimate the unremovable error caused by the representation of x.
2. Instead of the function f use the Taylor polynomial of f of degree 2 obtained by the expanison around the point 0. Estimate the approximation error.
3. Compute the value of the Taylor polynomial with the Horner algorithm. Based on computing in double precision estimate the rounding error that is caused by computing in the given arithmetics.

Solution. Ocenimo vsako izmed napak, ki se pojavi pri izvedbi postopka.

1. Najprej ocenimo neodstranljivo napako, ki nastane zaradi predstavitve x v predpisanem sistemu. Ker je $x = 0.0769230...$, je $\overline{x} = \mathrm{fl}(x) = 0.76923 \cdot 10^{-1}$. Neodstranljiva napaka D_n je podana z $D_n = f(x) - f(\overline{x})$. Njeno absolutno vrednost lahko s pomočjo izreka o povprečni vrednosti in ocene za relativno napako predstavitve x z $\overline{x}$ v dani aritmetiki ocenimo z

$$|D_n| = |f(x) - f(\overline{x})| \leq \max_{\xi \in (0,1)} |f'(\xi)|\, |x - \overline{x}| < 0.5 \cdot 10^{1-5}/2 = 0.25 \cdot 10^{-4}.$$

2. Napaka metode nastane, ker namesto s funkcijo f računamo s približkom, ki ga dobimo s pomočjo razvoja f v Taylorjevo vrsto. Konkretno, funkcijo f zamenjamo s polinomom $g(x) = 1 + x/2 - x^2/8$. Napaka metode je podana z $D_m = f(\overline{x}) - g(\overline{x})$, njeno absolutno vrednost pa lahko ocenimo z

$$|D_m| = |f(\overline{x}) - g(\overline{x})| \leq \frac{1}{3!} \max_{\xi \in (0,1)} |f'''(\xi)|\, \overline{x}^3 < \overline{x}^3/16 < 0.29 \cdot 10^{-4}.$$

3. Označimo $g(x) = a_0 + a_1 x + a_2 x^2$. Računanje vrednosti polinoma g v točki $\overline{x}$ s Hornerjevim postopkom poteka na sledeč način:

$$b_2 = a_2, \qquad b_1 = b_2 \overline{x} + a_1, \qquad b_0 = b_1 \overline{x} + a_0.$$

   Izhodni podatek postopka je b_0, ki ustreza vrednosti $g(\overline{x})$. V predpisani aritmetiki izvedbo postopka podaja naslednja tabela.

i	a_i	$b_{i+1} \cdot \overline{x}$	$c_i = \mathrm{fl}(b_{i+1} \cdot \overline{x})$	$c_i + a_i$	$b_i = \mathrm{fl}(a_i + c_i)$
2	-0.125				$-0.12500 \cdot 10^{0}$
1	0.5	-0.009615375	$-0.96154 \cdot 10^{-2}$	0.4903846	$0.49038 \cdot 10^{0}$
0	1	0.03772150074	$0.37722 \cdot 10^{-1}$	1.037722	$0.10377 \cdot 10^{1}$

   Rezultat, ki predstavlja približek za $f(x)$, označimo z y. Z izvedbo Hornerjevega postopka v dvojni natančnosti, rezultat katerega privzamemo za točno vrednost $g(\overline{x})$, dobimo približno 1.0377219, torej je zaokrožitvena napaka D_z po absolutni vrednosti manjša od $0.22 \cdot 10^{-4}$.

Ker je

$$|f(x) - y| = |f(x) - f(\overline{x}) + f(\overline{x}) - g(\overline{x}) + g(\overline{x}) - y| \leq |D_n| + |D_m| + |D_z|\,,$$

iz obravnave posameznih napak sledi, da je celotna napaka manjša od 10^{-4}.

Exercise 1.11. Two difference equations are given by

$$a_n = \tfrac{5}{2}a_{n-1} - a_{n-2}, \qquad n = 2, 3, \ldots, \qquad a_0 = 1,\ a_1 = \tfrac{1}{2},$$

$$b_n = \tfrac{10}{3}b_{n-1} - b_{n-2}, \qquad n = 2, 3, \ldots, \qquad b_0 = 1,\ b_1 = \tfrac{1}{3}.$$

1. Use $a_n = \lambda^n$, $\lambda \in \mathbb{R}$, and $b_n = \mu^n$, $\mu \in \mathbb{R}$, to find the exact solution of the difference equations.

2. In Matlab, generate arrays $\boldsymbol{a} = (a_0, a_1, \ldots, a_{50})$ and $\boldsymbol{b} = (b_0, b_1, \ldots, b_{50})$. Use command `scatter` to plot the points (n, a_n) and (n, b_n), $n = 0, 1, \ldots, 50$. Do the elements of the array match the exact values? Explain why or why not.

3. Reduce the errors that occur in computing the elements of the array $\boldsymbol{b}$ by generating the elements in the reverse order with the initial values $b_{50} = 0$ and $b_{49} = 1$ and scaling them by a constant that ensures $b_0 = 1$. Compare the obtained values with the exact ones.

Solution.

1. Uporabimo nastavka za $a_n = \lambda^n$ in $b_n = \mu^n$. Ker sta enačbi dvočlenski, dobimo v obeh primerih kvadratni enačbi z rešitvama $\lambda_1 = 1/2$, $\lambda_2 = 2$ in $\mu_1 = 1/3$, $\mu_2 = 3$. Od tod sledi, da sta splošni rešitvi oblike

$$a_n = A\left(\tfrac{1}{2}\right)^n + B\,2^n, \qquad b_n = C\left(\tfrac{1}{3}\right)^n + D\,3^n,$$

   kjer so A, B, C, D konstante, ki jih določimo iz začetnih pogojev. Dobimo $a_n = 1/2^n$ in $b_n = 1/3^n$. Vrednosti a_n in b_n z naraščajočim n padata proti 0.

2. Elemente seznamov izračunamo na podlagi rekurzivnih formul, ki določata diferenčno enačbo.

```
% seznam a
a = [1 1/2 zeros(1,49)];
for n = 3:51
    a(n) = 5/2*a(n-1) - a(n-2);
end

% seznam b
b = [1 1/3 zeros(1,49)];
for n = 3:51
    b(n) = 10/3*b(n-1) - b(n-2);
end
```

   Iz grafa na sliki 1.1a, ki ga narišemo z ukazom `scatter(0:50, a)`, je razvidno, da vrednosti a_n padajo proti 0, ko n raste, kar je glede na točno rešitev diferenčne enačbe pričakovano pričakovano. Graf je narisan v logaritemski skali, kar dosežemo z ukazom `set(gca, 'YScale', 'log')`. Graf na sliki 1.1b, narisan z

ukazom `scatter(0:50, b)`, po drugi strani kaže, da izračunane vrednosti b_n z naraščanjem n najprej padajo, nato pa začnejo rasti in so glede na točno rešitev diferenčne enačbe povsem napačne.

Razlog, da v prvem primeru dobimo točne rezultate, v drugem pa napačne, se skriva v tem, da je v prvem primeru rezultat vsake računske operacije predstavljivo število v dvojni natančnosti, medtem ko v drugem primeru operiramo z nepredstavljivimi števili. Za začetni podatek namesto $b_1 = 1/3$ uporabimo $\tilde{b}_1 = \mathrm{fl}(b_1) = b_1(1+\delta)$. Pri tem je δ sicer po absolutni vrednosti majhno število, a točna rešitev $\tilde{b}_n$ diferenčne enačbe z začetnima podatkoma b_0 in $\tilde{b}_1$ je

$$\tilde{b}_n = \left(1 - \tfrac{\delta}{8}\right)\left(\tfrac{1}{3}\right)^n + \tfrac{\delta}{8}3^n.$$

Vpliv faktorja 3^n se pri večjih vrednostih n močno pozna.

3. Iz rekurzivne zveze za vrednosti b_n izrazimo b_{n-2} in z zamikom indeksov dobimo

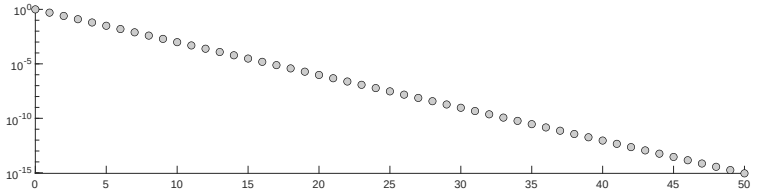
$$b_n = \tfrac{10}{3}b_{n+1} - b_{n+2}.$$

Vzamemo $b_{50} = 0$ in $b_{49} = 1$ ter elemente seznama $\boldsymbol{b}$ generiramo v obratnem vrstnem redu. Na koncu vse elemente seznama delimo z b_0 in s tem zagotovimo, da je v rezultatu $b_0 = 1$. Vrednosti tega seznama se zelo dobro ujemajo s točnimi, kar potrjuje graf na sliki 1.1c.

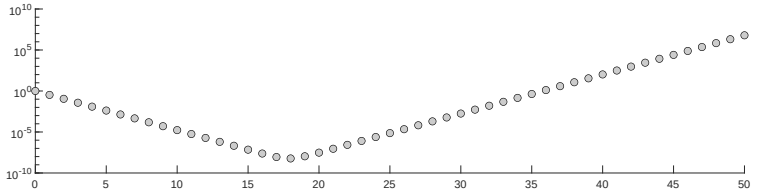
```
% seznam b, generiran v obrantem vrstnem redu
rb = [zeros(1,49) 1 0];
for n = 49:-1:1
    rb(n) = 10/3*rb(n+1) - rb(n+2);
end
b = rb/rb(1);
```

1.3. Computational Stability

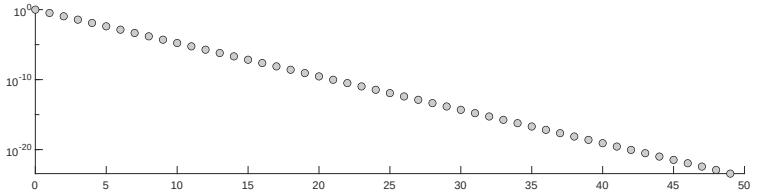
In numerical computing stability is considered in different contexts. In principal we are interested in the difference between the exact value and the computed approximation: this is the *forward error*. If the error is small for all input data, then we say that the method is forward stable. Analysis of the forward error is usually difficult, and for this reason we introduce the notion of *backward error*: this is the difference between exact input data and the input data modified in a way that the computed approximation matches the exact value on modified data. If the backward error is small for all input data, the method is considered backward stable. With backward stability we can prove forward stability of the method if the problem is unsensitive.



(A) *Elementi seznama* $\boldsymbol{a}$.



(B) *Elementi seznama* $\boldsymbol{b}$.



(C) *Elementi seznama* $\boldsymbol{b}$ *(generiran v obratnem vrstnem redu)*

SLIKA 1.1: *Prikaz rezultatov pri rekurzivnem računanju števil v nalogi 1.11.*

Exercise 1.12. A computing machine uses binary arithmetic with a mantissa of even length $t \geq 6$. Let $x = 2^{-1} + 2^{-k} + 2^{-t}$ in $y = 2^{-1} + 2^{-k}$ where $k = t/2 + 1$. We compute the value of $x^2 - y^2$ with the expression `x*x - y*y`. By analyzing the relative error, prove that the computation is not forward stable. Verify theoretical observations in Matlab by using single precision.

Solution. Za števili x in y velja

$$x^2 = 2^{-2} + 2^{-k} + 2^{-t} + 2^{-2k} + 2^{1-k-t} + 2^{-2t}, \quad y^2 = 2^{-2} + 2^{-k} + 2^{-2k}.$$

Ker je $2k = t + 2$ in $k \geq 4$, ob zaokroževanju dobimo

$$\mathrm{fl}(x^2) = 2^{-2} + 2^{-k} + 2^{-t} + 2^{-t-1}, \quad \mathrm{fl}(y^2) = 2^{-2} + 2^{-k}.$$

Pri tem upoštevamo, da je y^2 ravno na sredini med najbližjima predstavljivima številoma $2^{-2} + 2^{-k}$ in $2^{-2} + 2^{-k} + 2^{-t-1}$, in za $\mathrm{fl}(y^2)$ vzamemo prvega, ker ima sodo zadnjo števko. Vrednost $x^2 - y^2$, izračunana z izrazom `x*x - y*y`, je potem

$$z = \mathrm{fl}(\mathrm{fl}(x^2) - \mathrm{fl}(y^2)) = 2^{-t} + 2^{-t-1}$$

in za relativno napako izračuna velja

$$\frac{\left|z-(x^2-y^2)\right|}{|x^2-y^2|} = \frac{2^{-t-1}-2^{1-k-t}-2^{-2t}}{2^{-t}+2^{1-k-t}+2^{-2t}} > \frac{2^{-t-2}}{2^{-t+1}} = \frac{1}{8}.$$

Torej je relativna napaka bistveno večja od osnovne zaokrožitvene napake, ki je enaka 2^{-t}, zato izračun ni direktno stabilen.

Za preizkus v Matlabu izberemo $t = 24$, saj to ustreza dolžini mantise pri enojni natančnosti. Izračunana napaka je večja od ocene za relativno napako.

```
t = 24; k = 13;

dx = 2^-1 + 2^-k + 2^-t;
dy = 2^-1 + 2^-k;
dz = dx*dx - dy*dy;

x = single(dx);
y = single(dy);
z = x*x - y*y;

abs(double(z)-dz) / abs(dz)     % 0.4996
```

If a machine satisfies standard IEEE, every basic computer operation $\oplus$ (addition, substraction, multiplication, division) is implemented in such a way that for any two representable numbers x and y, under the assumption that no overflow occurs, it holds that

$$\mathrm{fl}(x \oplus y) = (x \oplus y)(1+\delta),$$

where δ is a number with absolute value $|\delta|$ not larger than the unit roundoff of the arithmetic in use.

Exercise 1.13. Let x and y be representable real numbers. On a machine satisfying IEEE standard we compute the value of x^2-y^2 with the expression `(x-y) * (x+y)`, which results in no overflow. Prove that the computation is backward and forward stable. In Matlab, on the example from Exercise 1.12, verify that the relative error of the computation in single precision is indeed small.

Solution. Ker računanje poteka po standardu IEEE, namesto vsote oziroma razlike števil x in y izračunamo $(x+y)(1+\alpha_1)$ in $(x-y)(1+\alpha_2)$, kjer sta α_1 in α_2 števili z absolutnima vrednostma $|\alpha_1|$ in $|\alpha_2|$, ki ne presegata osnovne zaokrožitvene napake u. Z množenjem teh dveh vrednosti dobimo

$$z = (x+y)(x-y)(1+\alpha_1)(1+\alpha_2)(1+\beta),$$

kjer je β število z lastnostjo $|\beta| \leq u$. Torej je

$$z = (x^2-y^2)(1+\delta),$$

kjer je δ število, za katero velja

$$(1-u)^3 \leq 1+\delta \leq (1+u)^3.$$

Ker je vrednost u majhna, na podlagi tega ocenimo $|\delta| \lesssim 3u$.

Iz zgornjih ugotovitev sledi, da je $z = (x\sqrt{1+\delta})^2 - (y\sqrt{1+\delta})^2$, torej je z točna rešitev pri malo zmotenih podatkih in izračun je obratno stabilen. Poleg tega za relativno napako velja

$$\frac{\left|z-(x^2-y^2)\right|}{|x^2-y^2|} \leq \frac{\left|(x^2-y^2)\delta\right|}{|x^2-y^2|} = |\delta| \lesssim 3u,$$

kar dokazuje direktno stabilnost izračuna. To potrdi tudi izračun v enojni natančnosti na primeru iz naloge 1.12. Izračunana relativna napaka je celo manjša od osnovne zaokrožitvene napake.

```
t = 24; k = 13;

dx = 2^-1 + 2^-k + 2^-t;
dy = 2^-1 + 2^-k;
dz = dx*dx - dy*dy;

x = single(dx);
y = single(dy);
z = (x-y) * (x+y);

abs(double(z)-dz) / abs(dz)     % 5.9590e-08
```

Exercise 1.14. Suppose a polynomial p is given by

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0 = a_n(x-x_n)(x-x_{n-1})\ldots(x-x_1).$$

Assume the coefficients $a_n, a_{n-1}, \ldots, a_1, a_0$ and zeros $x_n, x_{n-1}, \ldots, x_1$ are all representable real numbers. Analyze the relative error of polynomial evaluation at point x with the Horner algorithm and with the multiplication of the factors determined by the zeros of p. Is any of the procedures forward stable if we use IEEE standard? Demonstrate the procedures in Matlab in single precision for $n = 9$, $a_9 = 1$, $x_1 = x_2 = \ldots = x_9 = 2$, and $x = 2 + 2^{-5}$.

Solution. Pri Hornerjevem postopku izračun vrednosti polinoma p poteka v zaporedju

$$p(x) = ((\ldots((a_n x + a_{n-1})\,x + a_{n-2})\,x + \ldots)\,x + a_1)\,x + a_0.$$

Pri vsaki operaciji seštevanja in množenja pride do zaokrožitvene napake, ki je relativno manjša od osnovne zaokrožitvene napake u. Zato namesto $y = p(x)$ izračunamo

$$y_1 = a_n x^n(1+\delta_n) + a_{n-1}x^{n-1}(1+\delta_{n-1}) + \ldots + a_1 x(1+\delta_1) + a_0(1+\delta_0),$$

kjer za števila δ_i, $i = 0, 1, \ldots, n$, ocenjujemo, da so po absolutni vrednosti manjša od približno $2nu$ (pri vrednostih δ_i, $i = 0, 1, \ldots, n-1$, smo lahko tudi natančnejši, velja namreč $|\delta_i| \lesssim (2i+1)u$). To pomeni, da za relativno napako izračuna velja

$$\frac{|y - y_1|}{|y|} \lesssim \frac{2nu\left(|a_n|\,|x^n| + |a_{n-1}|\,\left|x^{n-1}\right| + \ldots + |a_1|\,|x| + |a_0|\right)}{|a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0|}.$$

Čeprav smo napako (približno) omejili navzgor in je lahko ta ocena pregroba, vseeno jasno odraža, da je za točke x blizu ničle polinoma relativna napaka lahko velika, če so koeficienti polinoma različno predznačeni. Postopek izračuna torej ni direktno stabilen.

V primeru, da obstaja faktorizacija polinoma na linearne faktorje, lahko izračun vrednosti opravimo stabilneje z množenjem faktorjev. Najprej izračunamo $x - x_i$, $i = 1, 2, \ldots, n$, pri čemer zaradi zaokroževanja dobimo $(x - x_i)(1 + \alpha_i)$ za neka števila α_i, ki so po absolutni vrednosti manjša od u. Nato vsako izmed n množenj botruje še k relativni napaki, manjši od u. Torej namesto $y = p(x)$ izračunamo

$$y_2 = a_n(x - x_n)\ldots(x - x_1)(1 + \delta) = y(1 + \delta),$$

pri čemer je $|\delta| \lesssim 2nu$. To pomeni, da za relativno napako velja

$$\frac{|y - y_2|}{|y|} = \frac{|y\delta|}{|y|} = |\delta| \lesssim 2nu$$

in izračun je direktno stabilen.

Zgornje ugotovitve potrjuje tudi konkreten primer izvedbe izračunov v Matlabu v enojni natančnosti. Točna vrednost pri podanih podatkih je $y = (2^{-5})^9 = 2^{-45}$. Pri računanju s Hornerjevim postopkom (funkcija `polyval`) dobimo zelo veliko relativno napako, medtem ko je napaka pri izračunu na podlagi množenja faktorjev v tem primeru celo enaka 0, saj so rezultati vseh operacij predstavljiva števila v enojni natančnosti.

```
X = single(2*ones(1,9));
A = poly(X);

x = single(2 + 2^-5);
y = 2^-45;

y1 = double(polyval(A,x));
abs(y-y1) / abs(y)     % 1.0737e+09

y2 = double(prod(x-X));
abs(y-y2) / abs(y)     % 0
```

2. Nonlinear Equations

If a nonlinear equation with the unknown x is represented in the form $f(x) = 0$, its solving can be interpreted as finding a zero of the function f. Numerical methods addressing such a problem are iterative, which means that the approximation to the solution is found in several steps with the repetition of the computations that in each step improve the approximation from the previous one.

2.1. Bisection

The bisection is a robust method for finding a zero of a continuous function f on an interval $[a, b]$ with a different sign at the endpoints of the interval (i. e., $f(a)f(b) < 0$). It is based on the fact that such a function has at least one zero on the interval $[a, b]$. As the name suggests, we perform the method by halving the interval in each step and then continue on the left or right interval, depending on which of the two the function f has a different sign.

Exercise 2.1. Implement the bisection in Matlab. Pay attention to the efficiency and numerical stability of the implementation. Run the program on the interval $[1, 2]$ for the function f given by $f(x) = x + 4 - e^{x^2}$. Perform a sufficient number of steps so that the error is less than 10^{-12}.

Solution. Pripravimo funkcijo `bisekcija`, ki kot vhodne podatke sprejme funkcijo `f`, števili `a` in `b`, ki določata robova intervala, ter število korakov bisekcije `N`. Pri implementaciji metode pazimo, da funkcijsko vrednost v določeni točki izračunamo le enkrat. S spremenljivko `s`, ki predstavlja dolžino trenutnega intervala, se izognemo morebitnim numeričnim težavam, ki bi se lahko pojavile pri izračunu središča intervala z izrazom `(a+b)/2`.

```
function x = bisekcija(f,a,b,N)
% funkcija
%  x = bisekcija(f,a,b,N)
% izvede bisekcijo in določi približek x za ničlo
% funkcije f na intervalu [a,b]
%
```

```
% vhodni podatki:
%  f     funkcija,
%  a     začetek intervala,
%  b     konec intervala,
%  N     število korakov bisekcije
%
% izhodni podatek:
%  x     približek za ničlo funkcije f

fa = f(a);
if sign(fa) == sign(f(b))
   error('f v točkah a in b ni nasprotno predznačena');
end

k = 0;
s = b-a;
while k < N
   s = s/2;
   x = a+s;
   fx = f(x);
   if sign(fa) == sign(fx)
      a = x;
      fa = fx;
   end
   k = k+1;
end

end
```

Za izračun približka za ničlo podane funkcije na intervalu $[1, 2]$, ki se od točne vrednosti razlikuje za manj kot 10^{-12}, zadošča $\lceil \log_2(10^{12}) \rceil = 40$ korakov bisekcije. S pomočjo vgrajene funkcije `fzero` se prepričamo, da v tem primeru dovolj natančen približek dobimo že z 39 koraki.

```
f = @(x) x+4-exp(x^2);
bisekcija(f,1,2,39)  % 1.290718421716520
bisekcija(f,1,2,40)  % 1.290718421715610
fzero(f,[1 2])       % 1.290718421715963
```

Exercise 2.2. Analyze how many zeros on the interval $[0, 1]$ the function f,

$$f(x) = \left(x - \frac{1}{2}\right)^2 \left(x - \frac{3}{4}\right) - \frac{1}{2^n},$$

has for a chosen $n \in \mathbb{N}$. What is the result of the bisection (according to the implementation in Exercise 2.1) if the computations are performed in single precision?

Solution. Funkcijo f lahko analiziramo na podlagi odvoda

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

Razvidno je, da je funkcija f na intervalu $[0, 1/2]$ naraščajoča, na intervalu $[0, 2/3]$ padajoča, na intervalu $[2/3, 1]$ pa ponovno naraščajoča. Ker je

$$f(0) = -\frac{3}{16} - \frac{1}{2^n}, \quad f(\tfrac{1}{2}) = -\frac{1}{2^n}, \quad f(1) = \frac{1}{16} - \frac{1}{2^n}$$

sklepamo, da je f na intervalu $[0, 2/3]$ negativna in zato tam nima ničle, na intervalu $[2/3, 1]$ pa ima eno ničlo, če je $n \geq 4$ (pri $n = 4$ je ničla ravno 1).

Pri izvedbi bisekcije za funkcijo f v enojni natančnosti vse poteka po pričakovanjih, razen če pride do podkoračitve in je $\mathrm{fl}(2^{-n}) = 0$. Najmanjše pozitivno predstavljivo število v enojni natačnosti je $2^{-24} \cdot 2^{-125}$, torej se to zgodi, če je $n > 149$. Pri izvedbi bisekcije v tem primeru dobimo $\mathrm{fl}(f(1/2)) = 0$ in (glede na implementacijo v nalogi 2.1, saj je `sign(0) = 0`) iskanje ničle nadaljujemo na intervalu $[0, 1/2]$ ter nato na intervalih $[1/4, 1/2], [3/8, 1/2], [7/16, 1/2], \ldots$. Končamo v bližini 1/2, ki pa ni dejanska ničla funkcije f.

```
a = single(0);
b = single(1);

f149 = @(x) (x-0.5)^2*(x-0.75)-2^-149;
bisekcija(f149,a,b,50)    % 0.7500
fzero(f149,[0 1])         % 0.7500

f150 = @(x) (x-0.5)^2*(x-0.75)-2^-150;
bisekcija(f150,a,b,50)    % 0.5000
fzero(f150,[0 1])         % 0.7500
```

2.2. Fixed-Point Iteration

One of the general approaches to find a zero of a function f or to solve a nonlinear equation $f(x) = 0$ is transformation of the equation into an equivalent form $x = g(x)$. The function g is called an iteration function since the solution of the equation can be found by iteration

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

If g is a contraction on a current interval that contains x_0, then by the Banach contraction principle the sequence $(x_r)_r$ converges to the fixed point of g, which corresponds to the zero of f.

Exercise 2.3. A function f is given by $f(x) = x^5 - 10x + 1$. We search for its zero with the iteration function $g(x) = (x^5 + 1)/10$.

1. Argue that the function f has exactly one zero on the interval $[0, 0.2]$.
2. Prove that the initial value $x_0 = 0$ guarantees the convergence of the iteration sequence $x_{r+1} = g(x_r)$ to the zero of f on the interval $[0, 0.2]$.
3. Estimate how good the approximation $x_2 = g(g(0))$ matches the zero of f on the interval $[0, 0.2]$.

Solution.

1. Funkcija f je za $x = 0$ enaka 1, za $x = 0.2$ pa $0.2^5 - 1 < 0$. Ker je zvezna, ima na intervalu $[0, 0.2]$ vsaj eno ničlo. Poleg tega je funkcija f na intervalu $[0, 0.2]$ padajoča, saj je $f'(x) < 0$ za vsak x s tega intervala. To potrjuje, da je ničla ena sama.

2. Odvod iteracijske funkcije g je enak $x^4/2$, zato je po absolutni vrednosti manjši od 1 natanko tedaj, ko je $|x| < \sqrt[4]{2}$. To zagotavlja konvergenco za vsak začetni približek z intervala $(-\sqrt[4]{2}, \sqrt[4]{2}) \approx (-1.1892, 1.1892)$.

3. Naj bo $\alpha \in (0, 0.2)$ ničla funkcije f ($f(\alpha) = 0$) oziroma negibna točka funkcije g ($g(\alpha) = \alpha$). Opazimo, da za vsak $x \in [0, \alpha]$ velja

$$g(x) - \alpha \leq \frac{\alpha^5 + 1}{10} - \alpha = \frac{f(\alpha)}{10} = 0,$$

kar pomeni, da se vsi členi iteracijskega zaporedja nahajajo na intervalu $[0, \alpha]$. Zato lahko po izreku o povprečni vrednosti razliko med zaporednima približkoma x_r in x_{r+1}, $r \in \mathbb{N}$, ocenimo z

$$|x_{r+1} - x_r| = |g(x_r) - g(x_{r-1})| < g'(0.2)\,|x_r - x_{r-1}| = 8 \cdot 10^{-4}\,|x_r - x_{r-1}|\,.$$

Ker začnemo iteracijo z $x_0 = 0$, v naslednjih dveh korakih dobimo $x_1 = 1/10$ in $x_2 = 1/10 + 1/10^6$. Ocenimo

$$|x_2 - \alpha| \leq |x_2 - x_3| + |x_3 - x_4| + \ldots < \left(8 \cdot 10^{-4} + \left(8 \cdot 10^{-4}\right)^2 + \ldots\right) |x_1 - x_2|$$

in od tod sklepamo

$$|x_2 - \alpha| < \frac{8 \cdot 10^{-4}}{1 - 8 \cdot 10^{-4}} 10^{-6} \approx 8 \cdot 10^{-10}.$$

Torej že drugi približek iteracije predstavlja dober približek za ničlo funkcije f.

Exercise 2.4. The iteration function g is given by $g(x) = -x^2 + 8x - 12$.

1. Find fixed points of the iteration $g(x) = x$ and determine which are attractive and which unattractive.

2. For which initial values in the neighborhood of the fixed points can we ensure the convergence of the iteration based on the derivative of g?

3. Determine for which initial values the iteration is convergent. What is the sequence limit?

Solution.

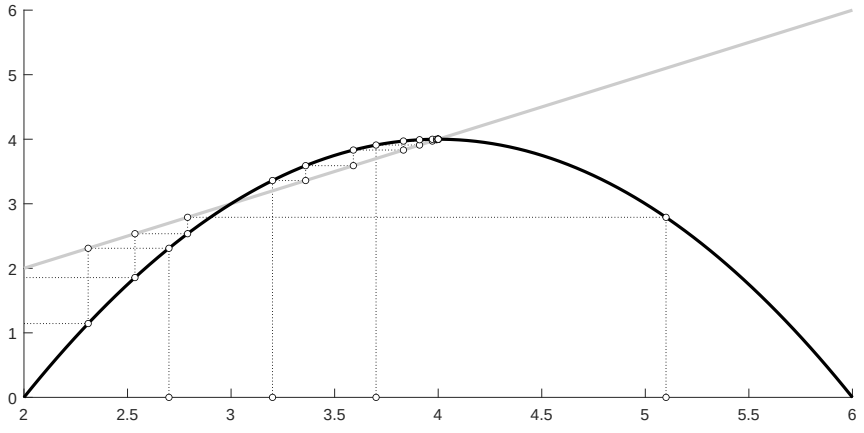
1. Negibne točke dobimo z reševanjem kvadratne enačbe $g(x) = x$. Negibni točki sta torej dve, prva je 3, druga pa 4. Z odvajanjem iteracijske funkcije dobimo, da je $g'(3) = 2$ in $g'(4) = 0$, kar pomeni, da je 3 odbojna, 4 pa privlačna točka.

2. Konvergenco navadne iteracije v okolici negibne točke $x = 4$ lahko na podlagi odvoda funkcije g zagotovimo za začetne približke x_0, za katere velja $|g'(x_0)| < 1$. Ta pogoj je izpolnjen natanko tedaj, ko je $|-2x_0 + 8| < 1$ oziroma $x_0 \in (3.5, 4.5)$.

3. Na podlagi grafa iteracijske funkcije g in simetrale lihih kvadrantov (slika 2.1 prikazuje navadno iteracijo pri začetnih približkih 2.7, 3.2, 3.7 in 5.1) domnevamo, da iteracija konvergira k 4 pri začetnih približkih z intervala $(3, 5)$, pri vseh drugih začetnih približki pa divergira ali obstane v 3. Opazimo, da za približek x_r na r-tem koraku iteracije velja

$$x_r - 4 = g(x_{r-1}) - 4 = -(x_{r-1} - 4)^2 = \ldots = -(x_0 - 4)^{2^r},$$

kar potrjuje, da za vsak $x_0 \in (3, 5)$ velja $\lim_{r\to\infty} x_r = 4$. Od tod sledi tudi, da se za vsak $x_0 \in (-\infty, 3) \cup (5, \infty)$ približki x_r zmanjšujejo brez meje, pri začetnih približkih $x_0 = 3$ in $x_0 = 5$ pa velja $x_r = 3$ za vsak $r \in \mathbb{N}$.



Slika 2.1: *Grafični prikaz navadne iteracije iz naloge 2.4.*

Exercise 2.5. In Matlab compose a function that performs the fixed-point iteration.

1. Let the input data be an iteration function, an initial value, a tolerance, and a maximal number of steps.
2. Let the output data be the approximation after finished iteration, the array of all computed approximations, and the number of performed steps.
3. Let the function perform the iteration until two last approximations differ for less than the specified tolerance or the number of steps exceeds the specified maximal number of steps.

Test the implementation with the iteration function from Exercise 2.4.

Solution. Funkcijo, ki izvede navadno iteracijo, poimenujemo `iteracija`.

```
function [x,X,k] = iteracija(g,x0,tol,N)
% funkcija
%  [x,X,k] = iteracija(g,x0,tol,N)
% izvede navadno iteracijo z dano iteracijsko funkcijo
% in začetnim približkom
%
% vhodni podatki:
%  g     iteracijska funkcija,
%  x0    začetni približek,
%  tol   toleranca absolutnega ujemanja dveh zaporednih
%        približkov,
%  N     maksimalno število korakov iteracije
%
% izhodni podatki:
%  x     zadnji približek izračunan z navadno iteracijo,
%  X     seznam vseh izračunanih približkov,
%  k     število opravljenih korakov iteracije

X = x0;
k = 0;
while k < N
    k = k+1;
    X(k+1) = g(X(k));
    if abs(X(k+1)-X(k)) < tol
        break;
    end
end
x = X(k+1);

end
```

Test kaže, da se računski rezultati ujemajo s teoretičnimi izpeljavami iz naloge 2.4.

```
g = @(x)-x^2+8*x-12; tol = 1e-10; N = 1e3;
[x,~,k] = iteracija(g,3.5,tol,N)     % x = 4, k = 7
[x,~,k] = iteracija(g,4.5,tol,N)     % x = 4, k = 7
[x,~,k] = iteracija(g,3,tol,N)       % x = 3, k = 1
[x,~,k] = iteracija(g,2,tol,N)       % x = -Inf, k = 1000
```

In practice it is imporant how fast the iteration sequence $(x_r)_r$ converges to the fixed point α. We say that the order of convergence is p if there exists a constant $C > 0$ such that

$$\lim_{r\to\infty} \frac{|x_{r+1} - \alpha|}{|x_r - \alpha|^p} = C.$$

If the function g is sufficiently continuously differentiable, the order p can be easily determined by derivation: it must hold that $g^{(k)}(\alpha) = 0$, $k = 1, 2, \ldots, p-1$, and $g^{(p)}(\alpha) \neq 0$ with assumption $|g'(\alpha)| < 1$ if $p = 1$.

Exercise 2.6. To find a zero of the function $f(x) = x^2 - x - 2$ use four different iteration functions:

1. $g_1(x) = x^2 - 2$,
2. $g_2(x) = \sqrt{x+2}$,
3. $g_3(x) = 1 + 2/x$,
4. $g_4(x) = (x^2 + 2)/(2x - 1)$.

For every function analyze the convergence in the neighborhood of the zeros -1 and 2 and determine its order. Verify the conclusions in Matlab with the help of the function implemented in Exercise 2.5. Plot graphs of the number of steps of iteration in dependence of initial values on the interval $[-2, 4]$.

Solution. Premislimo, kako se iteracijske funkcije obnašajo v okolici ničel funkcije f.

1. Ker je $g_1'(x) = 2x$, sta tako -1 kot 2 odbojni negibni točki iteracijske funkcije g_1. Ničlo funkcije f torej dobimo le v posebnih primerih, ko za začetni približek izberemo -2, -1, 0, 1 ali 2.

2. Iz $g_2'(x) = 1/\sqrt{x+2}$ sledi, da je 2 privlačna negibna točka, v okolici katere je red konvergence enak 1. Enostavno je dokazati, da iteracijsko zaporedje konvergira k 2 za vsak začetni približek, ki je večji ali enak -2. Na drugi strani -1 ni negibna točka funkcije g_2, zato te ničle funkcije f z g_2 ne moremo poiskati.

3. Najprej opazimo, da sta negibni točki funkcije g_3 tako -1 kot 2, vendar iz $g_3'(-1) = -2$ in $g_3'(2) = -1/2$ sklepamo, da je le 2 privlačna negibna točka. Za približke v okolici -1 torej ni pričakovati konvergence k -1, v -1 z iteracijo končamo le, če v -1 z njo začnemo. Po drugi strani vrednost odvoda g_3 v točki 2

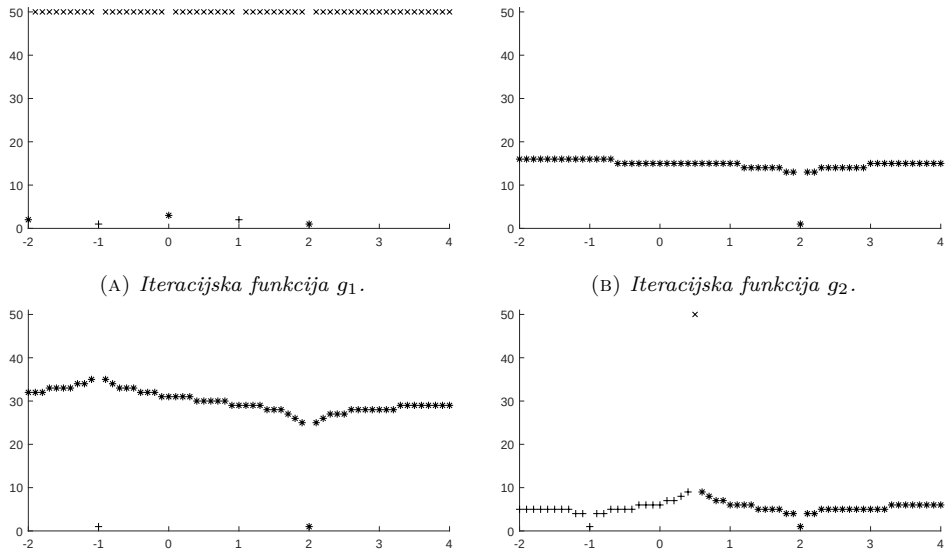
zagotavlja konvergenco k -2 za začetne približke v okolici -2, a ponovno le reda 1. S podrobnejšo analizo iteracijske funkcije lahko dokažemo, da iteracijsko zaporedje konvergira k -2 za vse začetne približke, razen -1. Pri začetnih približkih 0 in -2 v prvem oziroma drugem koraku iteracije delimo z 0.

4. Funkcija g_4 ima negibni točki -1 in 2, v obeh pa je vrednost odvoda g_4 enaka 0. Red konvergence v okolici -1 in 2 je torej vsaj reda 2 in za začetne približke blizu negibnih točk se lahko nadejamo hitre konvergence k eni ali drugi ničli funkcije f. Iz analize vrednost $g_4(x) - x$ sledi, da iteracijsko zaporedje konvergira k 2 za začetni približek večji od $1/2$ in k -1 za začetni približek manjši od $1/2$.

V Matlabu preverimo, kakšne rezultate dobimo z uporabo iteracijskih funkcij pri začetnih približkih $-1/2$ in 3. Iteracijo izvedemo s pomočjo funkcije `iteracija` iz naloge 2.5 pri toleranci `tol = 1e-10` in maksimalnem številu korakov `N = 100`.

```
g1 = @(x) x.^2-2;
g2 = @(x) sqrt(x+2);
g3 = @(x) 1+2./x;
g4 = @(x) (x.^2+2)./(2*x-1);

tol = 1e-10; N = 100;

[x,~,k] = iteracija(g1,-0.5,tol,N) % x = 0.5914, k = 100
[x,~,k] = iteracija(g1,3,tol,N)    % x = Inf,    k = 100
[x,~,k] = iteracija(g2,-0.5,tol,N) % x = 2,      k = 19
[x,~,k] = iteracija(g2,3,tol,N)    % x = 2,      k = 18
[x,~,k] = iteracija(g3,-0.5,tol,N) % x = 2,      k = 39
[x,~,k] = iteracija(g3,3,tol,N)    % x = 2,      k = 35
[x,~,k] = iteracija(g4,-0.5,tol,N) % x = -1,     k = 5
[x,~,k] = iteracija(g4,3,tol,N)    % x = 2,      k = 6
```

Na sliki 2.2 so prikazani grafi števila korakov iteracij pri začetnih približkih iz seznama `linspace(-2,4,61)`. Oznaka `*` predstavlja konvergenco k ničli 2, oznaka `+` pa konvergenco k ničli -1. Oznaka `x` pomeni, da se je iteracija končala brez konvergence po maksimalnem številu korakov (v tem primeru 50). Slika 2.2a prikazuje rezultate za iteracijsko funkcijo g_1, kjer v eni izmed ničel končamo le pri začetnih približkih -2, -1, 0, 1 in 2. Na sliki 2.2b je prikazano število korakov iteracij pri funkcji g_2, ki ima za negibno točko ničlo 2. Število korakov, s katerimi dosežemo natančnost približka v okviru predpisane tolerance, je manjše kot pri iteracijski funkciji g_3, kot kaže slika 2.2c. Iz te slike je tudi jasno razvidno, da je 1 odbojna točka iteracije z g_3, saj iteracijsko zaporedje le pri začetnem približku -1 konvergira k -1. Iteracijska zaporedja najhitreje konvergirajo pri iteracijski funkciji g_4. Slika 2.2d kaže, da je limita za začetne približke manjše od $1/2$ enaka -1, za začetne približke večje od $1/2$ pa 2. Pri začetnem približku $1/2$ iteracijsko zaporedje ne konvergira k nobeni izmed ničel.



(A) *Iteracijska funkcija* g_1.

(B) *Iteracijska funkcija* g_2.

(C) *Iteracijska funkcija* g_3.

(D) *Iteracijska funkcija* g_4.

SLIKA 2.2: *Število korakov v odvisnosti od začetnih približkov pri iteracijah iz naloge 2.6.*

Exercise 2.7. Prove that the square root of a positive number a can be computed with the iteration

$$x_{r+1} = x_r \frac{x_r^2 + 3a}{3x_r^2 + a}, \quad r = 0, 1, \ldots,$$

for any inital value $x_0 > 0$ and determine the order of convergence of the iteration sequence in the neighborhood of $\sqrt{a}$.

Solution. Enostavno je preveriti, da je funkcija $g(x) = x(x^2+3a)/(3x^2+a)$ iteracijska funkcija za enačbo $x(x^2 - a) = 0$. Negibne točke iteracije so $-\sqrt{a}$, 0 in $\sqrt{a}$.

Radi bi dokazali, da iteracija za $x_0 > 0$ konvergira k negibni točki $\sqrt{a}$. Najprej opazimo, da za vsak približek x_r, $r \in \mathbb{N}$, velja

$$x_r - \sqrt{a} = g(x_{r-1}) - \sqrt{a} = \frac{(x_{r-1} - \sqrt{a})^3}{3x_{r-1}^2 + a}.$$

Od tod po indukciji sledi, da je za začetni približek $x_0 < \sqrt{a}$ vsak približek x_r manjši od $\sqrt{a}$. Podobno, če je $x_0 > \sqrt{a}$, je tudi vsak približek x_r večji od $\sqrt{a}$. Nadalje, ker je

$$x_{r+1} - x_r = g(x_r) - x_r = \frac{2x_r(a - x_r^2)}{3x_r^2 + a},$$

za $x_r < \sqrt{a}$ velja $x_{r+1} > x_r$, za $x_r > \sqrt{a}$ pa $x_{r+1} < x_r$. To dokazuje, da je zaporedje približkov $(x_r)_r$ pri začetnem približku $x_0 > 0$ konvergentno, saj je bodisi strogo

naraščajoče in navzgor omejeno s $\sqrt{a}$ bodisi strogo padajoče in navzdol omejeno s $\sqrt{a}$. Naj bo limita zaporedja označena z α. Zanjo velja

$$\alpha = \lim_{r\to\infty} x_r = \lim_{r\to\infty} x_{r+1} = \lim_{r\to\infty} g(x_r) = g(\lim_{r\to\infty} x_r) = g(\alpha),$$

zato je negibna točka funkcije g. Ker je $\sqrt{a}$ edina neigbna točka g na intervalu $(0,\infty)$, je $\alpha = \sqrt{a}$.

Red konvergence določimo z odvajanjem. Ker je

$$g'(x) = \frac{3(x^2-a)^2}{(3x^2+a)^2}, \qquad g''(x) = \frac{48xa}{(3x^2+a)^3}(x^2-a),$$

je $g'(\sqrt{a}) = g''(\sqrt{a}) = 0$ in red konvergence je vsaj kubičen. Red v resnici je kubičen, saj je $g'''(\sqrt{a}) \neq 0$, kot sledi iz

$$g'''(x) = \left(\frac{48xa}{(3x^2+a)^3}\right)'(x^2-a) + \frac{48xa}{(3x^2+a)^3}2x.$$

Exercise 2.8. Determine the parameters α, β, and γ in the iteration formula

$$x_{r+1} = \alpha x_r + \beta\frac{a}{x_r^2} + \gamma\frac{a^2}{x_r^5}, \quad r = 0, 1, \ldots,$$

for computing the cubic square root of a non-zero number a such that the convergence of the iteration sequence for an initial value in the neighborhood of $\sqrt[3]{a}$ is cubic.

Solution. Obravnavajmo iteracijsko funkcijo $g(x) = \alpha x + \beta a/x^2 + \gamma a^2/x^5$. Če želimo, da je $\sqrt[3]{a}$ negibna točka g, mora veljati $\alpha + \beta + \gamma = 1$. Da bo red konvergence vsaj kubičen, mora biti $g'(\sqrt[3]{a}) = 0$ in $g''(\sqrt[3]{a}) = 0$, kar je ekvivalentno zahtevama $\alpha - 2\beta - 5\gamma = 0$ in $6\beta + 30\gamma = 0$. Z reševanjem sistema enačb za dane parametre ugotovimo, da mora biti $\alpha = 5/9$, $\beta = 5/9$ in $\gamma = -1/9$. Ker je pri teh parametrih $g'''(\sqrt[3]{a}) = 10/\sqrt[3]{a^2} \neq 0$, je red konvergence kubičen.

2.3. Method Derivations and Properties

To find a zero of a function f there exists a number of recipes that can be used to derive a suitable function for performing fixed-point iteration. One such is the Newton's (also called Newton–Raphson or tangent) method. The approximation x_{r+1} for a zero of a differentiable function f is produced from x_r by

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

Geometrically, x_{r+1} is the intersection of abcissa and the tangent line of f at the point x_r.

Exercise 2.9. The Babylonian method for computing the square root $\sqrt{a}$ of a positive number a is based on the iteration

$$x_{r+1} = \frac{1}{2}\left(x_r + \frac{a}{x_r}\right), \quad r = 0, 1, \ldots.$$

For performing one step of the iteration only three basic operations are needed.

1. Verify that the iteration corresponds to the Newton's method for $f(x) = x^2 - a$.
2. What is the order of convergence of the iteration in the neighborhood of $\sqrt{a}$?
3. Prove that the sequence $(x_r)_r$ converges to $\sqrt{a}$ for any initial value $x_0 > 0$.

Solution.

1. V formulo, ki določa približek po tangentni metodi, vstavimo $f(x) = x^2 - a$. Z nekaj preurejanja dobimo babilonsko metodo.
2. Red konvergence babilonske metode v okolici $\sqrt{a}$ lahko določimo z odvajanjem iteracijske funkcije $g(x) = (x + a/x)/2$. Izračunamo

   $$g'(x) = \frac{1}{2}\left(1 - \frac{a}{x^2}\right), \qquad g''(x) = \frac{a}{x^3}$$

   in iz $g'(\sqrt{a}) = 0$ in $g''(\sqrt{a}) = 1/\sqrt{a} \neq 0$ sklepamo, da je red konvergence v okolici $\sqrt{a}$ kvadratičen.
3. Odvod funkcije g je po absolutni vrednosti manjši od 1 za vsak $x \in (\sqrt{a/3}, \infty)$, zato zaporedje $(x_r)_r$ konvergira k $\sqrt{a}$ za vsak začetni približek x_0 s tega intervala. Treba je dokazati še, da to velja tudi v primeru, ko je $x_0 \in (0, \sqrt{a/3}]$. Opazimo, da je

   $$g(x_0) - \sqrt{a} = \frac{x_0}{2} + \frac{a}{2x_0} - \sqrt{a} = \frac{1}{2x_0}\left(x_0 - \sqrt{a}\right)^2,$$

   iz česar sledi, da za vsak začetni približek $x_0 > 0$ približek $x_1 = g(x_0)$ leži na intervalu $[\sqrt{a}, \infty)$; tu pa je g po prejšnjem razmisleku skrčitev, zato je limita iteracijskega zaporedja negibna točka $\sqrt{a}$.

Exercise 2.10. Let $f : \mathbb{R} \to \mathbb{R}$ be at least twice differentiable function with a zero α. We say that α is a zero of multiplicity m, $m \in \mathbb{N}$, if f can be expressed as $f(x) = (x - \alpha)^m$ for a function $h : \mathbb{R} \to \mathbb{R}$ that does not have a zero at α.

1. Let α be a simple zero ($m = 1$). Verify that the order of convergence of the Newton's method in its neighborhood is at least quadratic.
2. Let $m > 1$. Prove that the iteration function $g(x) = x - f(x)/f'(x)$ of the Newton's method satisfies

   $$\lim_{x \to \alpha} g'(x) = 1 - \frac{1}{m}.$$

   Based on this, comment on the speed of convergence of the Newton's method.

3. How would you modify the Newton's method to ensure that the order of convergence in the neighborhood of zero of multiplicity $m > 1$ is at least quadratic?

Solution.

1. Za odvod iteracijske funkcije g velja

$$g'(x) = \left(x - \frac{f(x)}{f'(x)}\right)' = 1 - \frac{f'(x)^2 - f(x)f''(x)}{f'(x)^2} = \frac{f(x)f''(x)}{f'(x)^2},$$

torej je $g'(\alpha) = 0$ in red konvergence v okolici enostavne ničle α je vsaj kvadratičen.

2. Glede na definicijo kratnosti ničle lahko prvi odvod funkcije f izrazimo kot

$$f'(x) = m(x-\alpha)^{m-1}h(x) + (x-\alpha)^m h'(x),$$

drugi odvod pa kot

$$f''(x) = (m-1)m(x-\alpha)^{m-2}h(x) + 2m(x-\alpha)^{m-1}h'(x) + (x-\alpha)^m h''(x).$$

Iz tega sledi, da za odvod iteracijske funkcije g velja

$$g'(x) = \frac{(x-\alpha)^2 h(x)h''(x) + 2m(x-\alpha)h(x)h'(x) + (m-1)mh(x)^2}{(x-\alpha)^2 h'(x)^2 + 2m(x-\alpha)h(x)h'(x) + m^2 h(x)^2},$$

na podlagi česar sklepamo, da je

$$\lim_{x\to\alpha} g'(x) = \frac{(m-1)m}{m^2} = 1 - \frac{1}{m}.$$

To pomeni, da je red konvergence v okolici večkratne ničle linearen in tudi, da je hitrost konvergence tem manjša, čim večja je kratnost ničle.

3. Iteracijsko funkcijo g bi radi zamenjali s sorodno funkcijo G, za katero velja $\lim_{x\to\alpha} G'(x) = 0$. Ker je $m \lim_{x\to\alpha} g'(x) = m - 1$, poskusimo s funkcijo

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

Njen odvod je

$$G'(x) = 1 - m\frac{f'(x)^2 + f(x)f''(x)}{f'(x)^2} = 1 - m + mg'(x)$$

in je v limiti, ko gre x proti α, enak 0. Seveda lahko metodo z iteracijsko funkcijo h izkoristimo le, če vnaprej poznamo kratnost ničle funkcije f.

Exercise 2.11. Let $f : \mathbb{R} \to \mathbb{R}$ be at least three times differentiable function and α its zero of multiplicity more than 1.

1. Prove that α is a simple zero of $F(x) = f(x)/f'(x)$.
2. Derive the method for finding the zero of f that corresponds to the Newton's method for the function F. Argue that on the neighborhood of α the convergence order of the derived method is at least quadratic.
3. Let g denote the iteration function from the second item. Prove that every zero of f is a fixed point of g and that every fixed point of g, which is not a zero of f, is unattractive.

Solution.

1. Naj $m \in \mathbb{N}$ označuje kratnost ničle α. Torej je $f(x) = (x-\alpha)^m h(x)$, kjer je $h : \mathbb{R} \to \mathbb{R}$ funkcija, za katero velja $h(\alpha) \neq 0$. Funkcijo F lahko potem zapišemo v obliki $F(x) = (x-\alpha)H(x)$ za funkcijo $H : \mathbb{R} \to \mathbb{R}$, ki je podana s predpisom

$$H(x) = \frac{h(x)}{mh(x) + (x-\alpha)h'(x)}.$$

   Ker je $H(\alpha) = \frac{1}{m} \neq 0$, je α enostavna ničla funkcije F.

2. Iteracijska funkcija je podana z

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

   in ker je red konvergence tangentne metode v okolici enostavne ničle kvadratičen, je ta lastnost za funkcijo g v okolici α zagotovljena po prvi točki. Z upoštevanjem predpisa za F iteracijsko funkcijo izrazimo glede na f kot

$$g(x) = x - \frac{f(x)f'(x)}{f'(x)^2 - f(x)f''(x)}.$$

3. Iz predpisa za funkcijo g je razvidno, da je $\beta \in \mathbb{R}$ njena negibna točka natanko tedaj, ko je $f(\beta) = 0$ ali $f'(\beta) = 0$. Če je torej β negibna točka g in ni ničla funkcije f, je $f'(\beta) = 0$ in iz predpisa za odvod

$$\begin{aligned} g'(x) = 1 &- \frac{f'(x)^2 + f(x)f''(x)}{f'(x)^2 - f(x)f''(x)} \\ &- f(x)f'(x)\frac{2f'(x)f''(x) - f'(x)f''(x) - f(x)f'''(x)}{\left(f'(x)^2 - f(x)f''(x)\right)^2} \end{aligned}$$

   sledi, da je $g'(\beta) = 2$, kar pomeni, da je β odbojna negibna točka.

Exercise 2.12. In dependence of the initial values analyze the convergence of the Newton's method for the function $f(x) = x^3 - x$.

1. Expand f into the Taylor series around the current approximation and evaluate it at point 1. Use the obtained formula to prove that for any initial value from $(1/\sqrt{3}, \infty)$ the method converges to 1. By a similar line of arguments verify that for an initial value from $(-\infty, -1/\sqrt{3})$ the method converges to -1.

2. Describe what issues arise with the initial values $\pm 1/\sqrt{3}$ and $\pm 1/\sqrt{5}$.

3. How does the iteration sequence behave for initial values from $(-1/\sqrt{5}, 1/\sqrt{5})$. Does the method converge to 0?

4. Find out what is the limit of the iteration sequence for the initial values from the intervals $(-1/\sqrt{3}, -1/\sqrt{5})$ and $(1/\sqrt{5}, 1/\sqrt{3})$.

Solution. Iteracijska funkcija je podana s predpisom

$$g(x) = x - \frac{x^3 - x}{3x^2 - 1} = \frac{2x^3}{3x^2 - 1}.$$

Vemo, da so negibne točke iteracije ničle funkcije f, to so -1, 0 in 1. Ker so enostavne ničle, vemo tudi, da je konvergenca v neki njihovi okolici vsaj kvadratična. Oglejmo si natančneje, kaj se dogaja z iteracijskimi zaporedji pri različnih začetnih približkih.

1. Iz razvoja f v Taylorjevo vrsto okoli x_{r-1}, izvrednotenega v ničli 1, sledi, da je

$$x_r - 1 = \frac{f''(\xi_{r-1})}{2f'(x_{r-1})}(x_{r-1} - 1)^2, \qquad r \in \mathbb{N},$$

   za nek ξ_{r-1} med 1 in x_{r-1}. Ker je f na intervalu $(1/\sqrt{3}, \infty)$ strogo naraščajoča in strogo konveksna, za vsak x_{r-1} s tega intervala velja $x_r > 1$ za vsak $r \in \mathbb{N}$. V posebnem to pomeni, da za začetni približek $x_0 \in (1/\sqrt{3}, \infty)$ velja $f(x_1) > 0$, zato iz definicije tangentne metode sledi $1 < x_2 < x_1$. Po indukciji lahko ta sklep nadaljujemo do ugotovitve, da je zaporedje $(x_r)_r$ strogo padajoče in navzdol omejeno z 1. Ker je 1 edina ničla funkcije f na intervalu $(1/\sqrt{3}, \infty)$, smo s tem dokazali, da za vsak začetni približek s tega intervala zaporedje $(x_r)_r$ konvergira k 1. Podoben razmislek pripelje do zaključka, da zaporedje $(x_r)_r$ konvergira k -1 za vsak začetni približek $x_0 \in (-\infty, -1/\sqrt{3})$.

2. Tangentna metoda pri začetnih približkih $x_0 = \pm 1/\sqrt{3}$ propade, saj sta $\pm 1/\sqrt{3}$ pola iteracijske funkcije g. Težave lahko pričakujemo tudi pri začetnih približkih $\pm 1/\sqrt{5}$, saj sta rešitvi enačbe

$$g(x_0) = \frac{2x_0^3}{3x_0^2 - 1} = -x_0.$$

   To pomeni, da za prva dva približka velja $x_2 = -x_1 = x_0$, iz česar sklepamo, da začetna približka $\pm 1/\sqrt{5}$ povzročita cikel reda 2.

3. Obravnavajmo začetne približke z intervala $(-1/\sqrt{5}, 1/\sqrt{5})$. Prepričajmo se, da zagotavljajo konvergenco k 0. Najprej iz padanja iteracijske funkcije g na obravnavanem intervalu sklepamo, da je $g(x) \in (0, 1/\sqrt{5})$ za vsak $x \in (-1/\sqrt{5}, 0)$ in $g(x) \in (-1/\sqrt{5}, 0)$ za vsak $x \in (0, 1/\sqrt{5})$. Nato opazimo še, da je $g(x) + x < 0$ za vsak $x \in (-1/\sqrt{5}, 0)$ in $g(x) + x > 0$ za vsak $x \in (0, 1/\sqrt{5})$. Od tod sledi, da členi zaporedja alternirajoče menjavajo predznak. Če je $x_0 > 0$, zaporedje sodih členov pada proti 0, zaporedje lihih členov pa raste proti 0. Za $x_0 < 0$ je situacija ravno obratna, ne glede na to pa celotno zaporedje približkov konvergira k 0.

4. Za začetne približke z intervala $(-1/\sqrt{3}, -1/\sqrt{5})$ lahko s pomočjo grafa iteracijske funkcije g in simetrale lihih kvadrantov razberemo, da zaporedje približkov konvergira bodisi k -1 bodisi k 1, ki je najbolj oddaljena ničla. Pri začetnem približku $x_0 = -1/2$ je na primer $x_1 = 1$, zato je zaradi zveznosti g pri začetnih približkih blizu $-1/2$ približek x_1 blizu 1, kar implicira konvergenco k 1. Konvergenco k -1 v resnici dobimo le za začetne približke x_0 na majhnem odseku, kjer je $g(x_0) \in (1/\sqrt{5}, 1/\sqrt{3})$. Po simetriji podobno velja za začetne približke z intervala $(1/\sqrt{5}, 1/\sqrt{3})$. Zaključimo, da je tangentna metoda za določene začetne približke lahko zelo nepredvidljiva.

Exercise 2.13. Give an example of a function f for which the Newton's method fails independently of the chosen initial value. Namely, find a function f such that for any initial value x_0 every approximation x_r, $r \in \mathbb{N}$, satisfies $x_r = -x_{r-1}$.

Solution. Poskušajmo določiti tako funkcijo f, da pri tangentni metodi za vsak začetni približek x_0 velja $x_r = -x_{r-1}$, $r \in \mathbb{N}$. To pomeni, da se tekom iteracije izmenjujeta približka x_0 in $-x_0$. Ker je $x_r = x_{r-1} - f(x_{r-1})/f'(x_{r-1})$, mora biti funkcija f rešitev navadne diferencialne enačbe

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

in zato zadošča $|f(x)| = C\sqrt{|x|}$ za neko konstanto C. Vzemimo na primer

$$f(x) = \sqrt{|x|}.$$

Ker je $f'(x) = x/(2\,|x|\,\sqrt{|x|})$, res velja

$$x_r = x_{r-1} - \frac{f(x_{r-1})}{f'(x_{r-1})} = x_{r-1} - \frac{\sqrt{|x_{r-1}|} \cdot 2\,|x_{r-1}|\,\sqrt{|x_{r-1}|}}{x_{r-1}} = -x_{r-1}$$

za vsak $r \in \mathbb{N}$.

Exercise 2.14. Implement the Newton's method in Matlab and use it to find the zero of the function f defined by $f(x) = x + 4 - e^{x^2}$. How many iterations with the initial value $x_0 = 1$ are needed so that the last computed approximation agrees with the approximation obtained by the built-in function `fzero` in 10 decimal places?

Solution. Pri implementaciji uporabimo funkcijo `iteracija` iz naloge 2.5.

```
function [x,X,k] = tangentna(f,df,x0,tol,N)
% funkcija
%  [x,X,k] = tangentna(f,df,x0,tol,N)
% izvede tangentno metodo za iskanje ničle funkcije f
%
% vhodna podatka:
%  f    funkcija, ničlo katere iščemo,
%  df   odvod funkcije f.
%
% ostali vhodni in izhodni podatki so enaki kot pri
% funkciji 'iteracija'

g = @(x) x - f(x)/df(x);
[x,X,k] = iteracija(g,x0,tol,N);

end
```

V tabeli 2.1 je podanih prvih osem približkov tangentne metode, ki jih dobimo z izvedbo ukaza `[x,X,k] = tangentna(f,df,1,1e-15,100)` za primerno definirani funkciji `f` in `df`. Primerjamo jih z vrednostjo 1.29071842171596, ki je rezultat ukaza `fzero(f,0)`. Ugotovimo, da je približek x_6 na šestem koraku prvi, ki se z rezultatom vgrajene metode ujema v več kot desetih decimalkah. V zadnjem stolpcu tabele je podano skupno število funkcijskih izračunov funkcij f in f', uporabljenih za izračun posameznega približka.

korak	približek	napaka	funkcijski izračuni
1	1.51429853102254	$2.2358 \cdot 10^{-1}$	2
2	1.36287475280475	$7.2156 \cdot 10^{-2}$	4
3	1.29944386111342	$8.7254 \cdot 10^{-3}$	6
4	1.29085495577572	$1.3653 \cdot 10^{-4}$	8
5	1.29071845546466	$3.3749 \cdot 10^{-8}$	10
6	1.29071842171597	$1.9984 \cdot 10^{-15}$	12
7	1.29071842171596	0	14
8	1.29071842171596	0	16

Tabela 2.1: *Izvedba tangentne metode v nalogi 2.14.*

A downside of the Newton's method is that alongside the function value also the value of its derivative is needed. To avoid this, we can use the secant method, which can be geometrically interpreted in a similar way as the Newton's method: a new approximation x_{r+1} is determined as the intersection of the abscissa and the secant line of f passing through x_r and x_{r-1}, which implies

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

The secant method is not an example of the fixed-point iteration since for the computation of the new approximation two previous approximations are needed. For this reason we need two initial values to start the iteration (x_0 and x_1).

Exercise 2.15. Prove that the iteration formula

$$x_{r+1} = \frac{x_{r-1}x_r + a}{x_{r-1} + x_r}, \quad r = 1, 2, \ldots,$$

for computing the square root of a positive number a corresponds to the secant method for the function $f(x) = x^2 - a$ and that the sequence $(x_r)_r$ converges for any initial values x_0 and x_1 greater than $\sqrt{a}$.

Solution. Po definiciji sekantne metode za funkcijo $f(x) = x^2 - a$ izračunamo

$$x_{r+1} = x_r - \frac{(x_r^2 - a)(x_r - x_{r-1})}{(x_r^2 - a) - (x_{r-1}^2 - a)} = \frac{x_r(x_r^2 - x_{r-1}^2) - (x_r^2 - a)(x_r - x_{r-1})}{x_r^2 - x_{r-1}^2},$$

kar se s krajšanjem izraza $x_r - x_{r-1}$ poenostavi v iskano formulo. Za dokaz konvergence zaporedja približkov najprej opazimo, da so pri pozitivnih začetnih približkih vsi členi zaporedja pozitivni. Ker je

$$x_{r+1} - x_r = \frac{x_{r-1}x_r + a}{x_{r-1} + x_r} - x_r = \frac{a - x_r^2}{x_{r-1} + x_r},$$

je približek x_{r+1} manjši od x_r, če je $x_r > \sqrt{a}$. Poleg tega je

$$x_{r+1} - \sqrt{a} = \frac{x_{r-1}x_r + a}{x_{r-1} + x_r} - \sqrt{a} = \frac{(x_{r-1} - \sqrt{a})(x_r - \sqrt{a})}{x_{r-1} + x_r},$$

iz česar lahko sklepamo, da je za poljubna začetna približka x_0 in x_1, ki sta večja od $\sqrt{a}$, zaporedje padajoče in navzdol omejeno s $\sqrt{a}$. Torej ima limito $\alpha \geq \sqrt{a}$, ki zadošča enačbi

$$\alpha = \frac{\alpha^2 + a}{\alpha + \alpha},$$

od kjer sledi $\alpha = \sqrt{a}$.

Exercise 2.16. Implement the secant method in Matlab and test it with the function f from Exercise 2.14 with the initial values $x_0 = 1$ and $x_1 = 1.1$. Compare the obtained results to the results of the Newton's method.

Solution. Funkcijo `sekantna`, ki izvede sekantno metodo, implementiramo na zelo podoben način kot funkcijo `iteracija` iz naloge 2.5. Pri tem pazimo, da funkcijsko vrednost za vsak približek izračunamo le enkrat, saj izvrednotenje funkcije predstavlja največji računski zalogaj pri izvedbi metode.

```
function [x,X,k] = sekantna(f,x0,x1,tol,N)
% funkcija
%  [x,X,k] = sekantna(f,x0,x1,tol,N)
% izvede sekantno metodo za iskanje ničle funkcije f
%
% vhodni podatki:
%  f         funkcija, ničlo katere iščemo,
%  x0, x1    začetna približka metode
%
% ostali vhodni in izhodni podatki so enaki kot pri
% funkciji 'iteracija'

X = [x0 x1];
k = 0;
fxk = f(X(1));
while k < N
    k = k+1;
    fxkn = f(X(k+1));
    X(k+2) = X(k+1) - fxkn*(X(k+1)-X(k))/(fxkn-fxk);
    fxk = fxkn;
    if abs(X(k+2)-X(k+1)) < tol
        break;
    end
end
x = X(k+2);

end
```

Približki, izračunani z izvedbo ukaza `[x,X,k] = sekantna(f,1,1.1,1e-15,100)`, so prikazani v tabeli 2.2. Približek x_8, dobljen na sedmem koraku, se z vrednostjo `fzero(f,0)` ujema v enajstih decimalkah, približek x_9 pa v vseh. Torej je število korakov pri sekantni metodi v tem primeru le za ena večje od števila korakov pri tangentni metodi. Pravzaprav je sekantna metoda učinkovitejša, saj smo na vsakem koraku opravili le en izračun funkcijske vrednosti namesto dveh pri tangentni metodi.

korak	približek	napaka	funkcijski izračuni
1	1.42632775255262	$1.3561 \cdot 10^{-1}$	3
2	1.24363560557646	$4.7083 \cdot 10^{-2}$	4
3	1.27978689949541	$1.0932 \cdot 10^{-2}$	5
4	1.29168083871343	$9.6242 \cdot 10^{-4}$	6
5	1.29069926729793	$1.9154 \cdot 10^{-5}$	7
6	1.29071838835441	$3.3362 \cdot 10^{-8}$	8
7	1.29071842171712	$1.1571 \cdot 10^{-12}$	9
8	1.29071842171596	0	10
9	1.29071842171596	0	11

Tabela 2.2: *Izvedba sekantne metode v nalogi 2.16.*

One of the generalizations of the Newton's method is the method (f, f', f''), which uses not only the first derivative but also the second derivative of the function f. The approximation x_{r+1} is computed based on x_r by the formula

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)} - \frac{f''(x_r)f(x_r)^2}{2f'(x_r)^3}.$$

In the literature this iteration is often called the Schrödinger method of second order. The methods of higher orders include higher order derivatives of f.

Exercise 2.17. Let f be an analytic function with a simple zero α. Use the Taylor expansion of the inverse of f to derive the method (f, f', f''). Compute the order of convergence of the iteration sequence in the neighborhood of α.

Solution. Ker je α enostavna ničla funkcije f, je $f'(\alpha) \neq 0$ in po izreku o inverzni funkciji obstaja $\delta > 0$, da je f na $(\alpha - \delta, \alpha + \delta)$ obrnljiva. Natančneje, obstaja $\varepsilon > 0$ in taka funkcija $F : (-\varepsilon, \varepsilon) \to (\alpha - \delta, \alpha + \delta)$, da je $F(f(x)) = x$ za vsak $x \in (\alpha - \delta, \alpha + \delta)$. Funkcijo F razvijemo v Taylorjevo vrsto okoli $y \in (-\varepsilon, \varepsilon)$:

$$F(z) = F(y) + F'(y)(z - y) + \frac{1}{2}F''(y)(z - y)^2 + \dots.$$

Vzemimo $z = 0$ in $y = f(x)$ za $x \in (\alpha - \delta, \alpha + \delta)$. Iz

$$F(f(x)) = x, \qquad F'(f(x))f'(x) = 1, \qquad F''(f(x))f'(x)^2 + F'(f(x))f''(x) = 0$$

izrazimo $F'(f(x)) = 1/f'(x)$ in $F''(f(x)) = -f''(x)/f'(x)^3$. Če v Taylorjevi vrsti zanemarimo člene s stopnjo, višjo od 2, dobimo

$$\alpha \approx x - \frac{f(x)}{f'(x)} - \frac{f''(x)f(x)^2}{2f'(x)^3},$$

kar določa iteracijsko funkcijo

$$g(x) = x - \frac{f(x)}{f'(x)} - \frac{f''(x)f(x)^2}{2f'(x)^3}.$$

Red konvergence metode določimo z odvajanjem g. Izračunamo

$$g'(x) = \frac{3f''(x)^2 - f'(x)f'''(x)}{2f'(x)^4} f(x)^2.$$

Od tod je razvidno, da je $g'(\alpha) = 0$ in $g''(\alpha) = 0$, medtem ko je vrednost $g'''(\alpha)$ v splošnem različna od nič, zato je red konvergence kubičen.

Another method for computing a zero of a function f based on a second derivative is the Halley's method. The approximation x_{r+1} is computed from x_r by the formula

$$x_{r+1} = x_r - \frac{2f(x_r)f'(x_r)}{2f'(x_r)^2 - f(x_r)f''(x_r)}.$$

The Halley's method belongs to the class of the Householder's methods

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

It is obtained for $d = 2$. The Newton's method corresponds to the choice $d = 1$.

Exercise 2.18. Let f be twice differentiable function. Verify that the Halley's method corresponds to the Newton's method for the function $F(x) = f(x)/\sqrt{|f'(x)|}$. What is the order of convergence of the method in the neighborhood of a simple zero of f assuming f is at least five times differentiable?

Solution. Izračunamo

$$F'(x) = \frac{2f'(x)^2 - f(x)f''(x)}{2f'(x)\sqrt{|f'(x)|}}$$

in izpeljemo iteracijsko funkcijo

$$g(x) = x - \frac{F(x)}{F'(x)} = x - \frac{2f(x)f'(x)}{2f'(x)^2 - f(x)f''(x)}.$$

Z odvajanjem iteracijske funkcije določimo še red metode. Z nekaj računanja dobimo, da je

$$g'(x) = \frac{3f''(x)^2 - 2f'(x)f'''(x)}{(f(x)f''(x) - 2f'(x)^2)^2} f(x)^2,$$

iz česar sledi, da je red konvergence iteracijskega zaporedja v okolici ničle funkcije f v splošnem kubičen.

Exercise 2.19. Simplify the Halley's method for the function f given by $f(x) = x^2 - a$, $a > 0$. The result of the iteration for a suitable initival value is an approximation for $\sqrt{a}$.

Solution. Prva odvoda funkcije $f(x) = x^2 - a$ sta $f'(x) = 2x$ in $f''(x) = 2$. Z nekaj računanja iteracijsko funkcijo g poenostavimo v

$$g(x) = x\frac{x^2 + 3a}{3x^2 + a}.$$

Zanjo smo v nalogi 2.7 že dokazali, da določa iteracijsko zaporedje, ki konvergira k $\sqrt{a}$ za vsak začetni približek $x_0 > 0$.

Exercise 2.20. In Matlab compare the Newton's method, the secant method, the method (f, f', f''), and the Halley's method for computing the zero of the function f from Exercise 2.14. Test the methods for the initial values from the array `1:0.1:10` (and $x_1 = x_0 + 0.1$ for the secant method). For each execution of the method find the smallest number k such that the approximation x_k absolutely differs from `fzero(f,1)` for less than 10^{-10}. Then, for each method plot the graph of k in dependence of the initial values and the graph of the number of function evaluations in dependence of the initial values. Comment the results.

Solution. Implementacija tangentne metode (funkcija `tangentna`) je opisana v nalogi 2.14, implementacija sekantne metode (funkcija `sekantna`) pa v nalogi 2.16. Funkciji `fdfddf` in `halley` za izvedbo metode (f, f', f'') in Halleyjeve metode implementiramo na podoben način, pomagamo si lahko s funkcijo `iteracija` iz naloge 2.5. Rezultati izvedbe ukazov `[x,X,k] = fdfddf(f,df,ddf,1,1e-15,100)` in `[x,X,k] = halley(f,df,ddf,1,1e-15,100)` ob primerno definiranih funkcijah `f`, `df`, `ddf`, ki določajo f, f', f'', so prikazani v tabelah 2.3 in 2.4.

korak	približek	napaka	funkcijski izračuni
1	1.02811573481699	$2.6260 \cdot 10^{-1}$	3
2	1.11691838286784	$1.7380 \cdot 10^{-1}$	6
3	1.25367570678920	$3.7043 \cdot 10^{-2}$	9
4	1.29047854328615	$2.3988 \cdot 10^{-4}$	12
5	1.29071842165685	$5.9109 \cdot 10^{-11}$	15
6	1.29071842171596	0	18
7	1.29071842171596	0	21

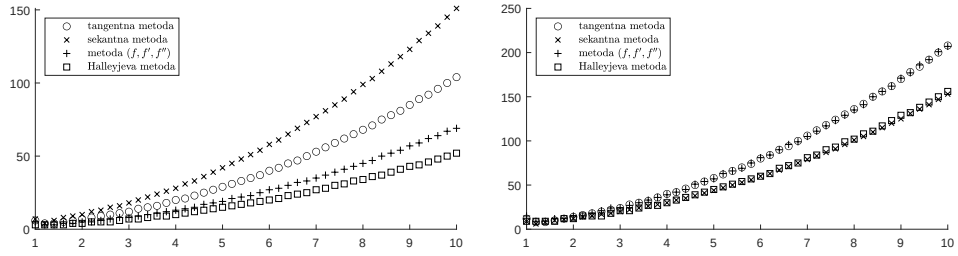
Tabela 2.3: *Izvedba metode (f, f', f'') v nalogi 2.20.*

korak	približek	napaka	funkcijski izračuni
1	1.26437572777572	$2.6343 \cdot 10^{-2}$	3
2	1.29070002729498	$1.8394 \cdot 10^{-5}$	6
3	1.29071842171596	$6.2172 \cdot 10^{-15}$	9
4	1.29071842171596	0	12
5	1.29071842171596	0	15

Tabela 2.4: *Izvedba Halleyjeve metode v nalogi 2.20.*

Za vsako metodo in vsak začetni približek x_0 iz seznama `1:0.1:10` poiščemo najmanjši k, da se približek x_k od vrednosti `x`, dobljene z vgrajeno funkcijo, absolutno razlikuje za manj kot `e = 1e-10`. Če je `X` seznam vseh izračunanih približkov (za katerega predpostavimo, da vsebuje približek x_k, ki zadošča kriteriju), lahko indeks k poiščemo z ukazom `find(abs(X - x) < e, 1)`. Ta namreč vrne indeks prvega elementa v seznamu `X`, ki se od `x` absolutno razlikuje za manj kot `e`. Ker je prvi element seznama `X` začetni približek x_0, moramo od rezultata odšteti 1 (oziroma 2 pri sekantni metodi).

```
f = @(x) x + 4 - exp(x^2);
df = @(x) 1 - 2*x*exp(x^2);
ddf = @(x) - 2*exp(x^2) - 4*x^2*exp(x^2);

x = fzero(f,1);
x0 = 1:0.1:10; tol = 1e-15; N = 200; e = 1e-10;

[k1,k2,k3,k4] = deal(zeros(size(x0)));
for i = 1:length(x0)
    [~,X1] = tangentna(f,df,x0(i),tol,N);
    k1(i) = find(abs(X1-x) < e, 1) - 1;

    [~,X2] = sekantna(f,x0(i),x0(i)+0.1,tol,N);
    k2(i) = find(abs(X2-x) < e, 1) - 2;

    [~,X3] = fdfddf(f,df,ddf,x0(i),tol,N);
    k3(i) = find(abs(X3-x) < e, 1) - 1;

    [~,X4] = halley(f,df,ddf,x0(i),tol,N);
    k4(i) = find(abs(X4-x) < e, 1) - 1;
end
```



(A) *Število korakov* (B) *Število funkcijskih izračunov*

Slika 2.3: *Primerjava metod za reševanje nelinearnih enačb na primeru iz naloge 2.20.*

Grafi, s katerimi primerjamo učinkovitost metod, so prikazani na sliki 2.3. Pri pripravi grafov števila funkcijskih izračunov upoštevamo, da pri tangentni metodi

na vsakem koraku izvrednotimo dve funkciji, pri sekantni eno, pri metodi (f, f', f'') in Halleyjevi metodi pa tri. Rezultati kažejo, da za izbrano funkcijo f najhitreje konvergira Halleyjeva metoda, najpočasneje pa sekantna metoda. Metoda (f, f', f'') konvergira hitreje od tangentne. Iz grafa funkcijskih izračunov pa je razvidno, da sta sekantna in Halleyjeva metoda učinkovitejši od tangentne metode in metode (f, f', f'').

3. Systems of Linear Equations

A system of $n \in \mathbb{N}$ linear equations with n unknowns can be presented in the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is a square matrix, $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of unknowns, and $\boldsymbol{b} \in \mathbb{R}^n$ is the vector of free terms. The solution of the system exists and is unique if and only if the matrix $\boldsymbol{A}$ is invertible. To compute $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ we usually use direct methods that avoid the computation of the matrix inverse. A particular method is chosen based on sensitivity and other special properties of the system matrix.

3.1. Matrix Norms and Sensitivity

A matrix norm $\|\cdot\|$ is a map assigning a real value to a square matrix. It is determined similarly as the vector norm, except that besides the three basic properties (positivity, homogenity, and triangular inequality) the fourth one (submultiplicativity) relating the norm of product with the product of norms is required. Namely, for every matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{C}^{n\times n}$ and every scalar value $\alpha \in \mathbb{C}$ it holds:

- $\|\boldsymbol{A}\| \geq 0$, and $\|\boldsymbol{A}\| = 0$ if and only if $\boldsymbol{A} = \boldsymbol{0}$ (positivity),
- $\|\alpha\boldsymbol{A}\| = |\alpha|\,\|\boldsymbol{A}\|$ (homogenity),
- $\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$ (triangular inequality),
- $\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\|\,\|\boldsymbol{B}\|$ (submultiplicativity).

Exercise 3.1. Let $\|\cdot\|$ be a matrix norm. Prove that the norm $\|\boldsymbol{I}\|$ of identity $\boldsymbol{I}$ is not smaller than 1 and argue that every invertible matrix $\boldsymbol{A}$ satisfies $\|\boldsymbol{A}^{-1}\| \geq \|\boldsymbol{A}\|^{-1}$.

Solution. Iz submultiplikativnosti norme sledi $\|\boldsymbol{I}\| \leq \|\boldsymbol{I}\|^2$ in ker je $\|\boldsymbol{I}\| > 0$, to dokazuje $\|\boldsymbol{I}\| \geq 1$. Za obrnljivo matriko $\boldsymbol{A}$ po submultiplikativnosti norme velja

$$\|\boldsymbol{I}\| = \left\|\boldsymbol{A}\boldsymbol{A}^{-1}\right\| \leq \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|,$$

kar potrjuje zvezo $\|\boldsymbol{A}^{-1}\| \geq \|\boldsymbol{A}\|^{-1}$.

Exercise 3.2. Let $\|\cdot\|$ be a matrix norm. Prove that every matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ satisfies $\rho(\boldsymbol{A}) \leq \|\boldsymbol{A}\|$, where $\rho(\boldsymbol{A})$ denotes the spectral radius of the matrix $\boldsymbol{A}$.

Solution. Spektralni radij $\rho(\boldsymbol{A})$ ustreza absolutni vrednosti dominantne lastne vrednosti λ matrike $\boldsymbol{A}$. Naj bo $\boldsymbol{x} \in \mathbb{C}^n$ pripadajoči lastni vektor in $\boldsymbol{X} \in \mathbb{C}^{n\times n}$ matrika, v kateri je vsak stolpec enak $\boldsymbol{x}$. Iz homogenosti in submultiplikativnosti matrične norme sledi

$$|\lambda|\,\|\boldsymbol{X}\| = \|\lambda\boldsymbol{X}\| = \|\boldsymbol{A}\boldsymbol{X}\| \leq \|\boldsymbol{A}\|\,\|\boldsymbol{X}\|\,.$$

Ker je $\boldsymbol{x} \neq \boldsymbol{0}$, je $\boldsymbol{X} \neq \boldsymbol{0}$ in zato $\|\boldsymbol{X}\| > 0$. Sledi $|\lambda| \leq \|\boldsymbol{A}\|$.

Often the matrix norms are defined by vector norms. A simple idea is to arrange the elements $a_{i,j}$, $i,j = 1,2,\ldots,n$, of a matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ into the vector

$$\mathrm{vec}(\boldsymbol{A}) = (a_{1,1}, a_{1,2}, \ldots, a_{1,n}, a_{2,1}, a_{2,2}, \ldots, a_{2,n}, \ldots, a_{n,1}, a_{n,2}, \ldots, a_{n,n})$$

and apply a vector p-norm $\|\cdot\|_p$, $p \geq 1$. This gives the map

$$N_p(\boldsymbol{A}) = \|\mathrm{vec}(\boldsymbol{A})\|_p$$

satisfying the first three properties of the matrix norm.

Exercise 3.3. Let $a_{i,j}$, $i,j = 0,1,\ldots,n$, denote the elements of a matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$, and let

$$N_\infty(\boldsymbol{A}) = \max_{i,j=1,\ldots,n} |a_{i,j}|.$$

1. Prove with example that N_∞ is not a matrix norm.
2. Argue that $\|\boldsymbol{A}\| = nN_\infty(\boldsymbol{A})$ is a matrix norm.

Solution.

1. Preslikava N_∞ ni matrična norma, saj submultiplikativnost ni zagotovljena; za

$$\boldsymbol{A} = \boldsymbol{B} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

   na primer velja $N_\infty(\boldsymbol{A}\boldsymbol{B}) = 2 > 1 = N_\infty(\boldsymbol{A})N_\infty(\boldsymbol{B})$.

2. Preverimo, da preslikava $\|\cdot\|$ izpolnjuje vse lastnosti iz definicije matrične norme. Naj bosta $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{C}^{n\times n}$ poljubni matriki z elementi $a_{i,j}$, $b_{i,j}$ ter naj bo $\alpha \in \mathbb{C}$ poljuben skalar.

   - Pozitivnost: Po definiciji absolutne vrednosti je

$$\|\boldsymbol{A}\| = n \max_{i,j} |a_{i,j}| \geq 0.$$

     Prav tako velja

$$\|\boldsymbol{A}\| = n \max_{i,j} |a_{i,j}| = 0 \quad \Leftrightarrow \quad a_{i,j} = 0 \;\; \forall\, i,j \quad \Leftrightarrow \quad \boldsymbol{A} = \boldsymbol{0}.$$

- Homogenost: Po pravilu za absolutno vrednost produkta je

$$\|\alpha \boldsymbol{A}\| = n \max_{i,j} |\alpha\, a_{i,j}| = n\ |\alpha| \max_{i,j} |a_{i,j}| = |\alpha|\, \|\boldsymbol{A}\|\,.$$

- Trikotniška neenakost: Iz trikotniške neenakosti za absolutno vrednost sledi

$$\|\boldsymbol{A}+\boldsymbol{B}\| = n \max_{i,j} |a_{i,j}+b_{i,j}| \le n \left(\max_{i,j} |a_{i,j}| + \max_{i,j} |b_{i,j}| \right) = \|\boldsymbol{A}\| + \|\boldsymbol{B}\|\,.$$

- Submultiplikativnost: Po trikotniški neenakosti in pravilu za absolutno vrednost produkta velja

$$\begin{aligned} \|\boldsymbol{A}\boldsymbol{B}\| = n \max_{i,j} \left| \sum_{k=1}^{n} a_{i,k} b_{k,j} \right| &\le n \max_{i,k} |a_{i,k}|\ \max_{j} \sum_{k=1}^{n} |b_{k,j}| \\ &\le n \max_{i,k} |a_{i,k}|\ \ n \max_{k,j} |b_{k,j}| = \|\boldsymbol{A}\|\, \|\boldsymbol{B}\|\,. \end{aligned}$$

Exercise 3.4. The Frobenius norm is derived from the Euclidean vector norm. For a matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ with the elements $a_{i,j}$, $i,j = 0,1,\ldots,n$, it is given by

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

1. Convince yourself that $\|\cdot\|_F$ is a matrix norm.
2. Prove that $\|\boldsymbol{A}\|_F^2 = \operatorname{trace}(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})$.

Solution.

1. Pozitivnost, homogenost in trikotniška neenakost sledijo iz lastnosti vektorske 2-norme. Preverimo submultiplikativnost. Ker za matriki $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{C}^{n\times n}$ z elementi $a_{i,j}$, $b_{i,j}$ po Cauchy–Schwarzevi neenakosti velja

$$\sum_{i,j=1}^{n} \left| \sum_{k=1}^{n} a_{i,k} b_{k,j} \right|^2 \le \sum_{i,j=1}^{n} \left(\sum_{k=1}^{n} |a_{i,k}|^2 \sum_{k=1}^{n} |b_{k,j}|^2 \right),$$

je

$$\|\boldsymbol{A}\boldsymbol{B}\|_F = \sqrt{\sum_{i,j=1}^{n} \left| \sum_{k=1}^{n} a_{i,k} b_{k,j} \right|^2} \le \sqrt{\sum_{i,k=1}^{n} |a_{i,k}|^2} \sqrt{\sum_{k,j=1}^{n} |b_{k,j}|^2} = \|\boldsymbol{A}\|_F\, \|\boldsymbol{B}\|_F\,.$$

2. Element matrike $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ na mestu (i,j) je enak $\sum_{k=1}^{n} \overline{a_{k,i}} a_{k,j}$, zato je

$$\operatorname{sled}(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}) = \sum_{i=1}^{n} \sum_{k=1}^{n} \overline{a_{k,i}} a_{k,i} = \sum_{i,k=1}^{n} |a_{k,i}|^2 = \|\boldsymbol{A}\|_F^2\,.$$

An important class of matrix norms are the operator norms, which are defined based on a chosen vector norm $\|\cdot\|$ for a matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ as

$$\|\boldsymbol{A}\| = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}.$$

The definition can be simplified, it holds $\|\boldsymbol{A}\| = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\|$.

Exercise 3.5. Let $\|\cdot\|$ be an operator norm. Prove that for an invertible matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ it holds

$$\left\|\boldsymbol{A}^{-1}\right\|^{-1} = \min_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}.$$

Solution. Ker je matrika $\boldsymbol{A}$ obrnljiva, za vsak $\boldsymbol{x} \in \mathbb{C}^n \setminus \{\boldsymbol{0}\}$ obstaja $\boldsymbol{y} \in \mathbb{C}^n \setminus \{\boldsymbol{0}\}$, da je $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{y}$. Torej po definiciji operatorske norme velja

$$\left\|\boldsymbol{A}^{-1}\right\| = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\left\|\boldsymbol{A}^{-1}\boldsymbol{x}\right\|}{\|\boldsymbol{x}\|} = \max_{\boldsymbol{A}\boldsymbol{y}\neq\boldsymbol{0}} \frac{\|\boldsymbol{y}\|}{\|\boldsymbol{A}\boldsymbol{y}\|}$$

oziroma, ker je množica $\{\boldsymbol{A}\boldsymbol{y};\, \boldsymbol{y} \in \mathbb{C}^n\} \setminus \{\boldsymbol{0}\}$ enaka množici $\mathbb{C}^n \setminus \{\boldsymbol{0}\}$,

$$\left\|\boldsymbol{A}^{-1}\right\| = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|} = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \left(\frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}\right)^{-1}.$$

Slednje po zvezi med maksimumom in minimumom dokazuje trditev.

The matrix p-norms $\|\cdot\|_p$, $p \geq 1$, are examples of the operator norms induced by the vector p-norms. Especially simple are the 1-norm and the ∞-norm satisfying

$$\|\boldsymbol{A}\|_1 = \max_{j=1,\ldots,n} \sum_{i=1}^{n} |a_{i,j}|\,, \quad \|\boldsymbol{A}\|_\infty = \max_{i=1,\ldots,n} \sum_{j=1}^{n} |a_{i,j}|\,,$$

which means that they can be determined with summation of absolute values of the elements $a_{i,j}$ of the matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$.

Exercise 3.6. Prove than for any matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ it holds $\|\boldsymbol{A}\|_1 = \|\boldsymbol{A}^{\mathsf{H}}\|_\infty$.

Solution. Iz lastnosti 1-norme in ∞-norme sledi

$$\left\|\boldsymbol{A}^{\mathsf{H}}\right\|_\infty = \max_{i=1,\ldots,n} \sum_{j=0}^{n} |\overline{a_{j,i}}| = \max_{j=1,\ldots,n} \sum_{i=0}^{n} |a_{i,j}| = \|\boldsymbol{A}\|_1\,.$$

The operator 2-norm is one of the most standard matrix norms. It is also called the spectral norm since for a matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ it holds

$$\|\boldsymbol{A}\|_2 = \max_{i=1,\dots,n} \sqrt{\lambda_i(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})},$$

where $\lambda_i(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})$, $i = 1, 2, \dots, n$, denote the eigenvalues of the matrix $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$. Since the matrix $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ is Germitian and non-negative definite, the eigenvalues are non-negative. The values $\sqrt{\lambda_i(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})}$ are called the singular values of $\boldsymbol{A}$. The computation of the 2-norm is usually demanding, but sometimes it is sufficient to estimate it based on computationally less expensive norms.

Exercise 3.7. Prove that for any matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ it holds $\|\boldsymbol{A}\|_2 = \|\boldsymbol{A}^{\mathsf{H}}\|_2$.

Solution. Naj bo $i \in \{1, 2, \dots, n\}$ in naj bo $\boldsymbol{x}_i$ lastni vektor, ki pripada lastni vrednosti $\lambda_i(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})$. Potem je

$$\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{x}_i = \boldsymbol{A}\,\lambda_i(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})\,\boldsymbol{x}_i = \lambda_i(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})\,\boldsymbol{A}\boldsymbol{x}_i,$$

kar dokazuje, da je $\boldsymbol{A}\boldsymbol{x}_i$ lastni vektor, $\lambda_i(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})$ pa lastna vrednost matrike $\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}}$. Podoben sklep velja tudi v obratno smer. Torej imata matriki $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ in $\boldsymbol{A}\boldsymbol{A}^{\mathsf{H}}$ enak nabor lastnih vrednosti.

Exercise 3.8. Prove that for the spectral norm $\|\boldsymbol{A}\|_2$ of the matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ the following estimates hold.

1. $\frac{1}{\sqrt{n}}\|\boldsymbol{A}\|_F \le \|\boldsymbol{A}\|_2 \le \|\boldsymbol{A}\|_F$
2. $\frac{1}{\sqrt{n}}\|\boldsymbol{A}\|_\infty \le \|\boldsymbol{A}\|_2 \le \sqrt{n}\,\|\boldsymbol{A}\|_\infty$
3. $\frac{1}{\sqrt{n}}\|\boldsymbol{A}\|_1 \le \|\boldsymbol{A}\|_2 \le \sqrt{n}\,\|\boldsymbol{A}\|_1$
4. $N_\infty(\boldsymbol{A}) \le \|\boldsymbol{A}\|_2 \le nN_\infty(\boldsymbol{A})$
5. $\|\boldsymbol{A}\|_2 \le \sqrt{\|\boldsymbol{A}\|_1\,\|\boldsymbol{A}\|_\infty}$

Solution.

1. Upoštevamo ugotovitev iz naloge 3.4, da je $\|\boldsymbol{A}\|_F^2 = \mathrm{sled}(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A})$. Obe neenakosti dokažemo z uporabo dejstev, da je sled matrike enaka vsoti njenih lastnih vrednosti in da so lastne vrednosti matrike $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}$ nenegativne.

2. Enostavno je preveriti, da za vsak vektor $\boldsymbol{x} \in \mathbb{C}^n$ velja

$$\|\boldsymbol{x}\|_\infty \le \|\boldsymbol{x}\|_2 \le \sqrt{n}\,\|\boldsymbol{x}\|_\infty .$$

Od tod sledi

$$\|\boldsymbol{A}\|_2 = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \le \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\sqrt{n}\,\|\boldsymbol{A}\boldsymbol{x}\|_\infty}{\|\boldsymbol{x}\|_\infty} = \sqrt{n}\,\|\boldsymbol{A}\|_\infty ,$$

$$\|\boldsymbol{A}\|_2 = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \ge \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_\infty}{\sqrt{n}\,\|\boldsymbol{x}\|_\infty} = \frac{1}{\sqrt{n}}\,\|\boldsymbol{A}\|_\infty .$$

3. Oceni lahko dokažemo s podobnim sklepanjem kot v prejšnji točki, saj za vsak vektor $\boldsymbol{x} \in \mathbb{C}^n$ velja
$$\frac{1}{\sqrt{n}} \|\boldsymbol{x}\|_1 \leq \|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_1 .$$
Lahko pa se poslužimo kar zveze, dokazane v prejšnji točki, za matriko $\boldsymbol{A}^{\mathsf{H}}$ in neenakosti utemeljimo na podlagi nalog 3.6 in 3.7.

4. Element $a_{i,j}$ matrike $\boldsymbol{A}$ na mestu (i,j) lahko zapišemo kot $\boldsymbol{e}_i^{\mathsf{T}} \boldsymbol{A} \boldsymbol{e}_j$, kjer sta $\boldsymbol{e}_i$ in $\boldsymbol{e}_j$ standardna enotska vektorja. Po Cauchy–Schwarzevi neenakosti in definiciji $\|\cdot\|_2$ je
$$|a_{i,j}| = \left|\boldsymbol{e}_i^{\mathsf{T}} \boldsymbol{A} \boldsymbol{e}_j\right| \leq \|\boldsymbol{e}_i\|_2 \, \|\boldsymbol{A}\boldsymbol{e}_j\|_2 \leq \|\boldsymbol{A}\|_2 ,$$
iz česar sledi $N_\infty(\boldsymbol{A}) \leq \|\boldsymbol{A}\|_2$. Neenakost $\|\boldsymbol{A}\|_2 \leq n N_\infty(\boldsymbol{A})$ lahko dokažemo s pomočjo ocene v prvi točki naloge, če sklepamo
$$\|\boldsymbol{A}\|_2 \leq \|\boldsymbol{A}\|_F \leq \sqrt{\sum_{i,j=1}^{n} |a_{i,j}|^2} \leq \sqrt{n^2 \max_{i,j} |a_{i,j}|^2} = n \max_{i,j} |a_{i,j}| = n N_\infty(\boldsymbol{A}).$$

5. Po nalogi 3.2 je spektralni radij matrike omejen z normo matrike, zato je
$$\|\boldsymbol{A}\|_2^2 = \max_{i=1,\ldots,n} \lambda_i(\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}) \leq \left\|\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\right\|_\infty .$$
Zaradi submultiplikativnosti in dejstva, dokazanega v nalogi 3.6, je
$$\left\|\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A}\right\|_\infty \leq \left\|\boldsymbol{A}^{\mathsf{H}}\right\|_\infty \|\boldsymbol{A}\|_\infty = \|\boldsymbol{A}\|_1 \, \|\boldsymbol{A}\|_\infty$$
in ocena sledi.

Exercise 3.9. Use the results of Exercise 3.8 to find the best possible estimate for the spectral norm of the matrix $\boldsymbol{A}$ with the elements
$$a_{i,j} = \begin{cases} -2i; & i = j \\ n - i; & j = i + 1 \\ n - j; & i = j + 1 \\ 0; & \text{otherwise} \end{cases}, \qquad i, j = 1, 2, \ldots, n.$$
Use Matlab built-in function `norm` to verify the derived estimates for $n = 1000$ and compute $\|\boldsymbol{A}\|_2$. Compare the times needed for the computations of different norms.

Solution. Najprej izračunamo
$$\|\boldsymbol{A}\|_1 = \max_{j=1,\ldots,n} (|-2j| + |n - j + 1| + |n - j|) = 2n + 1.$$
Ker je $\boldsymbol{A}$ simetrična, je tudi $\|\boldsymbol{A}\|_\infty = 2n+1$. Po absolutni vrednosti največji element matrike $\boldsymbol{A}$ je $-2n$, zato je $N_\infty(\boldsymbol{A}) = 2n$. Nadalje,
$$\|\boldsymbol{A}\|_F^2 = \sum_{i=1}^{n} (-2i)^2 + 2 \sum_{i=1}^{n-1} i^2 = 6 \sum_{i=1}^{n-1} i^2 + 4n^2,$$

od kjer po formuli za piramidna števila sledi

$$\|\boldsymbol{A}\|_F = \sqrt{(n-1)n(2n-1) + 4n^2} = \sqrt{2n^3 + n^2 + 1}.$$

Na podlagi izpeljanih ocen je najboljša spodnja meja za $\|\boldsymbol{A}\|_2$ enaka

$$\max\left\{\sqrt{\frac{2n^3+n^2+1}{n}}, \frac{2n+1}{\sqrt{n}}, 2n\right\},$$

najboljša zgornja meja pa

$$\min\left\{\sqrt{2n^3+n^2+1}, (2n+1)\sqrt{n}, 2n^2, 2n+1\right\}.$$

To poenostavimo v oceno

$$2n \leq \|\boldsymbol{A}\|_2 \leq 2n+1.$$

V Matlabu lahko matriko $\boldsymbol{A}$ zgeneriramo z uporabo ukazov `diag`. Norme matrik računamo s funkcijo `norm`, ki ji kot prvi vhodni podatek podamo matriko, z drugim pa določimo, katero normo računamo.

```
n = 1000;
A = diag(n-1:-1:1,1) - 2*diag(1:n) + diag(n-1:-1:1,-1);
norm(A,1)        % 2001
norm(A,Inf)      % 2001
norm(A,'fro')    % približno 4473.2550
norm(A(:),Inf)   % 2000
norm(A,2)        % približno 2000.9986
```

Z merjenjem časa izvajanja ukazov se lahko prepričamo, da je računanje 2-norme veliko počasnejše od računanja ostalih norm. Če tridiagonalno matriko $\boldsymbol{A}$ podamo v razpršeni obliki (ukaz `spdiags([n-(1:n)' -2*(1:n)' n-(0:n-1)'], [-1 0 1], n, n)`), spektralne norme ne moremo izračunati z ukazom `norm`. Namesto tega lahko uporabimo funkcijo `normest`, a ta vrne le (včasih nezanesljiv) približek za 2-normo.

Exercise 3.10. Let $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ be a matrix with the elements $a_{i,j}$, and let $|\boldsymbol{A}| \in \mathbb{R}^{n\times n}$ denote the matrix with the elements $|a_{i,j}|$.

1. Prove that $\|\boldsymbol{A}\|_2 \leq \||\boldsymbol{A}|\|_2$.
2. Find an example showing that in general it does not hold $\|\boldsymbol{A}\|_2 = \||\boldsymbol{A}|\|_2$.

Solution.

1. Naj bo $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{C}^n$ tak enotski vektor, da velja $\|\boldsymbol{A}\|_2 = \|\boldsymbol{A}\boldsymbol{x}\|_2$. Opazimo, da je

$$\|\boldsymbol{A}\boldsymbol{x}\|_2^2 = \sum_{i=1}^{n}\left|\sum_{j=1}^{n} a_{i,j}x_j\right|^2 \leq \sum_{i=1}^{n}\left(\sum_{j=1}^{n}|a_{i,j}|\,|x_j|\right)^2 = \||\boldsymbol{A}|\,|\boldsymbol{x}|\|_2^2,$$

kjer je $|\boldsymbol{x}| = (|x_1|, |x_2|, \ldots, |x_n|) \in \mathbb{R}^n$ enotski vektor. Zato po definiciji matrične 2-norme velja $\|\boldsymbol{A}\|_2 \leq \||\boldsymbol{A}|\|_2$.

2. Poskusimo poiskati simetrično matriko $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$, za katero velja $\|\boldsymbol{A}\|_2 < \||\boldsymbol{A}|\|_2$. Pišimo
$$\boldsymbol{A} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \quad a, b, c \in \mathbb{R},$$
in obravnavajmo normi $\|\boldsymbol{A}\|_2$ in $\||\boldsymbol{A}|\|_2$ na podlagi lastnosti, da 2-norma matrike ustreza največji singularni vrednosti matrike. Ker je
$$\boldsymbol{A}^\mathsf{T}\boldsymbol{A} = \begin{bmatrix} a^2+b^2 & ab+bc \\ ba+cb & b^2+c^2 \end{bmatrix}, \quad |\boldsymbol{A}|^\mathsf{T}|\boldsymbol{A}| = \begin{bmatrix} a^2+b^2 & |ab|+|bc| \\ |ba|+|cb| & b^2+c^2 \end{bmatrix},$$
je matrika $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ pri izbiri $c = -a$ diagonalna in velja $\|\boldsymbol{A}\|_2 = \sqrt{a^2+b^2}$. Po drugi strani v tem primeru $\||\boldsymbol{A}|\|_2$ ustreza korenu večje izmed rešitev λ kvadratne enačbe
$$\left(a^2+b^2-\lambda\right)^2 - (2|ab|)^2 = 0,$$
torej
$$\||\boldsymbol{A}|\|_2 = \sqrt{a^2+b^2+2|ab|}.$$
Če vzamemo na primer $a = 1$, $b = 1$, $c = -1$, dobimo matriko $\boldsymbol{A}$, za katero velja $\|\boldsymbol{A}\|_2 = \sqrt{2} < 2 = \||\boldsymbol{A}|\|_2$.

The condition number $\kappa(\boldsymbol{A}) = \|\boldsymbol{A}^{-1}\|\,\|\boldsymbol{A}\|$ of an invertible matrix $\boldsymbol{A}$ is a measure that tells us how sensitive is the solving of the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. For a numerically computed approximation $\widehat{\boldsymbol{x}}$ of the exact solution $\boldsymbol{x}$ it holds
$$\frac{\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} = \mathcal{O}(\kappa(\boldsymbol{A})u),$$
where u is the unit roundoff. Usually we are interested in the spectral condition number $\kappa_2(\boldsymbol{A})$ obtained by choosing the 2-norm in the definition of $\kappa(\boldsymbol{A})$, which is equal to the quotient of the largest and smallest singular value of $\boldsymbol{A}$.

Exercise 3.11. The polynomial
$$p(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_0$$
of degree $n \in \mathbb{N}_0$ with the real coefficients $a_0, a_1, \ldots, a_n$, that agrees with a chosen function f in pairwise different points $x_0, x_1, \ldots, x_n$ is determined by the solution of the system of linear equations $\boldsymbol{V} \cdot \boldsymbol{a} = \boldsymbol{f}$, where
$$\boldsymbol{V} = \left[x_i^j\right]_{i,j=0}^n = \begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix}$$

denotes the Vandermond matrix, $\boldsymbol{a} = (a_0, a_1, \ldots, a_n)$ represents the vector of wanted coefficients of the polynomial p, and the vector $\boldsymbol{f} = (f(x_0), f(x_1), \ldots, f(x_n))$ consists of the function values at points $x_0, x_1, \ldots, x_n$.

1. Comment the existence of the polynomial p.

2. In Matlab compute the spectral condition number of the Vandermond matrices determined by the points $x_k = k/n$, $k = 0, 1, \ldots, n$, for $n \in \{1, 2, \ldots, 10\}$.

3. Let
$$f(x) = x^5 + x^4 + x^3 + x^2 + x + 1.$$
The polynomial p of degree 5 that agrees with f in the points $k/5$, $k = 0, 1, \ldots, 5$, is the function f itself since it is a polynomial of degree 5. Verify that by solving the above system of equations in Matlab we do not obtain exactly the same result.

Solution.

1. Znano je, da je
$$\det(\boldsymbol{V}) = \prod_{0 \le j < k \le n} (x_k - x_j).$$
Ker so točke $x_0, x_1, \ldots, x_n$ paroma različne, je $\det(\boldsymbol{V}) \neq 0$ in sistem, ki določa p, ima enolično rešitev.

2. Vandermondovo matriko lahko v Matlabu zgeneriramo z ukazom `vander`, vendar, pozor, rezultat klica je matrika z elementi $[x_i^{n-j}]_{i,j=0}^{n}$. Za računanje (spektralne) občutljivosti matrike uporabimo vgrajeno funkcijo `cond`. Ugotovimo, da z večanjem n občutljivost hitro raste. Pri $n = 2$ je približno 2.6, pri $n = 5$ že približno $5.8 \cdot 10^4$, pri $n = 10$ pa kar $4.5 \cdot 10^{12}$.

3. Rešujemo sistem z Vandermondovo matriko velikosti 6×6. Uporabimo vgrajeni operator \. Rezultat bi moral biti vektor samih enic, vendar se v drugi normi od njega razlikuje za približno $2.3386 \cdot 10^{-13}$, kar ni zanemarljiva napaka.

```
X = (0:5)'/5;
V = fliplr(vander(X));
f = polyval(ones(1,6),X);
a = V\f;
norm(ones(6,1)-a)    % približno 2.3386 * 1e-13
```

Exercise 3.12. Prove that the spectral condition number of the Vandermond matrix $\boldsymbol{V}$ given in Exercise 3.11 equals 1 if the points $x_0, x_1, \ldots, x_n$ correspond to the roots of unity, i.e.,
$$x_k = e^{2\pi k i/(n+1)}, \quad k = 0, 1, \ldots, n.$$

Solution. Označimo $\omega = e^{2\pi i/(n+1)}$. Točka x_k ustreza vrednosti ω^k, zato je

$$\boldsymbol{V} = \begin{bmatrix} 1 & \omega^0 & \omega^0 & \dots & \omega^0 \\ 1 & \omega^1 & \omega^2 & \dots & \omega^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^n & \omega^{2n} & \dots & \omega^{n^2} \end{bmatrix}.$$

Oglejmo si produkt $\boldsymbol{V}$ in $\boldsymbol{V}^{\mathsf{H}}$. Diagonalni element $(\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}})_{j,j}$, $j = 0, 1, \dots, n$, je enak

$$\sum_{k=0}^{n} \omega^{jk}\,\overline{\omega^{jk}} = \sum_{k=0}^{n} (\omega\,\overline{\omega})^{jk} = n+1,$$

saj je $\overline{\omega} = \omega^{-1}$, medtem ko za vsak izvendiagonalni element velja

$$\sum_{k=0}^{n} \omega^k = \frac{1-\omega^{n+1}}{1-\omega} = 0,$$

saj je $\omega^{n+1} = 1$. Potemtakem je $\boldsymbol{V}\boldsymbol{V}^{\mathsf{H}} = (n+1)\boldsymbol{I}$ in $\kappa_2(\boldsymbol{V}) = 1$.

3.2. LU Decomposition

The basic numerical method for solving a system of linear equations is based on the LU decomposition. Every matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ with invertible leading principal submatrices can be uniquely expressed as $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$, where $\boldsymbol{L} \in \mathbb{R}^{n\times n}$ is a lower triangular matrix with ones on the diagonal and $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ an upper triangular matrix with non-zero elements on the diagonal.

The LU decomposition is computed in $n-1$ steps by transforming the initial matrix $\boldsymbol{A}^{(0)} = \boldsymbol{A}$ and in step $j \in \{1, 2, \dots, n-1\}$ substituting it with the matrix $\boldsymbol{A}^{(j)} = \boldsymbol{L}_j \boldsymbol{A}^{(j-1)}$, which is obtained with the elementary elimination matrix $\boldsymbol{L}_j = \boldsymbol{I} - \boldsymbol{l}_j \boldsymbol{e}_j^{\mathsf{T}}$. Here $\boldsymbol{e}_j \in \mathbb{R}^n$ denotes the standard unit vector and the vector

$$\boldsymbol{l}_j = \begin{bmatrix} 0 & \dots & 0 & \frac{\alpha_{j+1}}{\alpha_j} & \dots & \frac{\alpha_n}{\alpha_j} \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^n$$

is determined by elements α_i, $i = 1, 2, \dots, n$, of the column j of $\boldsymbol{A}^{(j-1)}$. Hence

$$\boldsymbol{L}_j \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_n \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_j & 0 & \dots & 0 \end{bmatrix}^{\mathsf{T}},$$

implying that $\boldsymbol{A}^{(n-1)}$ is an upper triangular matrix corresponding to $\boldsymbol{U}$. Moreover, since $\boldsymbol{L}_j^{-1} = \boldsymbol{I} + \boldsymbol{l}_j \boldsymbol{e}_j^{\mathsf{T}}$ and

$$\boldsymbol{L}_1^{-1}\boldsymbol{L}_2^{-1}\cdots\boldsymbol{L}_{n-1}^{-1} = \boldsymbol{I} + \boldsymbol{l}_1\boldsymbol{e}_1^{\mathsf{T}} + \boldsymbol{l}_2\boldsymbol{e}_2^{\mathsf{T}} + \ldots + \boldsymbol{l}_{n-1}\boldsymbol{e}_{n-1}^{\mathsf{T}},$$

the inverse of $\boldsymbol{L}_{n-1}\cdots\boldsymbol{L}_2\boldsymbol{L}_1$ is a lower triangular matrix, which corresponds to the matrix $\boldsymbol{L}$. With this procedure the LU decomposition is computed in $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ basic arithmetic operations.

Exercise 3.13. Given is the matrix

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & 3 & -4 \\ -4 & -1 & -4 & 7 \\ 2 & 3 & 5 & -3 \\ -2 & -2 & -7 & 9 \end{bmatrix}.$$

Compute the LU decomposition of $\boldsymbol{A}$.

Solution. Matriko $\boldsymbol{A}$ s pomočjo treh elementarnih eliminacij $\boldsymbol{L}_1$, $\boldsymbol{L}_2$, $\boldsymbol{L}_3$ preoblikujemo v zgornje trikotno matriko $\boldsymbol{U}$. Ker je matrika $\boldsymbol{L} = (\boldsymbol{L}_3\boldsymbol{L}_2\boldsymbol{L}_1)^{-1}$ spodnje trikotna in ima na diagonali enice, lahko vse bistvene elemente matrik $\boldsymbol{L}$ in $\boldsymbol{U}$ hranimo v matriki enake velikosti, kot je $\boldsymbol{A}$. Postopek je povzet v naslednjih korakih.

1. korak:

$$\begin{bmatrix} 2 & 1 & 3 & -4 \\ -4 & -1 & -4 & 7 \\ 2 & 3 & 5 & -3 \\ -2 & -2 & -7 & 9 \end{bmatrix} \overset{\boldsymbol{L}_1}{\sim} \begin{bmatrix} 2 & 1 & 3 & -4 \\ -2 & 1 & 2 & -1 \\ 1 & 2 & 2 & 1 \\ -1 & -1 & -4 & 5 \end{bmatrix}, \quad \boldsymbol{L}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

2. korak:

$$\begin{bmatrix} 2 & 1 & 3 & -4 \\ -2 & 1 & 2 & -1 \\ 1 & 2 & 2 & 1 \\ -1 & -1 & -4 & 5 \end{bmatrix} \overset{\boldsymbol{L}_2}{\sim} \begin{bmatrix} 2 & 1 & 3 & -4 \\ -2 & 1 & 2 & -1 \\ 1 & 2 & -2 & 3 \\ -1 & -1 & -2 & 4 \end{bmatrix}, \quad \boldsymbol{L}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -2 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

3. korak:

$$\begin{bmatrix} 2 & 1 & 3 & -4 \\ -2 & 1 & 2 & -1 \\ 1 & 2 & -2 & 3 \\ -1 & -1 & -2 & 4 \end{bmatrix} \overset{\boldsymbol{L}_3}{\sim} \begin{bmatrix} 2 & 1 & 3 & -4 \\ -2 & 1 & 2 & -1 \\ 1 & 2 & -2 & 3 \\ -1 & -1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{L}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Iskani matriki $\boldsymbol{L}$ in $\boldsymbol{U}$ sta dani z

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

Exercise 3.14. In Matlab prepare a function that computes the LU decomposition of the given matrix. Test the function on the matrices

$$\boldsymbol{A}_1 = \begin{bmatrix} 2 & -1 & 1 & 4 \\ 4 & -3 & 7 & 14 \\ 0 & -3 & 18 & 19 \\ 6 & -2 & 7 & 14 \end{bmatrix}, \qquad \boldsymbol{A}_2 = \begin{bmatrix} 2 & -2 & -4 & 4 \\ 4 & -3 & -6 & 14 \\ -8 & 3 & 6 & 19 \\ 6 & -2 & 7 & 14 \end{bmatrix}.$$

Solution. LU razcep $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ matrike $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ opravimo v $n-1$ korakih. Pri tem sproti

- dopolnjujemo matriko $\boldsymbol{L}$, ki jo na začetku z ukazom `eye` nastavimo na identično matriko, in
- spreminjamo matriko $\boldsymbol{A}$, ki jo na koncu z ukazom `triu` oklestimo v zgornjo trikotno matriko $\boldsymbol{U}$.

V koraku $j \in \{1, 2, \ldots, n-1\}$ popravimo vrstice matrike $\boldsymbol{A}$ z indeksi i od $j+1$ do n. Element $\boldsymbol{A}(i,j)$ delimo z diagonalnim elementom $\boldsymbol{A}(j,j)$ in dobimo $\boldsymbol{L}(i,j)$. Elemente v vrstici i matrike $\boldsymbol{A}$, ki se nahajajo v stolpcih z indeksi k od $j+1$ do n (to je desno od elementa $\boldsymbol{A}(i,j)$), popravimo tako, da od njih odštejemo produkt $\boldsymbol{L}(i,j)$ in $\boldsymbol{A}(j,k)$.

```
n = size(A,1);
L = eye(n);
for j = 1:n-1
    for i = j+1:n
        L(i,j) = A(i,j)/A(j,j);
        for k = j+1:n
            A(i,k) = A(i,k) - L(i,j)*A(j,k);
        end
    end
end
U = triu(A);
```

Pri izvedbi zgornjega postopka na matriki $\boldsymbol{A}_1$ dobimo LU razcep $\boldsymbol{A}_1 = \boldsymbol{L}_1\boldsymbol{U}_1$, ki je določen z matrikama

$$\boldsymbol{L}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 3 & -1 & 3 & 1 \end{bmatrix}, \qquad \boldsymbol{U}_1 = \begin{bmatrix} 2 & -1 & 1 & 4 \\ 0 & -1 & 5 & 6 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 5 \end{bmatrix}.$$

Pri računanju LU razcepa $\boldsymbol{A}_2 = \boldsymbol{L}_2\boldsymbol{U}_2$ matrike $\boldsymbol{A}_2$ pa pride do zapleta, saj dobimo

$$\boldsymbol{L}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ -4 & -5 & 1 & 0 \\ 3 & 4 & \infty & 1 \end{bmatrix}, \qquad \boldsymbol{U}_2 = \begin{bmatrix} 2 & -2 & -4 & 4 \\ 0 & 1 & 2 & 6 \\ 0 & 0 & 0 & 65 \\ 0 & 0 & 0 & -\infty \end{bmatrix}.$$

Težava je, da smo v predzadnjem koraku na diagonali dobili ničlo, po deljenju z njo pa v matriki $\boldsymbol{L}$ element ∞. Očitno torej LU razcep matrike $\boldsymbol{A}_2$ ne obstaja.

Exercise 3.15. Find such lower triangular matrix $\boldsymbol{L}$ and upper triangular matrix $\boldsymbol{U}$ that for the non-invertible matrix

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

it holds $\boldsymbol{A} = \boldsymbol{LU}$. Is the decomposition unique? Why does this not contradict the claim on the existence and uniqueness of the LU decomposition?

Solution. Po običajnem postopku LU razcepa dobimo

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \qquad \boldsymbol{U} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

V matriki $\boldsymbol{L}$ lahko element na mestu $(3,3)$ spremenimo na poljubno vrednost pa produkt $\boldsymbol{L}$ in $\boldsymbol{U}$ ostane enak $\boldsymbol{A}$. To ni v nasprotju s trditvijo o obstoju in enoličnosti LU razcepa, saj je element matrike $\boldsymbol{U}$ na mestu $(3,3)$ enak 0 in $\boldsymbol{U}$ ne ustreza lastnostim zgornje trikotne matrike iz LU razcepa. Prav tako $\boldsymbol{L}$ ustreza lastnostim spodnje trikotne matrike iz LU razcepa le, če je element na mestu $(3,3)$ enak 1.

Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a matrix and $\boldsymbol{b} \in \mathbb{R}^n$ a vector that determine the system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$. Suppose there exists the LU decomposition $\boldsymbol{A} = \boldsymbol{LU}$ of $\boldsymbol{A}$. Then $\boldsymbol{L}(\boldsymbol{Ux}) = \boldsymbol{b}$ and the solution $\boldsymbol{x} \in \mathbb{R}^n$ can be computed by solving the systems $\boldsymbol{Ly} = \boldsymbol{b}$ and $\boldsymbol{Ux} = \boldsymbol{y}$. Since both systems are based on a triangular matrix, computing their solutions is simple and requires $2n^2 + n$ operations.

Exercise 3.16. Let $\boldsymbol{A} \in \mathbb{R}^{4\times 4}$ be the matrix from Exercise 3.13 and

$$\boldsymbol{b} = \begin{bmatrix} 8 & -14 & 7 & -16 \end{bmatrix}^{\mathsf{T}}.$$

Use the LU decomposition of $\boldsymbol{A}$ to solve the system $\boldsymbol{Ax} = \boldsymbol{b}$.

Solution. Sistem $\boldsymbol{Ax} = \boldsymbol{b}$ rešimo z reševanjem sistemov $\boldsymbol{Ly} = \boldsymbol{b}$ in $\boldsymbol{Ux} = \boldsymbol{y}$ za matriki $\boldsymbol{L}$ in $\boldsymbol{U}$, podani v rešitvi naloge 3.13. Prvi sistem rešimo s premo substitucijo (od zgoraj navzdol):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ -1 & -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 8 \\ -14 \\ 7 \\ -16 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \\ -5 \\ -1 \end{bmatrix};$$

drugega pa z obratno substitucijo (od spodaj navzgor):

$$\begin{bmatrix} 2 & 1 & 3 & -4 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & -2 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \\ -5 \\ -1 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}.$$

The system $\boldsymbol{Ax} = \boldsymbol{b}$ has a unique solution if and only if the matrix $\boldsymbol{A}$ is invertible. To find the solution of the system also when the leading principal submatrices of $\boldsymbol{A}$ are not all invertible, we compute the LU decomposition by pivoting. In partial pivoting we ensure before performing the elementary elimination that the maximal absolute element on or below the diagonal of the current column is positioned on the diagonal of the matrix. This is achieved by the exchange of two rows, which is called pivoting and which is reflected by the permutation matrix $\boldsymbol{P}$ that appears in the final decomposition. The LU decomposition of $\boldsymbol{A}$ with partial pivoting has the form $\boldsymbol{PA} = \boldsymbol{LU}$, where $\boldsymbol{L}$ and $\boldsymbol{U}$ are matrices with the same properties as in the standard LU decomposition.

Exercise 3.17. Let

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 & -2 & 1 \\ 2 & 1 & -4 & 2 \\ 3 & -2 & 3 & -1 \\ -1 & 3 & -1 & 1 \end{bmatrix}.$$

Find the LU decomposition of the matrix $\boldsymbol{A}$ with partial pivoting and use it to compute the determinant of $\boldsymbol{A}$.

Solution. Razcep lahko shematično predstavimo podobno kot običajni LU razcep v nalogi 3.13, le da tu po potrebi na vsakem koraku postopka pivotiramo.

1. korak:

$$\begin{bmatrix} 2 & 1 & -2 & 1 \\ 2 & 1 & -4 & 2 \\ 3 & -2 & 3 & -1 \\ -1 & 3 & -1 & 1 \end{bmatrix} \underset{1\leftrightarrow 3}{\overset{\boldsymbol{P}_1}{\sim}} \begin{bmatrix} 3 & -2 & 3 & -1 \\ 2 & 1 & -4 & 2 \\ 2 & 1 & -2 & 1 \\ -1 & 3 & -1 & 1 \end{bmatrix} \overset{\boldsymbol{L}_1}{\sim} \begin{bmatrix} 3 & -2 & 3 & -1 \\ \frac{2}{3} & \frac{7}{3} & -6 & \frac{8}{3} \\ \frac{2}{3} & \frac{7}{3} & -4 & \frac{5}{3} \\ -\frac{1}{3} & \frac{7}{3} & 0 & \frac{2}{3} \end{bmatrix}$$

2. korak:

$$\begin{bmatrix} 3 & -2 & 3 & -1 \\ \frac{2}{3} & \frac{7}{3} & -6 & \frac{8}{3} \\ \frac{2}{3} & \frac{7}{3} & -4 & \frac{5}{3} \\ -\frac{1}{3} & \frac{7}{3} & 0 & \frac{2}{3} \end{bmatrix} \underset{2\leftrightarrow 2}{\overset{\boldsymbol{P}_2}{\sim}} \begin{bmatrix} 3 & -2 & 3 & -1 \\ \frac{2}{3} & \frac{7}{3} & -6 & \frac{8}{3} \\ \frac{2}{3} & \frac{7}{3} & -4 & \frac{5}{3} \\ -\frac{1}{3} & \frac{7}{3} & 0 & \frac{2}{3} \end{bmatrix} \overset{\boldsymbol{L}_2}{\sim} \begin{bmatrix} 3 & -2 & 3 & -1 \\ \frac{2}{3} & \frac{7}{3} & -6 & \frac{8}{3} \\ \frac{2}{3} & 1 & 2 & -1 \\ -\frac{1}{3} & 1 & 6 & -2 \end{bmatrix}$$

3. korak:

$$\begin{bmatrix} 3 & -2 & 3 & -1 \\ \frac{2}{3} & \frac{7}{3} & -6 & \frac{8}{3} \\ \frac{2}{3} & 1 & 2 & -1 \\ -\frac{1}{3} & 1 & 6 & -2 \end{bmatrix} \underset{3\leftrightarrow 4}{\overset{\boldsymbol{P}_3}{\sim}} \begin{bmatrix} 3 & -2 & 3 & -1 \\ \frac{2}{3} & \frac{7}{3} & -6 & \frac{8}{3} \\ -\frac{1}{3} & 1 & 6 & -2 \\ \frac{2}{3} & 1 & 2 & -1 \end{bmatrix} \overset{\boldsymbol{L}_3}{\sim} \begin{bmatrix} 3 & -2 & 3 & -1 \\ \frac{2}{3} & \frac{7}{3} & -6 & \frac{8}{3} \\ -\frac{1}{3} & 1 & 6 & -2 \\ \frac{2}{3} & 1 & \frac{1}{3} & -\frac{1}{3} \end{bmatrix}$$

Na ta način matriko $\boldsymbol{A}$ postopno transformiramo v zgornje trikotno matriko $\boldsymbol{U}$. Velja

$$\boldsymbol{L}_3\boldsymbol{P}_3\boldsymbol{L}_2\boldsymbol{P}_2\boldsymbol{L}_1\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{U}.$$

Če uvedemo matriki $\widetilde{\boldsymbol{L}}_2 = \boldsymbol{P}_3\boldsymbol{L}_2\boldsymbol{P}_3^{-1}$ in $\widetilde{\boldsymbol{L}}_1 = \boldsymbol{P}_3\boldsymbol{P}_2\boldsymbol{L}_1\boldsymbol{P}_2^{-1}\boldsymbol{P}_3^{-1}$, lahko zvezo zapišemo kot

$$\boldsymbol{L}_3\widetilde{\boldsymbol{L}}_2\widetilde{\boldsymbol{L}}_1\boldsymbol{P}_3\boldsymbol{P}_2\boldsymbol{P}_1\boldsymbol{A} = \boldsymbol{U}.$$

Permutacijsko matriko $\boldsymbol{P} = \boldsymbol{P}_3\boldsymbol{P}_2\boldsymbol{P}_1$ dobimo iz identitete z uporabo enakih zamenjav vrstic, kot smo jih izvedli pri pivotiranju:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underset{1\leftrightarrow 3}{\overset{\boldsymbol{P}_1}{\sim}} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \underset{3\leftrightarrow 4}{\overset{\boldsymbol{P}_3}{\sim}} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Matriki $\boldsymbol{L} = (\boldsymbol{L}_3\widetilde{\boldsymbol{L}}_2\widetilde{\boldsymbol{L}}_1)^{-1}$ in $\boldsymbol{U}$ razberemo iz izračunane tabele. Ključno je, da smo v postopku izvedli zamenjavo celotnih vrstic tabele in tako izračunali elemente matrik $\widetilde{\boldsymbol{L}}_1^{-1}$, $\widetilde{\boldsymbol{L}}_2^{-1}$, $\boldsymbol{L}_3^{-1}$, ki se nahajajo pod diagonalo. Razcep $\boldsymbol{PA} = \boldsymbol{LU}$ matrike $\boldsymbol{A}$ je torej določen z matrikami

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{2}{3} & 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 1 & 0 \\ \frac{2}{3} & 1 & \frac{1}{3} & 1 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} 3 & -2 & 3 & -1 \\ 0 & \frac{7}{3} & -6 & \frac{8}{3} \\ 0 & 0 & 6 & -2 \\ 0 & 0 & 0 & -\frac{1}{3} \end{bmatrix}, \quad \boldsymbol{P} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Determinanto matrike $\boldsymbol{A}$ izračunamo na podlagi lastnosti

$$\det(\boldsymbol{P})\det(\boldsymbol{A}) = \det(\boldsymbol{PA}) = \det(\boldsymbol{LU}) = \det(\boldsymbol{L})\det(\boldsymbol{U}).$$

Ker je $\det(\boldsymbol{P}) = 1$, $\det(\boldsymbol{L}) = 1$ in $\det(\boldsymbol{U}) = -14$, je $\det(\boldsymbol{A}) = -14$.

Exercise 3.18. In Matlab the LU decomposition with partial pivoting is computed with built-in function `lu`. Test it on the matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ from Exercise 3.14.

Solution. Iz dokumentacije funkcije `lu` (`help lu`) je moč razbrati, da jo lahko uporabimo z različnimi vhodnimi in izhodnimi podatki. Izberemo način, ki sprejme matriko, vrne pa seznam treh matrik.

```
[L,U,P] = lu(A);
```

Rezultat klica je LU razcep $\boldsymbol{PA} = \boldsymbol{LU}$ matrike $\boldsymbol{A}$ z delnim pivotiranjem. Matrika $\boldsymbol{L}$ je spodnje trikotna z enicami na diagonali, matrika $\boldsymbol{U}$ je zgornje trikotna, matrika $\boldsymbol{P}$ pa je permutacijska in povzema postopek pivotiranja pri izračunu razcepa. Pri izvedbi LU razcepa matrike $\boldsymbol{A}_1$ dobimo drugačen razcep od izračunanega po metodi brez pivotiranja. Smiseln razcep dobimo tudi pri matriki $\boldsymbol{A}_2$, na kateri je postopek LU razcepa brez pivotiranja odpovedal.

Exercise 3.19. In Matlab compose a function that finds the solution of the system $\boldsymbol{Ax} = \boldsymbol{b}$ for an invertible matrix $\boldsymbol{A}$ and a vector $\boldsymbol{b}$. In the function first use the built-in command to compute the LU decomposition of $\boldsymbol{A}$ by partial pivoting, and then determine $\boldsymbol{x}$ by forward and backward substitutions. Verify that the results of the implented function match the results of the built-in function `linsolve` or `\` for randomly chosen matrices and vectors generated by the command `rand`.

Solution. Če LU razcep $\boldsymbol{PA} = \boldsymbol{LU}$ matrike $\boldsymbol{A}$ izračunamo z vgrajenim ukazom `lu`, lahko rešitev sistema $\boldsymbol{Ax} = \boldsymbol{b}$ dobimo z reševanjem dveh sistemov: $\boldsymbol{Ly} = \boldsymbol{Pb}$ in $\boldsymbol{Ux} = \boldsymbol{y}$. V teh dveh sistemih nastopata spodnje in zgornje trikotna matrika, zato ju lahko rešimo s premo in obratno substitucijo.

Pri reševanju prvega sistema je treba določiti vse komponente vektorja $\boldsymbol{y}$, ki jih indeksiramo z indeksi i od 1 do n. Na začetku $\boldsymbol{y}$ definiramo kot produkt permutacijske matrike $\boldsymbol{P}$ in vektorja $\boldsymbol{b}$. Nato vsak element $\boldsymbol{y}(i)$ popravimo tako, da od njega odštejemo produkte $\boldsymbol{L}(i,k)$ in $\boldsymbol{y}(k)$ za vse predhodne indekse k od 1 do $i-1$. Ker je matrika $\boldsymbol{L}$ spodnje trikotna in ima na diagonali enice, je končni vektor $\boldsymbol{y}$ res rešitev sistema $\boldsymbol{Ly} = \boldsymbol{Pb}$.

```
y = P*b;
for i = 2:n
    for k = 1:i-1
        y(i) = y(i) - L(i,k)*y(k);
    end
end
```

Pri reševanju sistema $\boldsymbol{Ux} = \boldsymbol{y}$ uberemo obratni pristop. Na začetku za vektor $\boldsymbol{x}$ vzamemo kar $\boldsymbol{y}$, nato pa komponente vektorja $\boldsymbol{x}$, ki jih indeksiramo z i, popravljamo po vrsti od n proti 1. V koraku i od komponente $\boldsymbol{x}(i)$ odštejemo produkte $\boldsymbol{U}(i,k)$ in $\boldsymbol{x}(k)$ za vse indekse k od $i+1$ do n, na koncu pa vrednost $\boldsymbol{x}(i)$ še delimo z diagonalnim elementom $\boldsymbol{U}(i,i)$, ki tu (v splošnem) ni enak 1. Rezultat tega postopka je rešitev $\boldsymbol{x}$ sistema $\boldsymbol{Ux} = \boldsymbol{y}$, ki predstavlja tudi rešitev prvotnega sistema $\boldsymbol{Ax} = \boldsymbol{y}$.

```
x = y;
for i = n:-1:1
    for k = i+1:n
        x(i) = x(i) - U(i,k)*x(k);
    end
    x(i) = x(i)/U(i,i);
end
```

Exercise 3.20. Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $n \in \mathbb{N}$, be an invertible matrix for which the LU decomposition with partial pivoting is known. Let $\boldsymbol{B} \in \mathbb{R}^{n\times n}$ be an invertible matrix obtained from $\boldsymbol{A}$ by substituting one of the columns with a vector $\boldsymbol{b} \in \mathbb{R}^n$. Let $\boldsymbol{c} \in \mathbb{R}^n$. Derive an efficient algorithm for computing the solution $\boldsymbol{x}$ of the system $\boldsymbol{Bx} = \boldsymbol{c}$. Precisely count the number of basic computational operations needed for executing the proposed algorithm.

Solution. Opazimo, da lahko matriko $\boldsymbol{B}$ predstavimo kot $\boldsymbol{B} = \boldsymbol{AE}$, kjer matrika $\boldsymbol{E} \in \mathbb{R}^{n\times n}$ ustreza identični matriki v vseh stolpcih, razen v stolpcu, ki ima enak indeks kot stolpec, v katerem se matriki $\boldsymbol{A}$ in $\boldsymbol{B}$ razlikujeta. Naj bo ta stolpec matrike $\boldsymbol{E}$ označen z $\boldsymbol{e} \in \mathbb{R}^n$. Zanj velja $\boldsymbol{Ae} = \boldsymbol{b}$ in ga lahko ob znanem LU razcepu matrike $\boldsymbol{A}$ izračunamo s premo in obratno substitucijo ($2n^2 + n$ osnovnih računskih operacij).

Naj bo s $\boldsymbol{PA} = \boldsymbol{LU}$ predstavljen LU razcep matrike $\boldsymbol{A}$ z delnim pivotiranjem. Sistem $\boldsymbol{Bx} = \boldsymbol{c}$ lahko sedaj rešimo v treh korakih. Najprej poiščemo rešitev $\boldsymbol{z} \in \mathbb{R}^n$ sistema $\boldsymbol{Lz} = \boldsymbol{Pc}$ s premo substitucijo (n^2 osnovnih računskih operacij). Nato poiščemo rešitev $\boldsymbol{y} \in \mathbb{R}^n$ sistema $\boldsymbol{Uy} = \boldsymbol{z}$ z obratno substitucijo ($n^2 + n$ osnovnih računskih operacij). Nazadnje poiščemo še rešitev $\boldsymbol{x} \in \mathbb{R}^n$ sistema $\boldsymbol{Ex} = \boldsymbol{y}$, ki ustreza rešitvi sistema $\boldsymbol{Bx} = \boldsymbol{c}$.

Naj bo $k \in \mathbb{N}$ indeks stolpca $\boldsymbol{e}$ v matriki $\boldsymbol{E}$. Sistem $\boldsymbol{Ex} = \boldsymbol{y}$ lahko rešimo tako, da najprej z eno osnovno računsko operacijo izračunamo komponento $\boldsymbol{x}$ z indeksom k. Nato z obratno substitucijo z $2(k-1)$ osnovnimi računskimi operacijami izračunamo komponente $\boldsymbol{x}$ z indeksi od 1 do $k-1$ in s premo substitucijo z $2(n-k)$ osnovnimi računskimi operacijami še komponente $\boldsymbol{x}$ z indeksi od $k+1$ do n.

Celoten postopek izračuna $\boldsymbol{x}$ torej zahteva $4n^2 + 4n - 1$ osnovnih računskih operacij.

In computing the LU decomposition we can use the complete instead of the partial pivoting. In each step we search for a pivot element in the reminder of the matrix (and not only under the diagonal of the current column as for the partial pivoting). The pivot element is taken to the left upper part of the reminder of the matrix by exchanging rows and columns, which results in the LU decomposition of the form $\boldsymbol{PAQ} = \boldsymbol{LU}$. Here $\boldsymbol{P}$ and $\boldsymbol{Q}$ are permutation matrices outlining the exchange of rows and columns, and the matrices $\boldsymbol{L}$ and $\boldsymbol{U}$ have the same properties as in the standard LU decomposition. The decomposition with complete pivoting provides a theoretical guarantee for backward stability of the algorithm (as opposed to partial pivoting), but requires far more comparisons than the partial pivoting, which is way the latter is used more often.

Exercise 3.21. Let

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & -3 \\ 4 & 2 & -6 \\ -3 & 3 & 3 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 8 \\ 14 \\ 0 \end{bmatrix}.$$

Compute the LU decomposition of the matrix $\boldsymbol{A}$ by complete pivoting and find the solution of the system $\boldsymbol{Ax} = \boldsymbol{b}$.

Solution. Razcep izračunamo po naslednjih korakih.

1. korak:

$$\begin{bmatrix} 1 & 2 & -3 \\ 4 & 2 & -6 \\ -3 & 3 & 3 \end{bmatrix} \underset{(1,1)\leftrightarrow(2,3)}{\overset{\boldsymbol{P}_1,\,\boldsymbol{Q}_1}{\sim}} \begin{bmatrix} -6 & 2 & 4 \\ -3 & 2 & 1 \\ 3 & 3 & -3 \end{bmatrix} \overset{\boldsymbol{L}_1}{\sim} \begin{bmatrix} -6 & 2 & 4 \\ \frac{1}{2} & 1 & -1 \\ -\frac{1}{2} & 4 & -1 \end{bmatrix}$$

2. korak:

$$\begin{bmatrix} -6 & 2 & 4 \\ \frac{1}{2} & 1 & -1 \\ -\frac{1}{2} & 4 & -1 \end{bmatrix} \underset{(2,2)\leftrightarrow(3,2)}{\overset{\boldsymbol{P}_2,\,\boldsymbol{Q}_2}{\sim}} \begin{bmatrix} -6 & 2 & 4 \\ -\frac{1}{2} & 4 & -1 \\ \frac{1}{2} & 1 & -1 \end{bmatrix} \overset{\boldsymbol{L}_2}{\sim} \begin{bmatrix} -6 & 2 & 4 \\ -\frac{1}{2} & 4 & -1 \\ \frac{1}{2} & \frac{1}{4} & -\frac{3}{4} \end{bmatrix}$$

S tem postopkom smo matriko $\boldsymbol{A}$ transformirali v obrnljivo zgornje trikotno matriko $\boldsymbol{U} = \boldsymbol{L}_2\boldsymbol{P}_2\boldsymbol{L}_1\boldsymbol{P}_1\boldsymbol{A}\boldsymbol{Q}_1\boldsymbol{Q}_2$, na podlagi česar dobimo razcep $\boldsymbol{PAQ} = \boldsymbol{LU}$. Trikotni matriki, ki ju razberemo iz izračunane tabele, sta podani z

$$\boldsymbol{L} = \boldsymbol{P}_2\boldsymbol{L}_1^{-1}\boldsymbol{P}_2^{-1}\boldsymbol{L}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & \frac{1}{4} & 1 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} -6 & 2 & 4 \\ 0 & 4 & -1 \\ 0 & 0 & -\frac{3}{4} \end{bmatrix},$$

permutacijsko matriki, ki ju lahko dobimo z ustreznimi permutacijami vrstic oziroma stolpcev identite, pa z

$$\boldsymbol{P} = \boldsymbol{P}_2\boldsymbol{P}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{Q} = \boldsymbol{Q}_1\boldsymbol{Q}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Rešitev sistema $\boldsymbol{Ax} = \boldsymbol{b}$ določimo z reševanjem sistema $\boldsymbol{Ly} = \boldsymbol{Pb}$:

$$\begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & \frac{1}{4} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 0 \\ 8 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 7 \\ -\frac{3}{4} \end{bmatrix};$$

in sistema $\boldsymbol{Uz} = \boldsymbol{y}$:

$$\begin{bmatrix} -6 & 2 & 4 \\ 0 & 4 & -1 \\ 0 & 0 & -\frac{3}{4} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 14 \\ 7 \\ -\frac{3}{4} \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix};$$

ter permutacijo $\boldsymbol{x} = \boldsymbol{Qz} = (1, 2, -1)$.

In finding the solution $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$ of the system $\boldsymbol{Ax} = \boldsymbol{b}$ we do not compute the inverse $\boldsymbol{A}^{-1}$ of the matrix $\boldsymbol{A}$ since this is computationally more demanding and less stable than computing $\boldsymbol{x}$ by forward and backward substitutions after the LU decomposition. However, there exist problems where the explicit inverse of a matrix is needed. To achieve this the LU decomposition can be used again. An important consideration is also how to compute the inverse of an expanded or modified matrix based on the previously computed inverse of the initial matrix.

Exercise 3.22. Describe how the inverse of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ can be computed by applying the LU decomposition. How many operations in dependence of n are needed? Demonstrate the procedure on the matrix

$$\boldsymbol{A} = \begin{bmatrix} 2 & 3 & 1 \\ 4 & 8 & 3 \\ 0 & -2 & 3 \end{bmatrix}.$$

Solution. Naj bo $\boldsymbol{X} = \boldsymbol{A}^{-1}$ inverz matrike $\boldsymbol{A}$. Ker je $\boldsymbol{AX} = \boldsymbol{I}$, lahko $\boldsymbol{X}$ določimo z reševanjem sistemov $\boldsymbol{Ax}_j = \boldsymbol{e}_j$, $j = 1, 2, \ldots, n$, kjer $\boldsymbol{x}_j$ označuje j-ti stolpec matrike $\boldsymbol{X}$, $\boldsymbol{e}_j$ pa j-ti standardni enotski vektor velikosti n. Najprej izračunamo LU razcep matrike $\boldsymbol{A}$, za kar je potrebnih $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ operacij. Nato s premimi in obratnimi substitucijami rešimo vsakega izmed n sistemov, kar zahtevata $n(2n^2+n))$ operacij. Celoten postopek je torej zahtevnosti $\frac{8}{3}n^3 + \mathcal{O}(n^2)$.

Poiščimo inverz dane matrike $\boldsymbol{A}$. Najprej izračunamo LU razcep $\boldsymbol{PA} = \boldsymbol{LU}$ matrike $\boldsymbol{A}$ z delnim pivotiranjem. Določen je z matrikami

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} 4 & 8 & 3 \\ 0 & -2 & 3 \\ 0 & 0 & -2 \end{bmatrix}, \quad \boldsymbol{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Z reševanjem sistemov $\boldsymbol{Ly}_j = \boldsymbol{Pe}_j$ in $\boldsymbol{Ux}_j = \boldsymbol{y}_j$, $j = 1, 2, \ldots, n$, dobimo inverz

$$\boldsymbol{X} = \boldsymbol{A}^{-1} = \begin{bmatrix} \frac{15}{8} & -\frac{11}{16} & \frac{1}{16} \\ -\frac{3}{4} & \frac{3}{8} & -\frac{1}{8} \\ -\frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}.$$

Exercise 3.23. Given is an invertible matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and its inverse $\boldsymbol{A}^{-1}$. Compose an efficient algorithm to compute the inverse of the matrix

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{u} \\ \boldsymbol{v}^\mathsf{T} & \alpha \end{bmatrix}, \qquad \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n,\ \alpha \in \mathbb{R},$$

provided that $\boldsymbol{B}$ is invertible. How many operations are needed to perform the algorithm and what is the necessary and sufficient condition for invertibility of $\boldsymbol{B}$?

Solution. Zapišimo inverz $\boldsymbol{B}^{-1}$ matrike $\boldsymbol{B}$ v obliki

$$\boldsymbol{B}^{-1} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{x} \\ \boldsymbol{y}^\mathsf{T} & \beta \end{bmatrix}, \qquad \boldsymbol{C} \in \mathbb{R}^{n\times n},\ \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n,\ \beta \in \mathbb{R}.$$

Iz

$$\boldsymbol{BB}^{-1} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{u} \\ \boldsymbol{v}^\mathsf{T} & \alpha \end{bmatrix} \begin{bmatrix} \boldsymbol{C} & \boldsymbol{x} \\ \boldsymbol{y}^\mathsf{T} & \beta \end{bmatrix} = \boldsymbol{I}_{n+1}$$

sledi

$$\boldsymbol{AC} + \boldsymbol{uy}^\mathsf{T} = \boldsymbol{I}_n, \quad \boldsymbol{Ax} + \beta\boldsymbol{u} = 0, \quad \boldsymbol{v}^\mathsf{T}\boldsymbol{C} + \alpha\boldsymbol{y}^\mathsf{T} = 0, \quad \boldsymbol{v}^\mathsf{T}\boldsymbol{x} + \alpha\beta = 1.$$

S pomočjo $\boldsymbol{A}^{-1}$ lahko od tod izrazimo

$$\boldsymbol{C} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{uy}^\mathsf{T}, \quad \boldsymbol{x} = -\beta\boldsymbol{A}^{-1}\boldsymbol{u}, \quad \beta = \frac{1}{\alpha - \boldsymbol{v}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{u}}, \quad \boldsymbol{y}^\mathsf{T} = -\beta\boldsymbol{v}^\mathsf{T}\boldsymbol{A}^{-1}.$$

Torej lahko izračun inverza matrike $\boldsymbol{B}$ izvedemo v korakih:

- $\boldsymbol{z} = \boldsymbol{A}^{-1}\boldsymbol{u}$ ($2n^2 - n$ operacij),

- $\beta = 1/(\alpha - \boldsymbol{v}^\mathsf{T}\boldsymbol{z})$ ($2n+1$ operacij),
- $\boldsymbol{y}^\mathsf{T} = -\beta\boldsymbol{v}^\mathsf{T}\boldsymbol{A}^{-1}$ ($2n^2$ operacij),
- $\boldsymbol{x} = -\beta\boldsymbol{z}$ (n operacij),
- $\boldsymbol{C} = \boldsymbol{A}^{-1} - \boldsymbol{z}\boldsymbol{y}^\mathsf{T}$ ($2n^2$ operacij).

Celoten postopek izračuna torej zahteva $6n^2 + 2n + 1$ operacij. Iz postopka je razvidno, da inverz matrike $\boldsymbol{B}$ obstaja, če je $\alpha \neq \boldsymbol{v}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{u}$. Velja tudi obratno. Če je $\alpha = \boldsymbol{v}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{u}$, potem je

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{u} \\ \boldsymbol{v}^\mathsf{T} & \alpha \end{bmatrix} \begin{bmatrix} \boldsymbol{A}^{-1}\boldsymbol{u} \\ -1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ 0 \end{bmatrix},$$

kar pomeni, da v jedru matrike $\boldsymbol{B}$ obstaja neničeln vektor, zato ni obrnljiva.

Exercise 3.24. Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be an invertible matrix. Prove that for vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ the matrix $\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\mathsf{T}$ is invertible if and only if $1 + \boldsymbol{v}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{u} \neq 0$, and that in this case the Sherman–Morrison formula holds:

$$\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\mathsf{T}\right)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^\mathsf{T}\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{u}}.$$

Solution. Iz naloge 3.23 sledi, da je matrika

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{u} \\ \boldsymbol{v}^\mathsf{T} & -1 \end{bmatrix}$$

obrnljiva natanko tedaj, ko je $1 + \boldsymbol{v}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{u} \neq 0$. Opazimo, da je

$$\begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{u} \\ \boldsymbol{0}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{u} \\ \boldsymbol{v}^\mathsf{T} & -1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0} \\ \boldsymbol{v}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\mathsf{T} & \boldsymbol{0} \\ \boldsymbol{0}^\mathsf{T} & -1 \end{bmatrix},$$

pri čemer sta matriki, s katerimi smo množili matriko $\boldsymbol{B}$, obrnljivi:

$$\begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{u} \\ \boldsymbol{0}^\mathsf{T} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{u} \\ \boldsymbol{0}^\mathsf{T} & 1 \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0} \\ \boldsymbol{v}^\mathsf{T} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0} \\ -\boldsymbol{v}^\mathsf{T} & 1 \end{bmatrix}.$$

Sledi, da je tudi matrika $\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\mathsf{T}$ obrnljiva natanko tedaj, ko je $1 + \boldsymbol{v}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{u} \neq 0$. Za inverz glede na oznake iz naloge 3.23 velja

$$\begin{bmatrix} \left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\mathsf{T}\right)^{-1} & \boldsymbol{0} \\ \boldsymbol{0}^\mathsf{T} & -1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0} \\ -\boldsymbol{v}^\mathsf{T} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{C} & \boldsymbol{x} \\ \boldsymbol{y}^\mathsf{T} & \beta \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{u} \\ \boldsymbol{0}^\mathsf{T} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \\ \boldsymbol{0}^\mathsf{T} & -1 \end{bmatrix},$$

kar dokazuje Sherman–Morrisonovo formulo.

Exercise 3.25. Suppose that the solving of the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with the LU decomposition $\boldsymbol{P}\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ of an invertible matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is given. How would you efficiently compute the solution of the system $\boldsymbol{B}\boldsymbol{y} = \boldsymbol{b}$ with an invertible matrix $\boldsymbol{B}$ which differes from $\boldsymbol{A}$ in one single element?

Solution. Recimo, da se matriki $\boldsymbol{A}$ in $\boldsymbol{B}$ razlikujeta v elementih na mestu (i,j), ki ju označimo z $a_{i,j}$ in $b_{i,j}$. Potem je

$$\boldsymbol{B} = \boldsymbol{A} + (b_{i,j} - a_{i,j})\boldsymbol{e}_i\boldsymbol{e}_j^{\mathsf{T}},$$

pri čemer sta $\boldsymbol{e}_i, \boldsymbol{e}_j \in \mathbb{R}^{n\times n}$ standardna enotska vektorja. Rešitev sistema $\boldsymbol{By} = \boldsymbol{b}$ lahko izrazimo s pomočjo Sherman–Morrisonove formule iz naloge 3.24 kot

$$\boldsymbol{y} = \boldsymbol{B}^{-1}\boldsymbol{b} = \boldsymbol{A}^{-1}\boldsymbol{b} - \frac{\boldsymbol{e}_j^{\mathsf{T}}\boldsymbol{A}^{-1}\boldsymbol{b}}{1+\boldsymbol{e}_j^{\mathsf{T}}\boldsymbol{A}^{-1}\boldsymbol{e}_i}\boldsymbol{A}^{-1}\boldsymbol{e}_i = \boldsymbol{x} - \frac{\boldsymbol{e}_j^{\mathsf{T}}\boldsymbol{x}}{1+\boldsymbol{e}_j^{\mathsf{T}}\boldsymbol{A}^{-1}\boldsymbol{e}_i}\boldsymbol{A}^{-1}\boldsymbol{e}_i.$$

Za izračun $\boldsymbol{y}$ torej zadošča poiskati rešitev sistema $\boldsymbol{Az} = \boldsymbol{e}_i$, ki jo lahko dobimo z $2n^2+n$ operacijami preme in obratne substitucije ob pomoči LU razcepa matrike $\boldsymbol{A}$. Rešitev $\boldsymbol{y}$ potem ustreza vektorju $\boldsymbol{x} - (\boldsymbol{x}_j/(1+\boldsymbol{z}_j))\boldsymbol{z}$, ki ga izračunamo z dodatnimi $2n+2$ operacijami.

3.3. Solving Systems of Special Form

The systems of linear equations encountered in practice often have specific properties that reflect on the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ of the system $\boldsymbol{Ax} = \boldsymbol{b}$. Such properties can sometimes be exploited to reduce the computational complexity of solving the system or they can be used in stability analysis.

If the matrix $\boldsymbol{A}$ is symmetric positive definite ($\boldsymbol{A} = \boldsymbol{A}^{\mathsf{T}}$ and $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{Ax} > 0$ for every $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$), we solve the system not by the LU decomposition but by the Cholesky decomposition, which requires only $\frac{1}{3}n^3 + \mathcal{O}(n^2)$ operations. In this decomposition the matrix $\boldsymbol{A}$ is represented as the product of the lower triangular matrix $\boldsymbol{V}$ and its transpose: $\boldsymbol{A} = \boldsymbol{VV}^{\mathsf{T}}$. The matrix $\boldsymbol{V}$ is called the Cholesky factor and has positive elements on the diagonal. The Cholesky decomposition can be computed in Matlab as follows.

```
n = size(A,1);
V = zeros(n);
for j = 1:n
    V(j,j) = sqrt(A(j,j) - V(j,1:j-1)*V(j,1:j-1)');
    for i = j+1:n
        V(i,j) = (A(i,j)-V(i,1:j-1)*V(j,1:j-1)')/V(j,j);
    end
end
```

Exercise 3.26. Given is the matrix

$$\boldsymbol{A} = \begin{bmatrix} 4 & 6 & 2 & -4 \\ 6 & 18 & 0 & 3 \\ 2 & 0 & 3 & -4 \\ -4 & 3 & -4 & \alpha \end{bmatrix}, \qquad \alpha \in \mathbb{R}.$$

1. Determine all numbers α such that the matrix $\boldsymbol{A}$ is symmetric positive definite.

2. Let $\alpha = 23$. Use the Cholesky decomposition to solve the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ where $\boldsymbol{b} = (6, 15, 2, 1)$.

Solution. Za dano matriko $\boldsymbol{A}$ je faktor Choleskega podan z

$$\boldsymbol{V} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 3 & 3 & 0 & 0 \\ 1 & -1 & 1 & 0 \\ -2 & 3 & 1 & \sqrt{\alpha - 14} \end{bmatrix}$$

in obstaja natanko tedaj, ko je $\alpha > 14$.

1. Matrika $\boldsymbol{A}$ je simetrična pozitivno definitna natanko tedaj, ko obstaja razcep Choleskega, torej natanko tedaj, ko je $\alpha > 14$.

2. Rešitev sistema poiščemo z reševanjem dveh preprostejših sistemov: $\boldsymbol{V}\boldsymbol{y} = \boldsymbol{b}$ in $\boldsymbol{V}^\mathsf{T}\boldsymbol{x} = \boldsymbol{y}$. Pri prvem uporabimo premo substitucijo, pri drugem pa obratno substitucijo. Izračunamo $\boldsymbol{y} = (3, 2, 1, 0)$ in $\boldsymbol{x} = (-1/2, 1, 1, 0)$.

Exercise 3.27. Let

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{a} \\ \boldsymbol{a}^\mathsf{T} & \alpha \end{bmatrix}, \quad \boldsymbol{A} \in \mathbb{R}^{n\times n},\ \boldsymbol{a} \in \mathbb{R}^n,\ \alpha \in \mathbb{R}.$$

What are the minimal sufficient conditions that the matrix $\boldsymbol{B}$ is symmetric positive definite? Assume them and determine the Cholesky decomposition of $\boldsymbol{B}$.

Solution. Matrika $\boldsymbol{B}$ je simetrično pozitivno definitna natanko tedaj, ko obstaja razcep Choleskega

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{a} \\ \boldsymbol{a}^\mathsf{T} & \alpha \end{bmatrix} = \begin{bmatrix} \boldsymbol{V} & \boldsymbol{0} \\ \boldsymbol{b}^\mathsf{T} & \beta \end{bmatrix} \begin{bmatrix} \boldsymbol{V}^\mathsf{T} & \boldsymbol{b} \\ \boldsymbol{0}^\mathsf{T} & \beta \end{bmatrix}, \quad \boldsymbol{V} \in \mathbb{R}^{n\times n},\ \boldsymbol{b} \in \mathbb{R}^n,\ \beta \in \mathbb{R},$$

kjer je $\boldsymbol{V}$ spodnje trikotna matrika s pozitivnimi diagonalnimi elementi in $\beta > 0$. Iz tega sledi, da je $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{V}^\mathsf{T}$ in matrika $\boldsymbol{V}$ (če obstaja) je faktor Choleskega matrike $\boldsymbol{A}$. Torej je simetričnost in pozitivna definitnost matrike $\boldsymbol{A}$ potreben pogoj za to, da ima enako lastnost tudi matrika $\boldsymbol{B}$. Poleg tega nastavek implicira, da je $\boldsymbol{V}\boldsymbol{b} = \boldsymbol{a}$ in $\boldsymbol{b}^\mathsf{T}\boldsymbol{b} + \beta^2 = \alpha$ oziroma $\boldsymbol{b} = \boldsymbol{V}^{-1}\boldsymbol{a}$ (matrika $\boldsymbol{V}$ je obrnljiva) in $\beta = \sqrt{\alpha - \boldsymbol{a}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{a}}$. Slednje pomeni, da mora za pozitivno definitnost matrike $\boldsymbol{B}$ veljati še $\alpha > \boldsymbol{a}^\mathsf{T}\boldsymbol{A}^{-1}\boldsymbol{a}$.

Exercise 3.28. Given are the matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$. The matrix $\boldsymbol{B}$ is symmetric positive definite. Compose an efficient algorithm for the computation of the trace of the matrix $\boldsymbol{A}^\mathsf{T}\boldsymbol{B}^{-1}\boldsymbol{A}$ and count how many operations it requires.

Solution. Naj bo i-ti stolpec matrike $\boldsymbol{A}$ označen z $\boldsymbol{a}_i$, $i = 1, 2, \ldots, n$. Potem je

$$(\boldsymbol{A}^\mathsf{T}\boldsymbol{B}^{-1}\boldsymbol{A})_{i,j} = \boldsymbol{a}_i^\mathsf{T}\boldsymbol{B}^{-1}\boldsymbol{a}_j$$

in

$$\operatorname{sled}(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{A}) = \boldsymbol{a}_1^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{a}_1 + \boldsymbol{a}_2^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{a}_2 + \ldots + \boldsymbol{a}_n^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{a}_n.$$

Za učinkovit izračun sledi torej zadošča premisliti, kako učinkovito izračunati vrednosti $\boldsymbol{a}_i^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{a}_i$, $i = 1, 2, \ldots, n$. Ker je $\boldsymbol{B}$ simetrična pozitivno definitna matrika, jo lahko zapišemo v obliki $\boldsymbol{B} = \boldsymbol{V}\boldsymbol{V}^{\mathsf{T}}$, kjer je $\boldsymbol{V}$ faktor Choleskega. Iz razcepa sledi

$$\boldsymbol{a}_i^{\mathsf{T}}\boldsymbol{B}^{-1}\boldsymbol{a}_i = \boldsymbol{a}_i^{\mathsf{T}}(\boldsymbol{V}\boldsymbol{V}^{\mathsf{T}})^{-1}\boldsymbol{a}_i = (\boldsymbol{V}^{-1}\boldsymbol{a}_i)^{\mathsf{T}}(\boldsymbol{V}^{-1}\boldsymbol{a}_i) = \left\|\boldsymbol{V}^{-1}\boldsymbol{a}_i\right\|_2^2.$$

Izračun matrike $\boldsymbol{V}$ zahteva $\frac{1}{3}n^3 + \mathcal{O}(n^2)$ operacij. Vrednost $\boldsymbol{V}^{-1}\boldsymbol{a}_i$ izračunamo s premimi substitucijami z $n^2 + n$ operacijami. Za izračun kvadrata druge norme je potrebnih še dodatnih $2n - 1$ operacij. Celoten postopek izračuna sledi torej terja

$$\left(\frac{1}{3}n^3 + \mathcal{O}(n^2)\right) + n\left(\left(n^2 + n\right) + (2n - 1)\right) + (n - 1) = \frac{4}{3}n^3 + \mathcal{O}(n^2)$$

osnovnih računskih operacij.

A matrix is strictly diagonal dominant if in each of its rows the absolute value of the diagonal element is larger than the sum of the absolute values of the outer-diagonal elements. In computing the LU decomposition of such matrices the pivoting is not necessary. If additionally the matrix is tridiagonal, the LU decomposition can be significantly simplied and shortened.

Exercise 3.29. Prove that every strictly diagonal dominant matrix admits the LU decomposition (without pivoting).

Solution. Preverimo, da je po vrsticah strogo diagonalno dominantna matrika $\boldsymbol{A}$ velikosti $n \times n$ z elementi $a_{i,j}$ nesingularna. Denimo nasprotno, da je $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{0}$ za nek neničeln vektor $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$. Naj bo i indeks po absolutni vrednosti največjega elementa $\boldsymbol{u}$. Ker je

$$\sum_{j=1}^{n} a_{i,j}u_j = 0 \quad \Leftrightarrow \quad a_{i,i}u_i = -\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{i,j}u_j \quad \Leftrightarrow \quad a_{i,i} = -\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{u_j}{u_i}a_{i,j},$$

iz izbire indeksa i sledi

$$|a_{i,i}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{i,j}|,$$

kar je v protislovju s strogo diagonalno dominantnostjo matrike $\boldsymbol{A}$. Ker je vsaka vodilna podmatrika strogo diagonalno dominantne matrike tudi strogo diagonalno dominantna, so vse vodilne podmatrike $\boldsymbol{A}$ nesingularne, kar zagotavlja obstoj LU razcepa (brez pivotiranja).

Exercise 3.30. Given is a tridiagonal matrix

$$\boldsymbol{A} = \begin{bmatrix} a_1 & b_1 & & & \\ c_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & a_{n-1} & b_{n-1} \\ & & & c_{n-1} & a_n \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

1. Provided that the LU decomposition of $\boldsymbol{A}$ (without pivoting) exists, simplify the decomposition algorithm.

2. After computing the decomposition $\boldsymbol{A} = \boldsymbol{LU}$, the solution $\boldsymbol{x} \in \mathbb{R}^n$ to the system $\boldsymbol{Ax} = \boldsymbol{z}$, $\boldsymbol{z} \in \mathbb{R}^n$, is determined by forward and backward substitutions. Compare the number of operations in the general algorithm to the number of operations needed to compute the solution to the system in case of a tridiagonal matrix $\boldsymbol{A}$.

3. Implement the algorithm for solving the system with a tridiagonal system in Matlab. This is the so called Thomas algorithm. Generate random data for $n = 10^4$ such that the matrix $\boldsymbol{A}$ is strictly diagonal dominant. Then measure the time needed for computing the solution by Thomas algorithm and compare it to the time needed for computing the solution by the built-in function.

Solution.

1. Pri LU razcepu tridiagonalne matrike sta matriki $\boldsymbol{L}$ in $\boldsymbol{U}$ oblike

$$\boldsymbol{L} = \begin{bmatrix} 1 & & & & \\ l_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & l_{n-2} & 1 & \\ & & & l_{n-1} & 1 \end{bmatrix}, \qquad \boldsymbol{U} = \begin{bmatrix} u_1 & b_1 & & & \\ & u_2 & b_2 & & \\ & & \ddots & \ddots & \\ & & & u_{n-1} & b_{n-1} \\ & & & & u_n \end{bmatrix}.$$

   Velja $u_1 = a_1$, vrednosti l_k in u_{k+1} za $k = 1, 2, \ldots, n-1$ pa se izražajo kot

$$l_k = \frac{c_k}{u_k}, \qquad u_{k+1} = a_{k+1} - l_k b_k.$$

   Za izračun LU razcepa je torej potrebnih vsega $3n - 3$ operacij.

2. Sistem rešujemo s premo in obratno substitucijo. Namesto n^2 operacij je pri premi substituciji potrebnih le $2n-2$ operacij. Podobno je pri obratni substituciji namesto n^2+n operacij potrebnih le $3n-2$ operacij. Celoten postopek reševanja skupaj z LU razcepom torej zahteva $8n-7$ operacij, kar je za dva reda manj kot pri LU razcepu splošne matrike.

3. Funkcija, ki izračuna rešitev sistema $\boldsymbol{Ax} = \boldsymbol{z}$ sprejme obliko $\boldsymbol{A}$ v obliki treh seznamov: seznam $\boldsymbol{a}$ vsebuje elemente na diagonali matrike $\boldsymbol{A}$ in je dolžine n, seznama $\boldsymbol{b}$ in $\boldsymbol{c}$ pa vsebujeta elemente na naddiagonali oziroma poddiagonali

matrike $\boldsymbol{A}$ ter sta dolžine $n-1$. Na ta način matriko $\boldsymbol{A}$ namesto z n^2 elementi predstavimo le s seznami v skupni dolžini $3n-2$. Izhodni podatek funkcije je rešitev sistema $\boldsymbol{x}$, pomožna izhodna podatka pa sta še seznam $\boldsymbol{\ell}$ dolžine $n-1$, ki določa matriko $\boldsymbol{L}$, in seznam $\boldsymbol{u}$ dolžine n, ki določa matriko $\boldsymbol{U}$.

```
function [x,l,u] = thomas(a,b,c,z)

n = length(a);
u = zeros(n,1);
l = zeros(n-1,1);

% LU razcep
u(1) = a(1);
for k = 1:n-1
    l(k) = c(k)/u(k);
    u(k+1) = a(k+1) - l(k)*b(k);
end

% prema substitucija
y = z;
for k = 2:n
    y(k) = z(k) - l(k-1)*y(k-1);
end

% obratna substitucija
x = y;
x(n) = y(n)/u(n);
for k = n-1:-1:1
    x(k) = (y(k) - b(k)*x(k+1))/u(k);
end

end
```

Primerjava reševanja sistema z matriko velikosti $n = 10^4$ z uporabo Thomasovega postopka in vgrajene funkcije (`linsolve` oziroma operator \) pokaže, da je naša funkcija približno 500-krat hitrejša.

```
n = 1e4;
a = rand(n,1) + 2;
b = rand(n-1,1);
c = rand(n-1,1);
z = rand(n,1);

tic; x1 = thomas(a,b,c,z); t1 = toc;
```

```
A = diag(a) + diag(b,1) + diag(c,-1);
tic; x2 = A\z; t2 = toc;

t2/t1; % priblizno 500
```

The pivot growth g in the LU decomposition of the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is

$$g = \frac{\max_{i,j=1,\ldots,n} |u_{i,j}|}{\max_{i,j=1,\ldots,n} |a_{i,j}|},$$

where $a_{i,j}$ denote the elements of $\boldsymbol{A}$ and $u_{i,j}$ the elements of the invertible upper triangular matrix $\boldsymbol{U}$ in the decomposition of $\boldsymbol{A}$. The pivot growth can be used in backward stability analaysis of solving the system $\boldsymbol{Ax} = \boldsymbol{b}$ by the LU decomposition with pivoting. If the value of g is small, then solving of the system is backward stable since the computed $\widehat{\boldsymbol{x}}$ obtained in place of $\boldsymbol{x}$ is the exact solution of the system $(\boldsymbol{A}+\boldsymbol{\Delta A})\widehat{\boldsymbol{x}} = \boldsymbol{b}$, where $\boldsymbol{\Delta A}$ is the matrix satisfying

$$\|\boldsymbol{\Delta A}\|_\infty \leq 3nu \|\boldsymbol{L}\|_\infty \|\boldsymbol{U}\|_\infty \leq 3gn^3 u \|\boldsymbol{A}\|_\infty .$$

Here, u denotes the unit roundoff of the chosen arithmetic. The pivot growth g in the LU decomposition with partial pivoting is in general bounded upwards by 2^{n-1}, which means that the procedure is in threory not backward stable. For matrices with special structure this estimate can be improved.

Exercise 3.31. The matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ with the elements $a_{i,j}$ is upper Hessenberg if $a_{i,j} = 0$ for $i > j+1$. Prove that the pivot growth of an upper Hessenberg matrix in the LU decomposition with partial pivoting is bounded by n.

Solution. Naj bo $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ zgornja Hessenbergova matrika. Zaradi posebne oblike te matrike imamo na vsakem koraku LU razcepa z delnim pivotiranjem dve možnosti za pivot (ohranimo obstoječe stanje ali zamenjamo trenutno in naslednjo vrstico). Naj bo $M = \max_{i,j=1,\ldots,n} |a_{i,j}|$. V prvem koraku postopka je situacija naslednja.

- Če je $|a_{1,1}| \geq |a_{2,1}|$, pivotiranje ni potrebno in novi elementi $a^{(1)}_{2,k}$ na mestih $(2,k)$, $k = 2,3,\ldots,n$, so izračunani kot

  $$a^{(1)}_{2,k} = a_{2,k} - \ell_{2,1}a_{1,k}, \qquad \ell_{2,1} = \frac{a_{2,1}}{a_{1,1}}.$$

  Ker je $|\ell_{2,1}| \leq 1$, je $|a^{(1)}_{2,k}| \leq 2M$.

- Če je $|a_{1,1}| < |a_{2,1}|$, zamenjamo prvo in drugo vrstico in tedaj velja

  $$a^{(1)}_{2,k} = a_{1,k} - \ell_{2,1}a_{2,k}, \qquad \ell_{2,1} = \frac{a_{1,1}}{a_{2,1}}.$$

  Ker je $|\ell_{2,1}| < 1$, je tudi v tem primeru $|a^{(1)}_{2,k}| \leq 2M$.

Zaradi lastnosti matrike $\boldsymbol{A}$ preostale vrstice (od tretje dalje) v prvem koraku LU razcepa ostanejo nespremenjene. Recimo, da smo že opravili $r-1$ korakov postopka in predpostavimo, da je $|a_{r,k}^{(r-1)}| \leq rM$, $k = r, \ldots, n$. Ocenimo nove elemente $a_{r+1,k}^{(k)}$ na mestih $(r+1, k)$, $k = r+1, r+2 \ldots, n$.

- Če je $|a_{r,r}^{(1)}| \geq |a_{r+1,r+1}|$, pivotiranje ni potrebno in za

$$a_{r+1,k}^{(r)} = a_{r+1,k} - \ell_{r+1,r} a_{r,k}^{(r-1)}, \qquad \ell_{r+1,r} = \frac{a_{r+1,r}}{a_{r,r}^{(r-1)}},$$

  po predpostavki velja $|a_{r+1,k}^{(r)}| \leq M + rM = (r+1)M$.

- Če je $|a_{r,r}^{(1)}| < |a_{r+1,r+1}|$, pa zamenjamo vrstici z indeksoma r in $r+1$ ter izračunamo

$$a_{r+1,j}^{(r)} = a_{r,k}^{(r-1)} - \ell_{r+1,r} a_{r+1,k}, \qquad \ell_{r+1,r} = \frac{a_{r,r}^{(r-1)}}{a_{r+1,r}}.$$

  Tudi v tem primeru je po predpostavki $|a_{r+1,k}^{(r)}| \leq rM + M = (r+1)M$.

LU razcep izračunamo v $n-1$ korakih. Za element $a_{n,n}^{(n-1)}$ velja $|a_{n,n}^{(n-1)}| \leq nM$. Pokazali smo torej, da so vsi elementi v matriki $\boldsymbol{U}$ manjši od nM, zato je $g \leq n$.

Exercise 3.32. Verify that for the upper Hessenberg matrix

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 1 \\ -1 & 1 & 0 & \ldots & 0 & 1 \\ 0 & -1 & 1 & \ddots & \vdots & 1 \\ \vdots & \ddots & \ddots & \ddots & 0 & \vdots \\ 0 & \ldots & 0 & -1 & 1 & 1 \\ 0 & 0 & \ldots & 0 & -1 & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

the pivot growth g is equal to n.

Solution. Pri LU razcepu matrike $\boldsymbol{A}$ v nobenem koraku ni potrebe po pivotiranju. Po standardnem postopku izračunamo LU razcep $\boldsymbol{A} = \boldsymbol{LU}$, ki ga določata matriki

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & \ldots & 0 & 0 \\ -1 & 1 & \ddots & \vdots & \vdots \\ 0 & -1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & 1 & 0 \\ 0 & \ldots & 0 & -1 & 1 \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} 1 & 0 & \ldots & 0 & 1 \\ 0 & 1 & \ddots & \vdots & 2 \\ 0 & 0 & \ddots & 0 & 3 \\ \vdots & \ddots & \ddots & 1 & \vdots \\ 0 & \ldots & 0 & 0 & n \end{bmatrix}.$$

Po definiciji pivotne rasti je $g = n$.

4. Systems of Non-linear Equations

The system of n non-linear equations with n unknowns is given in the form $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$, where $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is a mapping, $\boldsymbol{0} \in \mathbb{R}^n$ is the vector of zeros and $\boldsymbol{x} \in \mathbb{R}^n$ is the vector representing the solution to the system. The solving of the system can be approached similarly as the solving of non-linear equations.

4.1. The Jacobi Method

The idea behind the basic method for solving the system $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ is the same as for the fixed-point iteration. The system is transformed into the form $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$, where $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^n$ denotes the iteration mapping. The solution is found by choosing an initial vector $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ and performing the iteration

$$\boldsymbol{x}^{(r+1)} = \boldsymbol{g}\left(\boldsymbol{x}^{(r)}\right), \quad r = 0, 1, \ldots.$$

This procedure is called the Jacobi method. It can usually be accelerated by using the already computed coordinates. The approximation in the step $r+1$ is

$$\boldsymbol{x}_i^{(r+1)} = \boldsymbol{g}_i\left(\boldsymbol{x}_1^{(r+1)}, \ldots, \boldsymbol{x}_{i-1}^{(r+1)}, \boldsymbol{x}_i^{(r)}, \ldots, \boldsymbol{x}_n^{(r)}\right), \quad i = 1, 2, \ldots, n.$$

where $\boldsymbol{g}_i$ denotes the component of the mapping $\boldsymbol{g}$ with index i. Such iteration is called the Seidel metod.

Exercise 4.1. Given is the system

$$x = \sin\left(\frac{2x - y}{4}\right), \quad y = \cos\left(\frac{x + 2y}{4}\right)$$

of two non-linear equations with the unknowns x and y. For the initial approximation $(x_0, y_0) = (2\pi, 0)$ compute the first and second approximation of the Jacobi and Seidel method. Using Matlab perform another 13 steps. Compare the approximations to $(-0.43752408, 0.93632532)$, which is an exact approximation of the solution.

Solution. Pri Jacobijevi iteraciji za prva približka (x_1, y_1) in (x_2, y_2) dobimo

$$x_1 = \sin\left(\tfrac{2x_0-y_0}{4}\right) = \sin(\pi) = 0, \qquad y_1 = \cos\left(\tfrac{x_0+2y_0}{4}\right) = \cos\left(\tfrac{\pi}{2}\right) = 0,$$
$$x_2 = \sin\left(\tfrac{2x_1-y_1}{4}\right) = \sin(0) = 0, \qquad y_2 = \cos\left(\tfrac{x_1+2y_1}{4}\right) = \cos(0) = 1,$$

pri Seidlovi iteraciji pa

$$x_1 = \sin\left(\tfrac{2x_0-y_0}{4}\right) = \sin(\pi) = 0, \qquad y_1 = \cos\left(\tfrac{x_1+2y_0}{4}\right) = \cos(0) = 1,$$
$$x_2 = \sin\left(\tfrac{2x_1-y_1}{4}\right) = -\sin\left(\tfrac{1}{4}\right), \qquad y_2 = \cos\left(\tfrac{x_2+2y_1}{4}\right) = \cos\left(\frac{-\sin(\frac{1}{4})+2}{4}\right).$$

Naslednjih 13 približkov izračunamo v Matlabu.

```
g1 = @(x)sin((2*x(1)-x(2))/4);
g2 = @(x)cos((x(1)+2*x(2))/4);
x0 = [2*pi; 0];
XJ = x0;
XS = x0;
for r = 1:15
    XJ(1,r+1) = g1(XJ(:,r));
    XJ(2,r+1) = g2(XJ(:,r));
    XS(1,r+1) = g1(XS(:,r));
    XS(2,r+1) = g2([XS(1,r+1); XS(2,r)]);
end
```

Rezultata prikazuje tabela 4.1. Napaka je izračunana kot norma razlike med trenutnim približkom in podanim natančnim približkom. Razvidno je, da so približki Seidlove iteracije boljši od približkov Jacobijeve iteracije.

The convergence of the sequence of the approximations in the Jacobi iteration depends on the properties of the mapping $\boldsymbol{g}$. Suppose $\boldsymbol{g}$ is continuously differentiable on a bounded and closed region $\Omega \subset \mathbb{R}^n$ and for every $\boldsymbol{x} \in \Omega$ satisfies $\boldsymbol{g}(\boldsymbol{x}) \in \Omega$ and $\|\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x})\| \leq m < 1$. Here, $\boldsymbol{J}_{\boldsymbol{g}}$ is the Jacobi matrix of the mapping $\boldsymbol{g}$ and $\|\cdot\|$ denotes some matrix norm. Then $\boldsymbol{g}$ is a contraction with the Lipschitz constant m and by the Banach fixed point theorem the iteration sequence converge for any initial approximation from Ω.

Exercise 4.2. Prove that the Jacobi and Seidel method for the system in Exercise 4.1 converge for any initial approximation.

Solution. Preslikava Jacobijeve iteracije

$$\boldsymbol{g}_J(x, y) = \left(\sin\left(\frac{2x-y}{4}\right), \cos\left(\frac{x+2y}{4}\right)\right)$$

korak	Jacobijeva iteracija			Seidlova iteracija		
r	x_r	y_r	napaka	x_r	y_r	napaka
1	0	0	$1.0 \cdot 10^{0}$	0	1	$4.4 \cdot 10^{-1}$
2	0	1	$4.4 \cdot 10^{-1}$	−0.247404	0.905539	$1.9 \cdot 10^{-1}$
3	−0.247404	0.877583	$2.0 \cdot 10^{-1}$	−0.342979	0.933399	$9.5 \cdot 10^{-2}$
4	−0.336406	0.929795	$1.0 \cdot 10^{-1}$	−0.393871	0.932965	$4.4 \cdot 10^{-2}$
5	−0.390019	0.928369	$4.8 \cdot 10^{-2}$	−0.417032	0.935111	$2.1 \cdot 10^{-2}$
6	−0.414234	0.933523	$2.3 \cdot 10^{-2}$	−0.428013	0.935702	$9.5 \cdot 10^{-3}$
7	−0.42639	0.934764	$1.1 \cdot 10^{-2}$	−0.433103	0.936046	$4.4 \cdot 10^{-3}$
8	−0.432159	0.93562	$5.4 \cdot 10^{-3}$	−0.435472	0.936194	$2.1 \cdot 10^{-3}$
9	−0.434952	0.935978	$2.6 \cdot 10^{-3}$	−0.436572	0.936265	$9.5 \cdot 10^{-4}$
10	−0.436289	0.93616	$1.2 \cdot 10^{-3}$	−0.437082	0.936297	$4.4 \cdot 10^{-4}$
11	−0.436932	0.936246	$6.0 \cdot 10^{-4}$	−0.437319	0.936312	$2.1 \cdot 10^{-4}$
12	−0.43724	0.936287	$2.9 \cdot 10^{-4}$	−0.437429	0.936319	$9.5 \cdot 10^{-5}$
13	−0.437388	0.936307	$1.4 \cdot 10^{-4}$	−0.43748	0.936323	$4.4 \cdot 10^{-5}$
14	−0.437459	0.936317	$6.6 \cdot 10^{-5}$	−0.437504	0.936324	$2.1 \cdot 10^{-5}$
15	−0.437493	0.936321	$3.2 \cdot 10^{-5}$	−0.437515	0.936325	$9.5 \cdot 10^{-6}$

TABELA 4.1: *Jacobijeva in Seidlova iteracija iz naloge 4.1.*

vsak $(x, y) \in \mathbb{R}^2$ preslika v $[-1, 1] \times [-1, 1]$, zato lahko analizo $\boldsymbol{g}_J$ omejimo na to območje. Jacobijeva matrika $\boldsymbol{J}_{\boldsymbol{g}_J}$ preslikave $\boldsymbol{g}_J$ je podana z

$$\boldsymbol{J}_{\boldsymbol{g}_J}(x, y) = \begin{bmatrix} \frac{1}{2}\cos\left(\frac{2x-y}{4}\right) & -\frac{1}{4}\cos\left(\frac{2x-y}{4}\right) \\ -\frac{1}{4}\sin\left(\frac{x+2y}{4}\right) & -\frac{1}{2}\sin\left(\frac{x+2y}{4}\right) \end{bmatrix}.$$

Za vsak (x, y) velja $\|\boldsymbol{J}_{\boldsymbol{g}_J}(x, y)\|_\infty \leq \frac{3}{4} < 1$, zato je $\boldsymbol{g}_J$ skrčitev in iteracijsko zaporedje po Banachovem skrčitvenem načelu konvergira za vsak začetni približek. Enak sklep velja za preslikavo

$$\boldsymbol{g}_S(x, y) = \left(\sin\left(\frac{2x-y}{4}\right), \cos\left(\frac{\sin\left(\frac{2x-y}{4}\right) + 2y}{4}\right)\right)$$

Seidlove iteracije, saj je $\|\boldsymbol{J}_{\boldsymbol{g}_S}(x, y)\|_\infty \leq \frac{3}{4} < 1$.

Methods for solving systems of non-linear equations can be applied to systems of linear equations. This is helpful especially when the system of linear equations is too large to be handled by the direct method (e.g. LU decomposition).

Exercise 4.3. Express the iteration function of the Jacobi and Seidel method for solving the system of linear equations $\boldsymbol{Ax} = \boldsymbol{b}$. Assume that all diagonal elements of $\boldsymbol{A}$ are non-zero.

Solution. Pri Jacobijevi iteraciji iteracijsko preslikavo zasnujemo tako, da v i-ti enačbi sistema $\boldsymbol{Ax} = \boldsymbol{b}$ izrazimo i-to komponento $\boldsymbol{x}_i$ vektorja $\boldsymbol{x}$. V vektorskem

smislu to pomeni, da sistem zapišemo kot $\boldsymbol{D}\boldsymbol{x} = -(\boldsymbol{A}-\boldsymbol{D})\boldsymbol{x}+\boldsymbol{b}$, kjer je $\boldsymbol{D}$ diagonalna matrika, ki ima enako diagonalo kot $\boldsymbol{A}$. Ker so vsi elementi na diagonali $\boldsymbol{A}$ različni od nič, je $\boldsymbol{D}$ obrnljiva in preurejeno enačbo lahko z leve pomnožimo z inverzom $\boldsymbol{D}$. Na ta način dobimo iteracijsko preslikavo

$$\boldsymbol{g}_J(\boldsymbol{x}) = -\boldsymbol{D}^{-1}(\boldsymbol{A}-\boldsymbol{D})\boldsymbol{x} + \boldsymbol{D}^{-1}\boldsymbol{b}.$$

Pri Seidlovi iteraciji v trenutnem koraku pri računanju posamezne komponente novega približka upoštevamo že izračunane približke v trenutnem koraku in sistem zapišemo kot $\boldsymbol{L}\boldsymbol{x} = -(\boldsymbol{A}-\boldsymbol{L})\boldsymbol{x}+\boldsymbol{b}$, kjer je $\boldsymbol{L}$ spodnje trikotna matrika, ki ima na in pod diagonalo enake elemente kot $\boldsymbol{A}$. Za iteracijsko preslikavo $\boldsymbol{g}_S$ zato vzamemo

$$\boldsymbol{g}_S(\boldsymbol{x}) = -\boldsymbol{L}^{-1}(\boldsymbol{A}-\boldsymbol{L})\boldsymbol{x} + \boldsymbol{L}^{-1}\boldsymbol{b}.$$

Zaradi predpostavke o neničelnosti diagonalnih elementov $\boldsymbol{A}$ je tudi matrika $\boldsymbol{L}$ obrnljiva. Pri izvedbi Seidlove iteracije ne računamo inverza matrike $\boldsymbol{L}$, pač pa korak iteracije izvedemo s premimi substitucijami.

Exercise 4.4. Given is the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ of linear equations determined by strictly diagonal dominant matrix $\boldsymbol{A}$. Prove that the Jacobi and Seidel method converge for any initial approximation.

Solution. Jacobijeva matrika iteracijske preslikave $\boldsymbol{g}_J$, ki je definirana v nalogi 4.3, je enaka $-\boldsymbol{D}^{-1}(\boldsymbol{A}-\boldsymbol{D})$. Označimo elemente matrike $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ z $a_{i,j}$, $i,j = 1,2,\ldots,n$. Iz stroge diagonalne dominantnosti matrike $\boldsymbol{A}$ sledi, da je $a_{i,i} \neq 0$ za vsak i, poleg tega pa za vsak $\boldsymbol{x} \in \mathbb{R}^n$ tudi

$$\|\boldsymbol{J}_{\boldsymbol{g}_J}(\boldsymbol{x})\|_\infty = \left\|-\boldsymbol{D}^{-1}(\boldsymbol{A}-\boldsymbol{D})\right\|_\infty = \max_{i=1,2,\ldots,n} \frac{\sum_{j=1,j\neq i}^{n} |a_{i,j}|}{|a_{i,i}|} < 1,$$

kar pomeni, da je $\boldsymbol{g}_J$ skrčitev.

Pri Seidlovi iteraciji Jacobijeva matrika $\boldsymbol{J}_{\boldsymbol{g}_S}(\boldsymbol{x})$ iteracijske preslikave $\boldsymbol{g}_S$ iz naloge 4.3 za vsak $\boldsymbol{x}$ ustreza matriki $\boldsymbol{J}_{\boldsymbol{g}_S} = -\boldsymbol{L}^{-1}(\boldsymbol{A}-\boldsymbol{L})$. Oglejmo si, v kakšni zvezi je približek $\boldsymbol{x}^{(r)}$ na r-tem koraku iteracije s točno rešitvijo sistema. Ker je

$$\boldsymbol{x}^{(r)} - \boldsymbol{A}^{-1}\boldsymbol{b} = \left(\boldsymbol{J}_{\boldsymbol{g}_S}\boldsymbol{x}^{(r-1)} + \boldsymbol{L}^{-1}\boldsymbol{b}\right) - \left(\boldsymbol{J}_{\boldsymbol{g}_S}\boldsymbol{A}^{-1}\boldsymbol{b} + \boldsymbol{L}^{-1}\boldsymbol{b}\right),$$

sledi $\boldsymbol{x}^{(r)} - \boldsymbol{A}^{-1}\boldsymbol{b} = \boldsymbol{J}_{\boldsymbol{g}_S}\left(\boldsymbol{x}^{(r-1)} - \boldsymbol{A}^{-1}\boldsymbol{b}\right)$. Če to zvezo uporabimo večkrat zapored, dobimo

$$\boldsymbol{x}^{(r)} - \boldsymbol{A}^{-1}\boldsymbol{b} = \boldsymbol{J}_{\boldsymbol{g}_S}\left(\boldsymbol{x}^{(r-1)} - \boldsymbol{A}^{-1}\boldsymbol{b}\right) = \ldots = \left(\boldsymbol{J}_{\boldsymbol{g}_S}\right)^r\left(\boldsymbol{x}^{(0)} - \boldsymbol{A}^{-1}\boldsymbol{b}\right).$$

Torej je $\boldsymbol{A}^{-1}\boldsymbol{b}$ limita iteracijskega zaporedja natanko tedaj, ko je $\lim_{r\to\infty}\left(\boldsymbol{J}_{\boldsymbol{g}_S}\right)^r = \boldsymbol{0}$, to pa velja natanko tedaj, ko so vse lastne vrednosti matrike $\boldsymbol{J}_{\boldsymbol{g}_S}$ po absolutni vrednosti manjše od 1 (v kar se lahko prepričamo s pomočjo Jordanove dekompozicije matrike). Obravnavajmo poljuben lastni par $(\lambda, \boldsymbol{v})$ matrike $\boldsymbol{J}_{\boldsymbol{g}_S}$. Po definiciji je

$$\lambda\boldsymbol{v} = \boldsymbol{J}_{\boldsymbol{g}_S}\boldsymbol{v} = -\boldsymbol{L}^{-1}(\boldsymbol{A}-\boldsymbol{L})\boldsymbol{v}$$

oziroma $\lambda \boldsymbol{L}\boldsymbol{v} = -(\boldsymbol{A} - \boldsymbol{L})\boldsymbol{v}$, iz česar sledi, da za vsak $i \in \{1, 2, \ldots, n\}$ velja

$$\lambda \sum_{j=1}^{i} a_{i,j}\boldsymbol{v}_j = -\sum_{j=i+1}^{n} a_{i,j}\boldsymbol{v}_j.$$

Iz teh enačb lahko izrazimo

$$-\lambda a_{i,i}\boldsymbol{v}_i = \lambda \sum_{j=1}^{i-1} a_{i,j}\boldsymbol{v}_j + \sum_{j=i+1}^{n} a_{i,j}\boldsymbol{v}_j$$

in če izberemo tisto, pri kateri je $|\boldsymbol{v}_i| = \|\boldsymbol{v}\|_\infty$, sledi

$$|\lambda|\,|a_{i,i}| \leq |\lambda| \sum_{j=1}^{i-1} |a_{i,j}|\,\frac{|\boldsymbol{v}_j|}{|\boldsymbol{v}_i|} + \sum_{j=i+1}^{n} |a_{i,j}|\,\frac{|\boldsymbol{v}_j|}{|\boldsymbol{v}_i|} \leq |\lambda| \sum_{j=1}^{i-1} |a_{i,j}| + \sum_{j=i+1}^{n} |a_{i,j}|\,.$$

Torej je

$$|\lambda| \leq \frac{\sum_{j=i+1}^{n} |a_{i,j}|}{|a_{i,i}| - \sum_{j=1}^{i-1} |a_{i,j}|}$$

in zaradi diagonalne dominantnosti velja $|\lambda| < 1$. S tem smo dokazali, da so vse lastne vrednosti matrike $\boldsymbol{J}_{\boldsymbol{g}_S}$ po absolutni vrednosti manjše od 1, kar dokazuje konvergenco iteracijskega zaporedja.

4.2. The Newton's method

The Newton's method for solving a system $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ is a generalization of the Newton's method for a single non-linear equation. We perform the iteration step

$$\boldsymbol{x}^{(r+1)} = \boldsymbol{x}^{(r)} - \boldsymbol{J}_{\boldsymbol{f}}\left(\boldsymbol{x}^{(r)}\right)^{-1} \boldsymbol{f}\left(\boldsymbol{x}^{(r)}\right)$$

by solving the system of linear equations

$$\boldsymbol{J}_{\boldsymbol{f}}\left(\boldsymbol{x}^{(r)}\right) \cdot \boldsymbol{\Delta x}^{(r)} = -\boldsymbol{f}\left(\boldsymbol{x}^{(r)}\right)$$

and then express the new approximation $\boldsymbol{x}^{(r+1)}$ from $\boldsymbol{\Delta x}^{(r)} = \boldsymbol{x}^{(r+1)} - \boldsymbol{x}^{(r)}$. In the quasi-Newton's methods the Jacobi matricx is replaced by matrices that are easier to compute. In the Broyden's method it is replaced in step r by the matrix $\boldsymbol{B}_r$ that is recursively determined by

$$\boldsymbol{B}_r = \boldsymbol{B}_{r-1} + \frac{\boldsymbol{f}\left(\boldsymbol{x}^{(r)}\right)\left(\boldsymbol{\Delta x}^{(r-1)}\right)^{\mathsf{T}}}{\left\|\boldsymbol{\Delta x}^{(r-1)}\right\|_2^2},$$

where $\boldsymbol{B}_0$ is an approximation for $\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}^{(0)})$. The Broyden's method may be seen as a generalization of the secant method for solving non-linear equations.

Exercise 4.5. Given is the system Dan je sistem

$$x^2 + y^2 = 4, \quad x^2 - y^2 = 1$$

of two non-linear equations for the unknowns x and y.

1. Interpret the system geometrically. How many solutions does it have? Find the solutions analytically.

2. Derive the Newton's method for solving the system and perform two iteration steps with the initial approximation $(x_0, y_0) = (2, 1)$.

3. Find an approximate solution to the system with two steps of the Broyden's method with the initial approximation $(x_0, y_0) = (2, 1)$ and $\boldsymbol{B}_0 = \boldsymbol{J_f}(x_0, y_0)$.

Solution.

1. Prva enačba predstavlja krožnico s polmerom 2, druga enačba pa hiperbolo. Sistem ima štiri rešitve, to so $(\pm\sqrt{5/2}, \pm\sqrt{3/2})$ in $(\pm\sqrt{5/2}, \mp\sqrt{3/2})$

2. Dani sistem enačb zapišemo v obliki $\boldsymbol{f}(x, y) = \boldsymbol{0}$ za preslikavo $\boldsymbol{f}$, podano s predpisom

$$\boldsymbol{f}(x, y) = (x^2 + y^2 - 4, x^2 - y^2 - 1).$$

   Jacobijeva matrika preslikave $\boldsymbol{f}$ je

$$\boldsymbol{J_f}(x, y) = \begin{bmatrix} 2x & 2y \\ 2x & -2y \end{bmatrix},$$

   zato v prvem koraku metode pri začetnem približku $(x_0, y_0) = (2, 1)$ rešujemo sistem

$$\begin{bmatrix} 4 & 2 \\ 4 & -2 \end{bmatrix} \begin{bmatrix} \Delta x_0 \\ \Delta y_0 \end{bmatrix} = - \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

   Izračunamo $(\Delta x_0, \Delta y_0) = (-\frac{3}{8}, \frac{1}{4})$, kar pomeni, da je novi približek

$$(x_1, y_1) = (x_0, y_0) + (\Delta x_0, \Delta y_0) = \left(\tfrac{13}{8}, \tfrac{5}{4}\right).$$

   V drugem koraku z reševanjem sistema

$$\begin{bmatrix} \frac{13}{4} & \frac{5}{2} \\ \frac{13}{4} & -\frac{5}{2} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta y_1 \end{bmatrix} = - \begin{bmatrix} \frac{13}{64} \\ \frac{5}{64} \end{bmatrix}$$

   dobimo $(\Delta x_1, \Delta y_1) = (-\frac{9}{208}, -\frac{1}{40})$, torej je

$$(x_2, y_2) = (x_1, y_1) + (\Delta x_1, \Delta y_1) = \left(\tfrac{329}{208}, \tfrac{49}{40}\right) \approx (1.5817, 1.2250).$$

   Iteracijsko zaporedje konvergira k rešitvi $(\sqrt{5/2}, \sqrt{3/2}) \approx (1.5811, 1.2247)$.

3. Zaradi izbire $\boldsymbol{B}_0$ je prvi korak Broydnove metode enak kot pri Newtonovi metodi. Za izvedbo drugega koraka najprej izračunamo

$$\boldsymbol{B}_1 = \boldsymbol{B}_0 + \frac{\boldsymbol{f}(x_1, y_1) \cdot (\Delta x_0, \Delta y_0)^{\mathsf{T}}}{\|(\Delta x_0, \Delta y_0)\|_2^2} = \begin{bmatrix} \frac{29}{8} & \frac{9}{4} \\ \frac{401}{104} & -\frac{99}{52} \end{bmatrix}.$$

Nato rešimo sistem $\boldsymbol{B}_1 \cdot (\Delta x_1, \Delta y_1) = -\boldsymbol{f}(x_1, y_1)$ in približek (x_1, y_1) izračunamo iz rešitve $(\Delta x_1, \Delta y_1) = (-\frac{13}{360}, -\frac{13}{405})$. Dobimo

$$(x_2, y_2) = \left(\tfrac{143}{90}, \tfrac{1973}{1620}\right) \approx (1.5589, 1.2179).$$

Exercise 4.6. Given are the points (x_i, y_i), $i = 1, 2, \ldots, n$, in the plane. We would like to find a function f of the form $f(x) = \alpha e^{\beta x}$, $\alpha, \beta \in \mathbb{R}$, that minimizes

$$\varphi(\alpha, \beta) = \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2 = \sum_{i=1}^{n} \left(y_i - \alpha e^{\beta x_i}\right)^2.$$

For such f we say that fits the points best in the least square sense.

1. Finding the minimum of $\varphi(\alpha, \beta)$ interpret as solving a system of non-linear equations with unknowns α and β.

2. Derive the Newton's method for the obtained system.

3. If the ordinates y_i are positive, a good initial approximation for the Newton's method can be found by the linearization of $\varphi(\alpha, \beta)$ into the form

$$\psi(\alpha, \beta) = \sum_{i=1}^{n} \left(\ln(y_i) - \ln(\alpha) - \beta x_i\right)^2.$$

Derive and solve the system of linear equations for the parameters α and β.

4. In Matlab, using the linearization and the Newton's method find and plot f that in the least square sense best fits to the points

$$(1, 8),\ (2, 9),\ (3, 6),\ (4, 7),\ (5, 4),\ (6, 3),\ (7, 2),\ (8, 3),\ (9, 2),\ (10, 1).$$

Compare the solution to the solution of the original problem computed by the built-in function `fminsearch`.

Solution.

1. Z odvajanjem φ dobimo

$$\frac{\partial \varphi}{\partial \alpha}(\alpha, \beta) = \sum_{i=1}^{n} 2\left(y_i - \alpha e^{\beta x_i}\right)\left(-e^{\beta x_i}\right),$$
$$\frac{\partial \varphi}{\partial \beta}(\alpha, \beta) = \sum_{i=1}^{n} 2\left(y_i - \alpha e^{\beta x_i}\right)\left(-\alpha x_i e^{\beta x_i}\right),$$

kar pomeni, da lahko minimum funkcije φ iščemo z reševanjem sistema nelinearnih enačb $\boldsymbol{f}(\alpha,\beta)=\mathbf{0}$ za

$$\boldsymbol{f}(\alpha,\beta)=\left(\sum_{i=1}^{n}e^{\beta x_i}\left(y_i-\alpha e^{\beta x_i}\right),\alpha\sum_{i=1}^{n}x_i e^{\beta x_i}\left(y_i-\alpha e^{\beta x_i}\right)\right).$$

2. Za izpeljavo Newtonove metode je treba določiti Jacobijevo matriko $\boldsymbol{J_f}$ preslikave $\boldsymbol{f}$. Izračunamo

$$\boldsymbol{J_f}(\alpha,\beta)=\begin{bmatrix}-\sum_{i=1}^{n}e^{2\beta x_i} & \sum_{i=1}^{n}x_i e^{\beta x_i}\left(y_i-2\alpha e^{\beta x_i}\right)\\ \sum_{i=1}^{n}x_i e^{\beta x_i}\left(y_i-2\alpha e^{\beta x_i}\right) & \alpha\sum_{i=1}^{n}x_i^2 e^{\beta x_i}\left(y_i-2\alpha e^{\beta x_i}\right)\end{bmatrix}.$$

Na vsakem koraku metode novi približek $(\alpha_{r+1},\beta_{r+1})$ izračunamo iz prejšnjega (α_r,β_r) tako, da rešimo sistem

$$\boldsymbol{J_f}(\alpha_r,\beta_r)\cdot(\Delta\alpha_r,\Delta\beta_r)=-\boldsymbol{f}(\alpha_r,\beta_r)$$

in vzamemo

$$(\alpha_{r+1},\beta_{r+1})=(\alpha_r,\beta_r)+(\Delta\alpha_r,\Delta\beta_r).$$

3. Naj bo $X_i=x_i$, $Y_i=\ln(y_i)$, $A=\ln(\alpha)$ in $B=\beta$. Tedaj je

$$\psi(A,B)=\sum_{i=1}^{n}\left(Y_i-A-BX_i\right)^2$$

in iz

$$\frac{\partial\psi}{\partial A}(A,B)=\sum_{i=1}^{n}2\left(Y_i-A-BX_i\right)(-1),$$
$$\frac{\partial\psi}{\partial B}(A,B)=\sum_{i=1}^{n}2\left(Y_i-A-BX_i\right)(-X_i)$$

podobno kot v prvi točki naloge sledi

$$\begin{bmatrix}n & \sum_{i=1}^{n}X_i\\ \sum_{i=1}^{n}X_i & \sum_{i=1}^{n}X_i^2\end{bmatrix}\begin{bmatrix}A\\B\end{bmatrix}=\begin{bmatrix}\sum_{i=1}^{n}Y_i\\ \sum_{i=1}^{n}X_iY_i\end{bmatrix}.$$

To ni nič drugega kot normalni sistem za predoločen sistem, ki določa najboljšo aproksimacijo parov (X_i,Y_i) s premico $x\mapsto A+Bx$ po metodi najmanjših kvadratov. Prvi približek za parametra α in β lahko torej določimo na podlagi rešitve

sistema

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} \ln(\alpha) \\ \beta \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \ln(y_i) \\ \sum_{i=1}^{n} x_i \ln(y_i) \end{bmatrix}.$$

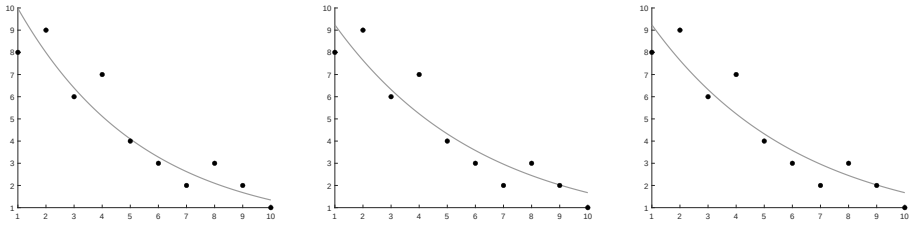
4. V Matlabu pripravimo implementacijo Newtonove metode, ki za dano preslikavo $\boldsymbol{f}$ in njeno Jacobijevo matriko vrne približek za rešitev sistema $\boldsymbol{f}(\boldsymbol{x}) = \mathbf{0}$. Funkcija deluje tako, da pri podanem začetnem približku izvede toliko iteracij, da je norma razlike dveh zaporednih približkov manjša od podane tolerance, razen če število iteracij preseže maksimalno predpisano število le-teh.

```
function [x,X,k] = newton(f,Jf,x0,tol,N)
% Opis:
%  newton izvede Newtonovo metodo za reševanje sistema
%  nelinearnih enačb
%
% Definicija:
%  [x,X,k] = newton(f,Jf,x0,tol,N)
%
% Vhodni podatki:
%  f    preslikava, ki določa sistem f(x) = 0,
%  Jf   Jacobijeva matrika preslikave f,
%  x0   začetni približek (stolpec),
%  tol  toleranca norme razlike dveh zaporednih
%       približkov,
%  N    maksimalno število korakov iteracije
%
% Izhodni podatki:
%  x    končni približek za rešitev sistema f(x) = 0,
%  X    tabela vseh izračunanih približkov,
%  k    število opravljenih korakov

X = x0;
dx = Inf;
k = 0;
while norm(dx) > tol && k < N
   k = k+1;
   dx = -Jf(X(:,k))\f(X(:,k));
   X(:,k+1) = X(:,k) + dx;
end
x = X(:,k+1);

end
```

Začetni približek za Newtonovo metodo poiščemo z linearizacijo.

```
X = (1:10)';
Y = [8; 9; 6; 7; 4; 3; 2; 3; 2; 1];

A = [10 sum(X); sum(X) X'*X];
logY = log(Y);
logx0 = A\[sum(logY); X'*logY];
x0 = [exp(logx0(1)); logx0(2)];
```

Za začetni približek vzamemo par, ki je rezultat zgornjega izračuna in je približno enak $(12.4733, -0.2228)$. Nato z ustrezno definiranima funkcijama, ki za dani vektor $\boldsymbol{x}$ vrneta vektor $\boldsymbol{f}(\boldsymbol{x})$ in matriko $\boldsymbol{J_f}(\boldsymbol{x})$, izvedemo funkcijo `newton`. Pri toleranci 10^{-10} v 5 korakih iteracije dobimo približek, ki je približno enak $(11.1713, -0.1898)$. Skoraj povsem enak približek dobimo, če rešitev problema poiščemo z vgrajeno funkcijo `fminsearch`. Kot vhodna podatka ji podamo funkcijo φ in začetni približek, ki smo ga dobili z linearizacijo. Rezultati so grafično predstavljeni na sliki 4.1.



(A) *linearizacija* (B) *Newtonova metoda* (C) *funkcija* `fminsearch`

Slika 4.1: *Grafi funkcije f, ki jih dobimo pri reševanju naloge 4.6.*

5. Overdetermined Systems

A system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ may have more equations than unknowns. This means that $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is a rectangular matrix with more rows than columns ($m > n$). A vector $\boldsymbol{x} \in \mathbb{R}^n$ satisfying the system for any $\boldsymbol{b} \in \mathbb{R}^m$ does not exist, and we usually look for $\boldsymbol{x}$ minimizing the norm of the remineder $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$..

5.1. Normal System

The problem of finding the vector $\boldsymbol{x} \in \mathbb{R}^n$ minimizing the value of $\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2$ has a unique solution if $\boldsymbol{A}$ is a full rank matrix ($\mathrm{rang}(\boldsymbol{A}) = n$). The solution is called the best least square approximation. It is determined by solving the normal system $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\mathsf{T}\boldsymbol{b}$. The matrix $\boldsymbol{A}^\mathsf{T}\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is symmetric and positive definite, hence the system can be solved by the Cholesky decomposition.

Exercise 5.1. Prove that the solution of the system

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{A} \\ \boldsymbol{A}^\mathsf{T} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix}$$

corresponds to the solution of the overdetermined system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ in the least square sense. Interpret the system geometrically.

Solution. Vektorja $\boldsymbol{r}$ in $\boldsymbol{x}$ ustrezata enakostma $\boldsymbol{r} + \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ in $\boldsymbol{A}^\mathsf{T}\boldsymbol{r} = \boldsymbol{0}$. Če v prvi izrazimo $\boldsymbol{r}$ in ga vstavimo v drugo, dobimo normalni sistem $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\mathsf{T}\boldsymbol{b}$. Vektor $\boldsymbol{r}$ predstavlja razliko med vektorjem $\boldsymbol{b}$ in vektorjem $\boldsymbol{A}\boldsymbol{x}$, ki leži v prostoru $\mathrm{im}(\boldsymbol{A})$. Z zahtevo $\boldsymbol{A}^\mathsf{T}\boldsymbol{r} = \boldsymbol{0}$ predpišemo, da je $\boldsymbol{r}$ pravokoten na $\mathrm{im}(\boldsymbol{A})$ oziroma da je $\boldsymbol{A}\boldsymbol{x}$ pravokotna projekcija $\boldsymbol{b}$ na $\mathrm{im}(\boldsymbol{A})$.

Exercise 5.2. Values of a function f are given at four points:

$$f(-1) = \tfrac{11}{4}, \quad f(0) = \tfrac{7}{4}, \quad f(1) = \tfrac{1}{4}, \quad f(2) = \tfrac{13}{4}.$$

Find the parabola that by the least square method best fits to the given values of the function f. Derive the normal system and solve it by using the Cholesky decomposition.

Solution. Parabolo predstavimo v obliki $p(x) = a_0 + a_1 x + a_2 x^2$. Veljati mora $p(x) = f(x)$ za $x \in \{-1, 0, 1, 2\}$, torej so koeficienti a_0, a_1, a_2 določeni s predoločenim sistemom

$$\begin{bmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 11 \\ 7 \\ 1 \\ 13 \end{bmatrix}.$$

Naj bo matrika sistema označena z $\boldsymbol{A}$, desna stran sistema pa z $\boldsymbol{b}$. Normalni sistem $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\mathsf{T}\boldsymbol{b}$ je podan kot

$$\begin{bmatrix} 4 & 2 & 6 \\ 2 & 6 & 8 \\ 6 & 8 & 18 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 16 \end{bmatrix}.$$

Njegova rešitev ustreza vektorju $\boldsymbol{x}$, ki minimizira $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ oziroma vsoto kvadratov razlik med vrednostmi parabole p in funkcije f v točkah -1, 0, 1, 2. Faktor Choleskega za matriko $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$ je podan z

$$\boldsymbol{V} = \begin{bmatrix} 2 & 0 & 0 \\ 1 & \sqrt{5} & 0 \\ 3 & \sqrt{5} & 2 \end{bmatrix}.$$

Ker je $\boldsymbol{A}^\mathsf{T}\boldsymbol{A} = \boldsymbol{V}\boldsymbol{V}^\mathsf{T}$, lahko koeficiente parabole izračunamo z reševanjem sistemov $\boldsymbol{V}\boldsymbol{y} = \boldsymbol{A}^\mathsf{T}\boldsymbol{b}$ in $\boldsymbol{V}^\mathsf{T}\boldsymbol{x} = \boldsymbol{y}$. Dobimo $\boldsymbol{y} = (4, 0, 2)$ in $\boldsymbol{x} = (1, -1, 1)$, kar pomeni, da je $p(x) = x^2 - x + 1$ parabola, ki se po metodi najmanjših kvadratov najbolje danim podatkom.

Exercise 5.3. Derive the normal system that determines the polynomial of degree n that is the beast least square approximation for the values of the function f at $m + 1$ points distinct $x_0, x_1, \ldots, x_m$.

Solution. Polinom zapišimo v obliki

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0.$$

Predoločen sistem določajo enačbe $p(x_i) = f(x_i)$, $i = 0, 1, \ldots, m$, kar lahko v vektorski obliki zapišemo kot

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

Normalni sistem je zato

$$\left[\sum_{k=0}^{m} x_k^{j+i} \right]_{i,j=0}^{n} \cdot \left[\; a_i \; \right]_{j=0}^{n} = \left[\sum_{k=0}^{m} x_k^j \, f(x_k) \right]_{j=0}^{n}.$$

Exercise 5.4. A planet is travelling on an elliptical orbit. Some of its positions are given in the below table. They are represented by the Cartesian coordinates (x_i, y_i), $i = 1, 2, \ldots, 10$, with respect to the orbit plane.

x_i	-0.5	-0.3	0.1	0.8	0.9	0.5	0.2	-0.1	-0.3	-0.6
y_i	-0.3	-0.7	-0.8	-0.7	-0.1	0.5	0.6	0.7	0.6	0.2

Determine the coefficients a, b, c, d, e in the quadratic form

$$ax^2 + bxy + cy^2 + dx + ey = 1$$

that minimizes the value of

$$\sum_{i=1}^{10} \left(ax_i^2 + bx_iy_i + cy_i^2 + dx_i + ey_i - 1\right)^2.$$

Interpret the problem as solving the system of an overdetermined system of equations and in Matlab plot the orbit of the planet determined by the least square method.

Solution. Iskani koeficienti a, b, c, d, e ustrezajo rešitvi predoločenega sistema

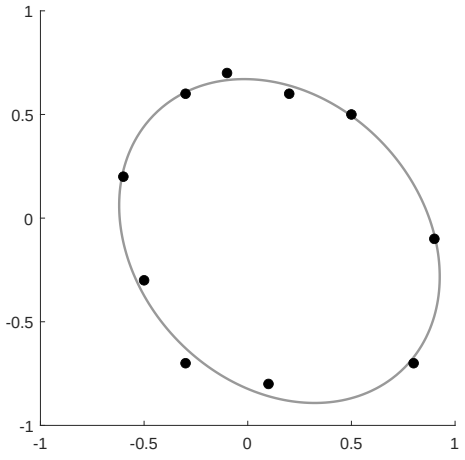
$$\begin{bmatrix} x_1^2 & x_1y_1 & y_1^2 & x_1 & y_1 \\ x_2^2 & x_2y_2 & y_2^2 & x_2 & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{10}^2 & x_{10}y_{10} & y_{10}^2 & x_{10} & y_{10} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

po metodi najmanjših kvadratov. V Matlabu lahko rešitev predoločenega sistema izračunamo kar s pomočjo operatorja \, enako kot da bi reševali sistem linearnih enačb z enakim številom enačb in neznank.

```
X = [-0.5;-0.3;0.1;0.8;0.9;0.5;0.2;-0.1;-0.3;-0.6];
Y = [-0.3;-0.7;-0.8;-0.7;-0.1;0.5;0.6;0.7;0.6;0.2];

A = [X.^2 X.*Y Y.^2 X Y];
b = ones(10,1);

k = A\b;
```

Kvadratno formo, ki je določena s koeficienti vektorja $\boldsymbol{k} = (a, b, c, d, e)$, lahko v Matlabu narišemo z uporabo vgrajenih funkcij `meshgrid` in `contour`. Skupaj s podanimi položaji planeta je prikazana na sliki 5.1.

Problems converted to solving an overdetermined system often include additional constraints. To meet these constraints an adaptation of the system solving is needed.



Slika 5.1: *Položaji planeta, podani v nalogi 5.4, in kvadratna forma, ki se jim najbolje prilega po metodi najmanjših kvadratov.*

Exercise 5.5. The overdetermined system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with the matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $m > n$, and the vector $\boldsymbol{b} \in \mathbb{R}^m$ contains additional constraints expressed by the matrix $\boldsymbol{C} \in \mathbb{R}^{p\times n}$, $p < n$, and the vector $\boldsymbol{d} \in \mathbb{R}^p$, in the form of an underdetermined system $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{d}$. Assuming that the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ have full rank, we are looking for $\boldsymbol{x} \in \mathbb{R}^n$ minimizing $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ under the additional condition $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{d}$.

1. Let $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{R}^n \times \mathbb{R}^p$ be the solution of the system

$$\begin{bmatrix} \boldsymbol{A}^\mathsf{T}\boldsymbol{A} & \boldsymbol{C}^\mathsf{T} \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}^\mathsf{T}\boldsymbol{b} \\ \boldsymbol{d} \end{bmatrix}.$$

   Prove that $\boldsymbol{x}$ is the solution of the minimization problem.

2. Argue that the solution $\boldsymbol{x}$ exists and is unique.

Solution.

1. Naj bo $\widetilde{\boldsymbol{x}} \in \mathbb{R}^n$ poljuben vektor, za katerega velja $\boldsymbol{C}\widetilde{\boldsymbol{x}} = \boldsymbol{d}$. Tudi vektor $\boldsymbol{x}$, ki je delna rešitev zgornjega bločnega sistema, zadošča pogoju $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{d}$. Dokažimo, da je $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \le \|\boldsymbol{A}\widetilde{\boldsymbol{x}} - \boldsymbol{b}\|_2$. Kvadrat $\|\boldsymbol{A}\widetilde{\boldsymbol{x}} - \boldsymbol{b}\|_2$ lahko zapišemo kot

$$\|\boldsymbol{A}(\widetilde{\boldsymbol{x}} - \boldsymbol{x}) + \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \|\boldsymbol{A}(\widetilde{\boldsymbol{x}} - \boldsymbol{x})\|_2^2 + 2(\widetilde{\boldsymbol{x}} - \boldsymbol{x})^\mathsf{T}\boldsymbol{A}^\mathsf{T}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) + \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

   Sedaj upoštevamo, da za $\boldsymbol{x}$ velja $\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{C}^\mathsf{T}\boldsymbol{y} = \boldsymbol{A}^\mathsf{T}\boldsymbol{b}$ in za člen

$$2(\widetilde{\boldsymbol{x}} - \boldsymbol{x})^\mathsf{T}\boldsymbol{A}^\mathsf{T}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = -2(\widetilde{\boldsymbol{x}} - \boldsymbol{x})^\mathsf{T}\boldsymbol{C}^\mathsf{T}\boldsymbol{y}$$

   sklepamo, da je enak 0, saj je $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{C}\widetilde{\boldsymbol{x}}$. Od tod potem takoj sledi

$$\|\boldsymbol{A}\widetilde{\boldsymbol{x}} - \boldsymbol{b}\|_2^2 = \|\boldsymbol{A}(\widetilde{\boldsymbol{x}} - \boldsymbol{x})\|_2^2 + \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \ge \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

2. Zadošča utemeljitev, da je bločna matrika, ki nastopa v sistemu iz prve točke, obrnljiva. Denimo, da ni. Potem obstaja neničeln vektor $(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{y}}) \in \mathbb{R}^n \times \mathbb{R}^p$, da je

$$\begin{bmatrix} \boldsymbol{A}^\mathsf{T}\boldsymbol{A} & \boldsymbol{C}^\mathsf{T} \\ \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{x}} \\ \widehat{\boldsymbol{y}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}.$$

Od tod sledi

$$\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\widehat{\boldsymbol{x}} + \boldsymbol{C}^\mathsf{T}\widehat{\boldsymbol{y}} = \boldsymbol{0}, \qquad \boldsymbol{C}\widehat{\boldsymbol{x}} = \boldsymbol{0}.$$

Jasno je, da je $\widehat{\boldsymbol{x}}^\mathsf{T}(\boldsymbol{A}^\mathsf{T}\boldsymbol{A}\widehat{\boldsymbol{x}} + \boldsymbol{C}^\mathsf{T}\widehat{\boldsymbol{y}}) = 0$, kar pomeni, da je $\|\boldsymbol{A}\widehat{\boldsymbol{x}}\|_2 = 0$, saj je $\boldsymbol{C}\widehat{\boldsymbol{x}} = \boldsymbol{0}$. Torej je $\boldsymbol{A}\widehat{\boldsymbol{x}} = \boldsymbol{0}$ in ker je $\boldsymbol{A}$ polnega ranga, je $\widehat{\boldsymbol{x}} = \boldsymbol{0}$. Sedaj prva enačba implicira $\boldsymbol{C}^\mathsf{T}\widehat{\boldsymbol{y}} = \boldsymbol{0}$; tudi matrika $\boldsymbol{C}$ je polnega ranga, zato mora biti $\widehat{\boldsymbol{y}} = \boldsymbol{0}$, kar je v protislovju z začetno predpostavko.

Exercise 5.6. Let $\boldsymbol{C} \in \mathbb{R}^{m\times m}$ be a symmetric positive definite matrix that determines the norm

$$\|\boldsymbol{x}\|_{\boldsymbol{C}} = \sqrt{\boldsymbol{x}^\mathsf{T}\boldsymbol{C}\boldsymbol{x}}.$$

Derive the normal system that determines the solution to the problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{C}}.$$

Solution. Ker je $\boldsymbol{C}$ simetrična pozitivno definitna, jo lahko zapišemo kot $\boldsymbol{C} = \boldsymbol{V}\boldsymbol{V}^\mathsf{T}$, kjer je $\boldsymbol{V}$ faktor Choleskega. Kvadrat norme $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{C}}$ je zato enak

$$(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})^\mathsf{T}\boldsymbol{C}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = \left(\boldsymbol{V}^\mathsf{T}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\right)^\mathsf{T}\left(\boldsymbol{V}^\mathsf{T}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\right),$$

torej ustreza kvadratu norme $\left\|\boldsymbol{V}^\mathsf{T}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{V}^\mathsf{T}\boldsymbol{b}\right\|_2$. To pomeni, da je minimalna vrednost $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_{\boldsymbol{C}}$ dosežena pri vektorju $\boldsymbol{x} \in \mathbb{R}^n$, ki je rešitev sistema

$$\left(\boldsymbol{V}^\mathsf{T}\boldsymbol{A}\right)^\mathsf{T}\left(\boldsymbol{V}^\mathsf{T}\boldsymbol{A}\right)\boldsymbol{x} = \left(\boldsymbol{V}^\mathsf{T}\boldsymbol{A}\right)^\mathsf{T}\boldsymbol{V}^\mathsf{T}\boldsymbol{b}$$

in ga lahko poenostavimo v $\boldsymbol{A}^\mathsf{T}\boldsymbol{C}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\mathsf{T}\boldsymbol{C}\boldsymbol{b}$. To je normalni sistem, ki določa rešitev problema po metodi najmanjših kvadratov v normi $\|\cdot\|_{\boldsymbol{C}}$.

5.2. QR Decomposition

To find the solution of an overdetermined system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ we usually do not solve the normal system since this can be numerically unstable when a pair of columns of the matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is almost linearly dependent. We can avoid such problems if we replace the columns of $\boldsymbol{A}$ by an orthonormal basis for $\mathrm{im}(\boldsymbol{A})$, which is achieved by the decomposition $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ of $\boldsymbol{A}$ to the orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$ ($\boldsymbol{Q}^\mathsf{T}\boldsymbol{Q} = \boldsymbol{I}$) and the upper triangular matrix $\boldsymbol{R} \in \mathbb{R}^{n\times n}$.

Exercise 5.7. Use the QR decomposition of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ to prove the Hadamard inequality

$$|\det(\boldsymbol{A})| \leq \prod_{j=1}^{n} \|\boldsymbol{a}_j\|_2 ,$$

where $\boldsymbol{a}_j$, $j = 1, 2, \ldots, n$, denote the columns of $\boldsymbol{A}$.

Solution. Če je matrika $\boldsymbol{A}$ singularna, je trditev trivialna, zato privzemimo, da je $\operatorname{rang}(\boldsymbol{A}) = n$. Naj bosta $\boldsymbol{Q}$ in $\boldsymbol{R}$ matriki QR razcepa $\boldsymbol{A} = \boldsymbol{QR}$ matrike $\boldsymbol{A}$. Ker je $\boldsymbol{Q}$ ortogonalna, je absolutna vrednost determinante $\boldsymbol{A}$ enaka determinanti $\boldsymbol{R}$, to je produktu diagonalnih elementov $r_{j,j}$, $j = 1, 2, \ldots, n$, matrike $\boldsymbol{R}$. Poleg tega iz razvoja

$$\boldsymbol{a}_j = \sum_{i=1}^{j} r_{i,j}\boldsymbol{q}_i$$

zaradi ortonormiranosti vektorjev $\boldsymbol{q}_i$ sledi, da je $|r_{j,j}| \leq \|\boldsymbol{a}_j\|_2$. S tem je neenakost dokazana.

In the modified Gram–Schmidt algorithm the QR decomposition $\boldsymbol{A} = \boldsymbol{QR}$ of the matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is computed in n steps. Let the columns of $\boldsymbol{A}$ be denoted by $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n$ and the columns of $\boldsymbol{Q}$ by $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n$. In the step $j \in \{1, 2, \ldots, n\}$ we start with the vector $\boldsymbol{a}_j$ and then subtract the orthogonal projections of the current vector to the vectors $\boldsymbol{q}_i$, $i = 1, 2, \ldots, j-1$:

$$\boldsymbol{q}_j^{(1)} = \boldsymbol{a}_j, \qquad \boldsymbol{q}_j^{(i+1)} = \boldsymbol{q}_j^{(i)} - r_{i,j}\boldsymbol{q}_i, \quad r_{i,j} = \boldsymbol{q}_i^{\mathsf{T}}\boldsymbol{q}_j^{(i)}.$$

We conclude the step by normalizing the obtained vector:

$$\boldsymbol{q}_j = \frac{1}{r_{j,j}}\boldsymbol{q}_j^{(j)}, \qquad r_{j,j} = \left\|\boldsymbol{q}_j^{(j)}\right\|_2 .$$

By this procedure, which requires approximately $2mn^2$ operations, it is ensured that the vectors $\boldsymbol{q}_j$ are orthonormal, and the values $r_{i,j}$, $i = 1, 2, \ldots, j$, corresponding to the elements of $\boldsymbol{R}$ determine the expansion of $\boldsymbol{a}_j$ over $\boldsymbol{q}_i$. The algorithm also ensures that the diagonal elements of $\boldsymbol{R}$ are positive.

Exercise 5.8. Let

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -3 \\ 0 & 1 & 1 \\ 0 & -1 & 1 \end{bmatrix}.$$

Compute the QR decomposition of $\boldsymbol{A}$ by the modified Gram–Schmidt algorithm.

Solution. QR razcep $\boldsymbol{A} = \boldsymbol{QR}$ matrike $\boldsymbol{A}$ s stolpci

$$\boldsymbol{a}_1 = (1, 1, 0, 0), \quad \boldsymbol{a}_2 = (0, 0, 1, -1), \quad \boldsymbol{a}_3 = (-1, -3, 1, 1)$$

izračunamo v treh korakih.

1. korak: Definiramo $\boldsymbol{q}_1^{(1)} = \boldsymbol{a}_1$ in izračunamo

$$i = 1: \quad r_{1,1} = \left\|\boldsymbol{q}_1^{(1)}\right\|_2 = \sqrt{2}, \qquad \boldsymbol{q}_1 = \frac{1}{r_{1,1}}\boldsymbol{q}_1^{(1)} = \frac{\sqrt{2}}{2}(1,1,0,0).$$

2. korak: Definiramo $\boldsymbol{q}_2^{(1)} = \boldsymbol{a}_2$ in izračunamo

$$i = 1: \quad r_{1,2} = \boldsymbol{q}_1^{\mathsf{T}}\boldsymbol{q}_2^{(1)} = 0, \qquad \boldsymbol{q}_2^{(2)} = \boldsymbol{q}_2^{(1)} - r_{1,2}\boldsymbol{q}_1 = \boldsymbol{q}_2^{(1)},$$

$$i = 2: \quad r_{2,2} = \left\|\boldsymbol{q}_2^{(2)}\right\|_2 = \sqrt{2}, \qquad \boldsymbol{q}_2 = \frac{1}{r_{2,2}}\boldsymbol{q}_2^{(2)} = \frac{\sqrt{2}}{2}(0,0,1,-1).$$

3. korak: Definiramo $\boldsymbol{q}_3^{(1)} = \boldsymbol{a}_3$ in izračunamo

$$i = 1: \quad r_{1,3} = \boldsymbol{q}_1^{\mathsf{T}}\boldsymbol{q}_3^{(1)} = -2\sqrt{2}, \qquad \boldsymbol{q}_3^{(2)} = \boldsymbol{q}_3^{(1)} - r_{1,3}\boldsymbol{q}_1 = (1,-1,1,1),$$

$$i = 2: \quad r_{2,3} = \boldsymbol{q}_2^{\mathsf{T}}\boldsymbol{q}_3^{(2)} = 0, \qquad \boldsymbol{q}_3^{(3)} = \boldsymbol{q}_3^{(2)} - r_{2,3}\boldsymbol{q}_2 = (1,-1,1,1),$$

$$i = 3: \quad r_{3,3} = \left\|\boldsymbol{q}_3^{(3)}\right\|_2 = 2, \qquad \boldsymbol{q}_3 = \frac{1}{r_{3,3}}\boldsymbol{q}_3^{(3)} = \frac{1}{2}(1,-1,1,1).$$

Ortogonalna matrika $\boldsymbol{Q}$ je določena s stolpci $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, $\boldsymbol{q}_3$, zgornje trikotna matrika $\boldsymbol{R}$ pa z elementi $r_{i,j}$, $1 \le i \le j \le 3$:

$$\boldsymbol{Q} = \frac{1}{2}\begin{bmatrix} \sqrt{2} & 0 & 1 \\ \sqrt{2} & 0 & -1 \\ 0 & \sqrt{2} & 1 \\ 0 & -\sqrt{2} & 1 \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} \sqrt{2} & 0 & -2\sqrt{2} \\ & \sqrt{2} & 0 \\ & & 2 \end{bmatrix}.$$

In theory the coefficient $r_{i,j}$ of $\boldsymbol{R}$ could be computed as the dot product of $\boldsymbol{q}_i$ and $\boldsymbol{a}_j$. This is the classical Gram–Schmidt algorithm, which is numerically less precise. As it turns out, it is better to correct the orthogonality with respect to the vectors containing the rounding errors. For the same reason we do not solve the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ by decomposing $\boldsymbol{A}$ and converting to the (adapted normal) system $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{Q}^{\mathsf{T}}\boldsymbol{b}$ but instead compute the QR decomposition of the extended matrix

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{q}_{n+1} \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{z} \\ & \rho \end{bmatrix},$$

where $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$, $\boldsymbol{q}_{n+1} \in \mathbb{R}^m$, $\boldsymbol{R} \in \mathbb{R}^{n\times n}$, $\boldsymbol{z} \in \mathbb{R}^n$, and $\rho \in \mathbb{R}$. From

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{q}_{n+1} \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{z} \\ & \rho \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ -1 \end{bmatrix} = \boldsymbol{Q}(\boldsymbol{R}\boldsymbol{x} - \boldsymbol{z}) - \rho\boldsymbol{q}_{n+1} \end{aligned}$$

and the fact that $\boldsymbol{q}_{n+1}$ is by construction orthogonal to the columns of $\boldsymbol{Q}$, it then follows that the value of $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ is minimal if and only if $\boldsymbol{x}$ is the solution to the system $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{z}$. The minimal value is equal to $|\rho| = \rho$.

Exercise 5.9. Let $\boldsymbol{A}$ be the matrix from Exercise 5.8 and $\boldsymbol{b} = (4, 6, -1, 2)$. Continue the modified Gram–Schmidt algorithm on the matrix $\boldsymbol{A}$ and find the QR decomposition of $\boldsymbol{A}$ extended by the vector $\boldsymbol{b}$. With the result compute the solution of the overdetermined system $\boldsymbol{Ax} = \boldsymbol{b}$.

Solution. Nadaljujemo postopek iz naloge 5.8 s še enim korakom.

4. korak: Definiramo $\boldsymbol{q}_4^{(1)} = \boldsymbol{b}$ in izračunamo

$$\begin{aligned}
i = 1: &\quad r_{1,4} = \boldsymbol{q}_1^{\mathsf{T}} \boldsymbol{q}_4^{(1)} = 5\sqrt{2}, &\quad \boldsymbol{q}_4^{(2)} &= \boldsymbol{q}_4^{(1)} - r_{1,4}\boldsymbol{q}_1 = (-1, 1, -1, 2),\\
i = 2: &\quad r_{2,4} = \boldsymbol{q}_2^{\mathsf{T}} \boldsymbol{q}_4^{(2)} = -\frac{3\sqrt{2}}{2}, &\quad \boldsymbol{q}_4^{(3)} &= \boldsymbol{q}_4^{(2)} - r_{2,4}\boldsymbol{q}_2 = \frac{1}{2}(-2, 2, 1, 1),\\
i = 3: &\quad r_{3,4} = \boldsymbol{q}_3^{\mathsf{T}} \boldsymbol{q}_4^{(3)} = -\frac{1}{2}, &\quad \boldsymbol{q}_4^{(4)} &= \boldsymbol{q}_4^{(3)} - r_{3,4}\boldsymbol{q}_3 = \frac{1}{4}(-3, 3, 3, 3),\\
i = 4: &\quad r_{4,4} = \left\| \boldsymbol{q}_4^{(4)} \right\|_2 = \frac{3}{2}, &\quad \boldsymbol{q}_4 &= \frac{1}{r_{4,4}} \boldsymbol{q}_4^{(4)} = \frac{1}{2}(-1, 1, 1, 1).
\end{aligned}$$

Velja torej

$$\left[\begin{array}{cc} \boldsymbol{A} & \boldsymbol{b} \end{array}\right] = \frac{1}{2} \begin{bmatrix} \sqrt{2} & 0 & 1 & -1 \\ \sqrt{2} & 0 & -1 & 1 \\ 0 & \sqrt{2} & 1 & 1 \\ 0 & -\sqrt{2} & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \sqrt{2} & 0 & -2\sqrt{2} & 5\sqrt{2} \\ & \sqrt{2} & 0 & -\frac{3\sqrt{2}}{2} \\ & & 2 & -\frac{1}{2} \\ & & & \frac{3}{2} \end{bmatrix},$$

iz česar sklepamo, da je $\min_{\boldsymbol{x} \in \mathbb{R}^3} \|\boldsymbol{Ax} - \boldsymbol{b}\|_2 = \frac{3}{2}$. Vektor $\boldsymbol{x} \in \mathbb{R}^3$, pri katerem je minimum dosežen, izračunamo z reševanjem sistema

$$\begin{bmatrix} \sqrt{2} & 0 & -2\sqrt{2} \\ & \sqrt{2} & 0 \\ & & 2 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 5\sqrt{2} \\ -\frac{3\sqrt{2}}{2} \\ -\frac{1}{2} \end{bmatrix}$$

in je enak $\boldsymbol{x} = \frac{1}{4}(18, -6, -1)$.

Exercise 5.10. Using the modified Gram–Schmidt algorithm determine the function f of the form

$$f(x) = a + bx^2 + c \sin(\tfrac{\pi x}{3}), \quad a, b, c \in \mathbb{R},$$

that by the least square method best fits to the points $(-2, 2)$, $(-1, -2)$, $(1, 2)$, $(2, 4)$. Are the parameters a, b, c uniquely determined? Does the same hold also if the function $x \mapsto \sin(\frac{\pi x}{3})$ in the expression for f is replaced by $x \mapsto \cos(\frac{\pi x}{3})$?

Solution. V prvem delu naloge iščemo rešitev predoločenega sistema, ki je določen z enačbami

$$a + 4b - \frac{\sqrt{3}}{2}c = 2, \quad a + b - \frac{\sqrt{3}}{2}c = -2, \quad a + b + \frac{\sqrt{3}}{2}c = 2, \quad a + 4b + \frac{\sqrt{3}}{2}c = 4.$$

Z modificiranim Gram–Schmidtovim postopkom ga prevedemo v (prilagojeni normalni) sistem

$$\begin{bmatrix} 2 & 5 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}$$

z enolično rešitvijo $a = -1$, $b = 1$, $c = \sqrt{3}$. Torej je $f(x) = -1 + x^2 + \sqrt{3}\sin(\frac{\pi x}{3})$ iskana funkcija.

Če v funkciji f zamenjamo $x \mapsto \sin(\frac{\pi x}{3})$ z $x \mapsto \cos(\frac{\pi x}{3})$, dobimo predoločeni sistem

$$a + 4b - \frac{1}{2}c = 2, \quad a + b + \frac{1}{2}c = -2, \quad a + b + \frac{1}{2}c = 2, \quad a + 4b - \frac{1}{2}c = 4,$$

ki ni polnega ranga, saj leve strani enačb ustrezajo le dvema različnima izrazoma. Če uporabimo modificirani Gram–Schmidtov postopek, dobimo (prilagojeni normalni) sistem

$$\begin{bmatrix} 2 & 5 & 0 \\ 0 & 3 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix},$$

kar pomeni, da rešitvi problema ustreza funkcija $f(x) = a + bx^2 + c\cos(\frac{\pi x}{3})$ pri vseh parametrih a, b, c, ki zadoščajo zvezama $2a + 5b = 3$ in $3b - c = 3$.

To compute the QR decomposition of a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ or to solve an overdetermined system with a matrix $\boldsymbol{A}$ the Givens rotations can be used. The principal idea is to transform the matrix by using a plane rotation to eliminate one of its elements. To eliminate the element $a_{i,j}$ of $\boldsymbol{A}$ at the position (i, j), $i > j$, we rotate the vector $(a_{j,j}, a_{i,j})$, where $a_{j,j}$ denotes the diagonal element of $\boldsymbol{A}$ at the position (j, j), to the vector $(r, 0)$, $r = \|(a_{j,j}, a_{i,j})\|_2$. This rotation can be described by the matrix $\boldsymbol{R}^{\mathsf{T}}_{j,i} \in \mathbb{R}^{m\times m}$, which is equal to the identity matrix everywhere, except in the rows i and j. In these two rows the non-zero elements are given by

$$\boldsymbol{R}^{\mathsf{T}}_{j,i}([j, i], [j, i]) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

with $c = a_{j,j}/r$ and $s = a_{i,j}/r$. The values c and s can be interpreted as $\cos(\varphi)$ and $\sin(\varphi)$, where φ denotes the rotation angle in the negative (i.e. clockwise) direction. The matrix $\boldsymbol{R}^{\mathsf{T}}_{j,i}$ applied to $\boldsymbol{A}$ affects only the rows i and j, the elements in other rows remain unchanged. Hence, by applying successive Givens rotations, $\boldsymbol{A}$ can be transformed into the upper trapezoidal form, the upper triangle of which is the triangular matrix from the QR decomposition of $\boldsymbol{A}$.

Exercise 5.11. Let $\boldsymbol{a} = (3, -4, 12, -84)$ and $\boldsymbol{b} = (7, -1, 5, 0)$. Use Givens rotations to compute $\min_{x\in\mathbb{R}} \|\boldsymbol{a}x - \boldsymbol{b}\|_2$ and determine $x \in \mathbb{R}$ at which the minimum is attained.

Solution. Konstruiramo tri Givensove rotacije, s katerimi v vektorju $\boldsymbol{a}$ po vrsti izničimo elemente na drugem, tretjem in četrtem mestu.

1. korak: Z rotacijo $\boldsymbol{R}_{1,2}^{\mathsf{T}}$ izničimo element $\boldsymbol{a}$ na drugem mestu.

$$\boldsymbol{R}_{1,2}^{\mathsf{T}} = \begin{bmatrix} \frac{3}{5} & -\frac{4}{5} & & \\ \frac{4}{5} & \frac{3}{5} & & \\ & & 1 & \\ & & & 1 \end{bmatrix} : \quad \begin{aligned} \boldsymbol{a}^{(1)} &= \boldsymbol{R}_{1,2}^{\mathsf{T}}\boldsymbol{a} = (5, 0, 12, -84), \\ \boldsymbol{b}^{(1)} &= \boldsymbol{R}_{1,2}^{\mathsf{T}}\boldsymbol{b} = (5, 5, 5, 0). \end{aligned}$$

2. korak: Z rotacijo $\boldsymbol{R}_{1,3}^{\mathsf{T}}$ izničimo element $\boldsymbol{a}^{(1)}$ na tretjem mestu.

$$\boldsymbol{R}_{1,3}^{\mathsf{T}} = \begin{bmatrix} \frac{5}{13} & & \frac{12}{13} & \\ & 1 & & \\ -\frac{12}{13} & & \frac{5}{13} & \\ & & & 1 \end{bmatrix} : \quad \begin{aligned} \boldsymbol{a}^{(2)} &= \boldsymbol{R}_{1,3}^{\mathsf{T}}\boldsymbol{a}^{(1)} = (13, 0, 0, -84), \\ \boldsymbol{b}^{(2)} &= \boldsymbol{R}_{1,3}^{\mathsf{T}}\boldsymbol{b}^{(1)} = (85/13, 5, -35/13, 0). \end{aligned}$$

3. korak: Z rotacijo $\boldsymbol{R}_{1,4}^{\mathsf{T}}$ izničimo element $\boldsymbol{a}^{(2)}$ na četrtem mestu.

$$\boldsymbol{R}_{1,4}^{\mathsf{T}} = \begin{bmatrix} \frac{13}{85} & & & -\frac{84}{85} \\ & 1 & & \\ & & 1 & \\ \frac{84}{85} & & & \frac{13}{85} \end{bmatrix} : \quad \begin{aligned} \boldsymbol{a}^{(3)} &= \boldsymbol{R}_{1,4}^{\mathsf{T}}\boldsymbol{a}^{(2)} = (85, 0, 0, 0), \\ \boldsymbol{b}^{(3)} &= \boldsymbol{R}_{1,4}^{\mathsf{T}}\boldsymbol{b}^{(2)} = (1, 5, -35/13, 84/13). \end{aligned}$$

Ker so matrike rotacij ortogonalne matrike, je

$$\|\boldsymbol{a}x - \boldsymbol{b}\|_2 = \left\|\boldsymbol{R}_{1,4}^{\mathsf{T}}\boldsymbol{R}_{1,3}^{\mathsf{T}}\boldsymbol{R}_{1,2}^{\mathsf{T}}(\boldsymbol{a}x - \boldsymbol{b})\right\|_2 = \left\|\boldsymbol{a}^{(3)}x - \boldsymbol{b}^{(3)}\right\|_2 .$$

Sledi, da je $\min_{x\in\mathbb{R}} \|\boldsymbol{a}x - \boldsymbol{b}\|_2$ dosežen pri rešitvi enačbe $85x = 1$, to je $x = 1/85$. Minimum ustreza normi vektorja $(5, -35/13, 84/13)$, ki je enaka $\sqrt{74}$.

Exercise 5.12. Derive an algorithm for solving the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}$ with a tridiagonal matrix

$$\boldsymbol{A} = \begin{bmatrix} a_1 & b_1 & & & \\ c_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-2} & a_{n-1} & b_{n-1} \\ & & & c_{n-1} & a_n \end{bmatrix} \in \mathbb{R}^{n\times n},$$

which is based on Givens rotations, and implement it in Matlab. How many operations are needed to compute $\boldsymbol{x}$?

Solution. Razširjeno matriko $[\boldsymbol{A}\ \boldsymbol{z}]$ želimo z Givensovimi rotacijami preoblikovati tako, da bo na mestu matrike $\boldsymbol{A}$ zgornje trikotna matrika. Reševanje sistema, ki

ustreza postopku izračuna QR razcepa razširjene matrike, lahko torej zastavimo tako, da najprej določimo rotacijo $\boldsymbol{R}_{1,2}^{\mathsf{T}}$, s katero izničimo element c_1 na mestu $(2,1)$. Spremenita se prva in druga vrstica razširjene matrike, ostale vrstice ostanejo nespremenjene. Zato lahko nadaljujemo z rotacijami $\boldsymbol{R}_{j,j+1}^{\mathsf{T}}$, $j = 2, 3, \ldots, n-1$, ki jih po enakem razmisleku določimo tako, da izničijo elemente c_j na mestih $(j+1, j)$.

Pri implementaciji metode za reševanje sistema $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}$ v Matlabu pazimo tako na računsko kot prostorsko učinkovitost. Metoda sprejme matriko $\boldsymbol{A}$ v obliki treh seznamov: seznama diagonalnih elementov $\boldsymbol{a}$ ter seznamov $\boldsymbol{b}$ in $\boldsymbol{c}$, ki predstavljata naddiagonalo in poddiagonalo matrike $\boldsymbol{A}$. Pri transformaciji matrike $\boldsymbol{A}$ v zgornje trikotno ne konstruiramo matrik Givensovih rotacij, temveč v vsakem izmed $n-1$ korakov določimo le c in s ter popravimo 5 elementov matrike (v koraku $n-1$ samo 3). Zgornje trikotna matrika $\boldsymbol{R}$, ki v razširjeni matriki na koncu nadomesti matriko $\boldsymbol{A}$, ima neničelne elemente le na diagonali in na dveh naddiagonalah, zato jo lahko predstavimo s seznami $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ in $\boldsymbol{r}_3$, ki so po vrsti dolžin n, $n-1$ in $n-2$. Na vsakem koraku postopka na podoben način kot elemente spreminjajoče se matrike $\boldsymbol{A}$ popravimo tudi dva elementa v vektorju $\boldsymbol{z}$.

```
n = length(a);

% diagonala in naddiagonali matrike R
r1 = a;
r2 = b;
r3 = zeros(n-2,1);

for j = 1:n-1
    r = sqrt(r1(j)^2 + c(j)^2);
    if r > 0
        C = r1(j)/r; S = c(j)/r;

        % transformacija matrike sistema
        r1(j) = r;
        r1(j+1) = -S*r2(j) + C*a(j+1);
        r2(j) = C*r2(j) + S*a(j+1);
        if j < n-1
            r2(j+1) = C*b(j+1);
            r3(j) = S*b(j+1);
        end

        % transformacija desne strani sistema
        z([j j+1]) = [C S; -S C]*z([j j+1]);
    end
end
```

Na koncu iz razširjene matrike $[\boldsymbol{A}\ \boldsymbol{z}]$ dobimo razširjeno matriko $[\boldsymbol{R}\ \widetilde{\boldsymbol{z}}]$. Ortogonalna matrika $\boldsymbol{Q}$ iz QR razcepa ustreza transponirani matriki, ki je produkt uporabljenih

Givensovih rotacij, a je ni treba eksplicitno izračunati. Matrika $\boldsymbol{Q}$ podaja povezavo med razširjenima matrikama $[\boldsymbol{A}\ \boldsymbol{z}]$ in $[\boldsymbol{R}\ \widetilde{\boldsymbol{z}}]$, saj velja $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ in $\boldsymbol{z} = \boldsymbol{Q}\widetilde{\boldsymbol{z}}$. Rešitev sistema $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}$ lahko tako izračunamo z reševanjem sistema $\boldsymbol{R}\boldsymbol{x} = \widetilde{\boldsymbol{z}}$ z obratno subsitucijo.

```
x = zeros(n,1);

x(n) = z(n)/r1(n);

if n > 1
    x(n-1) = (z(n-1) - r2(n-1)*x(n))/r1(n-1);
end

for i = n-2:-1:1
    x(i) = (z(i) - r2(i)*x(i+1) - r3(i)*x(i+2))/r1(i);
end
```

Preštejemo, da je za celoten postopek izračuna $\boldsymbol{x}$ po zgornji implementaciji potrebnih $25n-28$ osnovnih računskih operacij ($n > 1$). Tudi prostorska zahtevnost je linearna glede na n.

Exercise 5.13. Given is the QR decomposition $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ of the matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $m \geq n$. Use Givens rotations to compose an efficient algorithm for the QR decomposition of the matrix

$$\boldsymbol{A}' = \begin{bmatrix} \boldsymbol{a}^{\mathsf{T}} \\ \boldsymbol{A} \end{bmatrix}, \qquad \boldsymbol{a} \in \mathbb{R}^n.$$

Count the number of operations that the proposed algorithm requires.

Solution. Matriko $\boldsymbol{A}'$ zapišemo kot

$$\boldsymbol{A}' = \begin{bmatrix} \boldsymbol{a}^{\mathsf{T}} \\ \boldsymbol{Q}\boldsymbol{R} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \\ & \boldsymbol{Q} \end{bmatrix}}_{\boldsymbol{U}} \underbrace{\begin{bmatrix} \boldsymbol{a}^{\mathsf{T}} \\ \boldsymbol{R} \end{bmatrix}}_{\boldsymbol{H}}, \qquad \boldsymbol{U} \in \mathbb{R}^{(m+1)\times(n+1)},\ \boldsymbol{H} \in \mathbb{R}^{(n+1)\times n}.$$

Matrika, označena z $\boldsymbol{U}$, je sestavljena iz ortogonalnih stolpcev. Matrika $\boldsymbol{H}$ je zgornja Hessenbergova (to pomeni, da so vsi elementi pod diagonalo matrike enaki nič, razen tistih, ki se nahajajo neposredno pod diagonalo) in jo lahko z uporabo Givensovih rotacij transformiramo v zgornje trapezno. Naj bo $\boldsymbol{H}^{(0)} = \boldsymbol{H}$ in

$$\boldsymbol{H}^{(j)} = \boldsymbol{R}_{j,j+1}^{\mathsf{T}} \boldsymbol{H}^{(j-1)}, \quad j = 1, 2, \ldots, n,$$

kjer je $\boldsymbol{R}_{j,j+1}^{\mathsf{T}} \in \mathbb{R}^{(n+1)\times(n+1)}$ Givensova rotacija, ki v matriki $\boldsymbol{H}^{(j-1)}$ izniči element na mestu $(j+1, j)$. Pri $j = n$ dobimo

$$\boldsymbol{H}^{(n)} = \boldsymbol{R}_{n,n+1}^{\mathsf{T}} \ldots \boldsymbol{R}_{2,3}^{\mathsf{T}} \boldsymbol{R}_{1,2}^{\mathsf{T}} \boldsymbol{H} = \begin{bmatrix} \boldsymbol{R}' \\ 0 \end{bmatrix},$$

kjer $\boldsymbol{R}' \in \mathbb{R}^{n\times n}$ označuje zgornje trikotno matriko z neničelnimi diagonalnimi elementi. Vpeljimo še matrike $\boldsymbol{U}^{(0)} = \boldsymbol{U}$ in

$$\boldsymbol{U}^{(j)} = \boldsymbol{U}^{(j-1)}\boldsymbol{R}_{j,j+1}, \quad j = 1, 2, \ldots, n.$$

Pri $j = n$ velja

$$\boldsymbol{U}^{(n)} = \boldsymbol{U}\boldsymbol{R}_{1,2}\boldsymbol{R}_{2,3}\ldots\boldsymbol{R}_{n,n+1} = \left[\begin{array}{cc} \boldsymbol{Q}' & \boldsymbol{q}' \end{array}\right],$$

kjer $\boldsymbol{Q}' \in \mathbb{R}^{(m+1)\times n}$ označuje ortogonalno matriko in $\boldsymbol{q}' \in \mathbb{R}^{m+1}$ vektor, ki je pravokoten na stolpce matrike $\boldsymbol{Q}'$. Zaradi ortogonalnosti Givensovih rotacij velja

$$\boldsymbol{A}' = \boldsymbol{U}\boldsymbol{H} = \boldsymbol{U}^{(n)}\boldsymbol{H}^{(n)} = \boldsymbol{Q}'\boldsymbol{R}',$$

torej matriki $\boldsymbol{Q}'$ in $\boldsymbol{R}'$ določata QR razcep matrike $\boldsymbol{A}'$.

Povzemimo postopek izračuna matrik $\boldsymbol{Q}'$ in $\boldsymbol{R}'$ ter preštejmo število računskih operacij, potrebnih za njegovo izvedbo. Za vsak $j \in \{1, 2, \ldots, n\}$ pripravimo Givensovo rotacijo $\boldsymbol{R}^{\mathsf{T}}_{j,j+1}$, ki je določena z vrednostma c in s v vrsticah in stolpcih j in $j + 1$:

$$\boldsymbol{R}^{\mathsf{T}}_{j,j+1}([j, j+1], [j, j+1]) = \left[\begin{array}{cc} c & s \\ -s & c \end{array}\right].$$

Izračun vrednosti c in s terja 6 operacij. Rotacijo uporabimo za izračun matrike $\boldsymbol{H}^{(j)}$, ki jo dobimo z

$$\boldsymbol{H}^{(j)}([j, j+1], j : n) = \left[\begin{array}{cc} c & s \\ -s & c \end{array}\right] \boldsymbol{H}^{(j-1)}([j, j+1], j : n),$$

za kar je potrebnih $6(n - j)$ operacij. Poleg tega s to rotacijo izračunamo tudi $\boldsymbol{U}^{(j)}$ po formuli

$$\boldsymbol{U}^{(j)}(1 : (m+1), [j, j+1]) = \boldsymbol{U}^{(j-1)}(1 : (m+1), [j, j+1]) \left[\begin{array}{cc} c & -s \\ s & c \end{array}\right],$$

za kar je potrebnih $6(m + 1)$ operacij. Celoten postopek torej zahteva

$$\sum_{j=1}^{n} (6 + 6(n - j) + 6(m + 1)) = 6mn + 3n^2 + 9n$$

operacij.

The use of Givens rotations hints to an alternative form of the QR decomposition, the so-called extended QR decomposition $\boldsymbol{A} = \widetilde{\boldsymbol{Q}}\widetilde{\boldsymbol{R}}$ of the matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. In this form $\widetilde{\boldsymbol{Q}} \in \mathbb{R}^{m\times m}$ is an orthogonal matrix, the columns of which correspond to the orthonormal basis of $\mathbb{R}^m$, and $\widetilde{\boldsymbol{R}} \in \mathbb{R}^{m\times n}$ is an upper trapezoidal matrix. In solving the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ the matrix $\widetilde{\boldsymbol{Q}}$ is usually not computed and is only implicitly constructed in the process. Nonetheless, the procedure based on Givens rotations is computationally more expensive than the Gram–Schmidt algorithm, unless the matrix $\boldsymbol{A}$ has many zero elements. But instead of rotations the reflections can be used to eliminate several elements of the matrix at once.

Using the Householder reflections, the overdetermined system $\boldsymbol{Ax} = \boldsymbol{b}$ can be solved by multiplying the extended matrix $[\boldsymbol{A}\ \boldsymbol{b}]$ by orthogonal transformations. In the step $k \in \{1, 2, \ldots, n\}$ we eliminate all elements in the column k that lay below the k-th diagonal element of the current matrix. This way we gradually obtain an upper trapezoidal matrix corresponding to the exteded upper triangular matrix of the QR decomposition, while the composition of the transformations determine the orthogonal matrix. An individual orthogonal transformation is defined as a reflection over a suitably chosen hyperplane. To eliminate all but first element of the vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, we reflect it over $\mathrm{im}(\boldsymbol{w})^\perp$ where

$$\boldsymbol{w} = (x_1 + \mathrm{sign}(x_1)\,\|x\|_2\,, x_2, \ldots, x_n).$$

The reflection matrix is given by

$$\boldsymbol{P} = \boldsymbol{I} - \frac{2}{\boldsymbol{w}^\mathsf{T}\boldsymbol{w}}\boldsymbol{w}\boldsymbol{w}^\mathsf{T},$$

but we usually do not need to compute it since the transformation $\boldsymbol{Pv}$ of the vector $\boldsymbol{v}$ can be expressed as

$$\boldsymbol{Pv} = \boldsymbol{v} - \frac{2}{\boldsymbol{w}^\mathsf{T}\boldsymbol{w}}(\boldsymbol{w}^\mathsf{T}\boldsymbol{v})\boldsymbol{w}.$$

The vector $\boldsymbol{x}$ is transformed into $\boldsymbol{Px} = (-\mathrm{sign}(x_1)\,\|x\|_2\,, 0, \ldots, 0)$.

Exercise 5.14. Let

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & -1 \\ 1 & -4 & 1 \\ 1 & -4 & 3 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 0 \\ 3 \\ 0 \\ 3 \end{bmatrix}.$$

Use Householder reflections to solve the overdetermined system $\boldsymbol{Ax} = \boldsymbol{b}$. What is the value of $\min_{\boldsymbol{x}\in\mathbb{R}^3} \|\boldsymbol{Ax} - \boldsymbol{b}\|_2$?

Solution.

1. korak: Najprej zrcalimo prvi stolpec matrike $\boldsymbol{A}$. Householderjevo zrcaljenje $\boldsymbol{P}_1$ je določeno z vektorjem $\boldsymbol{w}_1 = (3, 1, 1, 1)$. Ker je

$$\begin{aligned}
\boldsymbol{P}_1\boldsymbol{a}_1 &= \boldsymbol{a}_1 - \frac{2}{\boldsymbol{w}_1^\mathsf{T}\boldsymbol{w}_1}\boldsymbol{w}_1(\boldsymbol{w}_1^\mathsf{T}\boldsymbol{a}_1) = (-2, 0, 0, 0), & \boldsymbol{a}_1 &= (1, 1, 1, 1), \\
\boldsymbol{P}_1\boldsymbol{a}_2 &= \boldsymbol{a}_2 - \frac{2}{\boldsymbol{w}_1^\mathsf{T}\boldsymbol{w}_1}\boldsymbol{w}_1(\boldsymbol{w}_1^\mathsf{T}\boldsymbol{a}_2) = (2, 2, -4, -4), & \boldsymbol{a}_2 &= (2, 2, -4, -4), \\
\boldsymbol{P}_1\boldsymbol{a}_3 &= \boldsymbol{a}_3 - \frac{2}{\boldsymbol{w}_1^\mathsf{T}\boldsymbol{w}_1}\boldsymbol{w}_1(\boldsymbol{w}_1^\mathsf{T}\boldsymbol{a}_3) = (-2, -2, 0, 2), & \boldsymbol{a}_3 &= (1, -1, 1, 3), \\
\boldsymbol{P}_1\boldsymbol{a}_4 &= \boldsymbol{a}_4 - \frac{2}{\boldsymbol{w}_1^\mathsf{T}\boldsymbol{w}_1}\boldsymbol{w}_1(\boldsymbol{w}_1^\mathsf{T}\boldsymbol{a}_4) = (-3, 2, -1, 2), & \boldsymbol{a}_4 &= (0, 3, 0, 3),
\end{aligned}$$

se prvotni predoločeni sistem preoblikuje v

$$\begin{bmatrix} -2 & 2 & -2 \\ 0 & 2 & -2 \\ 0 & -4 & 0 \\ 0 & -4 & 2 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} -3 \\ 2 \\ -1 \\ 2 \end{bmatrix}$$

2. korak: Zrcalimo vektor $(2, -4, -4)$. Householderjevo zrcaljenje $\boldsymbol{P}_2$ je določeno z vektorjem $\boldsymbol{w}_2 = (8, -4, -4)$. Iz

$$\begin{aligned} \boldsymbol{P}_2\boldsymbol{a}_1 &= \boldsymbol{a}_1 - \frac{2}{\boldsymbol{w}_2^{\mathsf{T}}\boldsymbol{w}_2}\boldsymbol{w}_2(\boldsymbol{w}_2^{\mathsf{T}}\boldsymbol{a}_1) = (-6, 0, 0), & \boldsymbol{a}_1 &= (2, -4, -4), \\ \boldsymbol{P}_2\boldsymbol{a}_2 &= \boldsymbol{a}_2 - \frac{2}{\boldsymbol{w}_2^{\mathsf{T}}\boldsymbol{w}_2}\boldsymbol{w}_2(\boldsymbol{w}_2^{\mathsf{T}}\boldsymbol{a}_2) = (2, -2, 0), & \boldsymbol{a}_2 &= (-2, 0, 2), \\ \boldsymbol{P}_2\boldsymbol{a}_3 &= \boldsymbol{a}_3 - \frac{2}{\boldsymbol{w}_2^{\mathsf{T}}\boldsymbol{w}_2}\boldsymbol{w}_2(\boldsymbol{w}_2^{\mathsf{T}}\boldsymbol{a}_3) = (0, 0, 3), & \boldsymbol{a}_3 &= (2, -1, 2), \end{aligned}$$

sledi, da je reševanje prvotnega predoločenega sistema ekvivalentno reševanju sistema

$$\begin{bmatrix} -2 & 2 & -2 \\ 0 & -6 & 2 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} -3 \\ 0 \\ 0 \\ 3 \end{bmatrix}.$$

3. korak: Zrcaljenje v tretjem koraku ni potrebno, saj je matrika predoločenega sistema iz prejšnjega koraka že zgornje trapezna.

Kompozicija zrcaljenj v obratnem vrstnem redu, kot smo jih uporabili na matriki $\boldsymbol{A}$, določa matriko $\widetilde{\boldsymbol{Q}} = [\boldsymbol{Q}\ \boldsymbol{q}]$ razširjenega QR razcepa matrike $\boldsymbol{A}$. Zgornje trapezno matriko, dobljeno v drugem koraku, označimo z $\widetilde{\boldsymbol{R}}$, njen zgornji trikotnik pa z $\boldsymbol{R}$. Ker je

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = \left\|\widetilde{\boldsymbol{Q}}^{\mathsf{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})\right\|_2 = \left\|\widetilde{\boldsymbol{R}}\boldsymbol{x} - \widetilde{\boldsymbol{Q}}^{\mathsf{T}}\boldsymbol{b}\right\|_2 = \left\|\begin{bmatrix} \boldsymbol{R}\boldsymbol{x} - \boldsymbol{Q}^{\mathsf{T}}\boldsymbol{b} \\ -\boldsymbol{q}^{\mathsf{T}}\boldsymbol{b} \end{bmatrix}\right\|_2,$$

je

$$\min_{\boldsymbol{x}\in\mathbb{R}^3} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = \left|-\boldsymbol{q}^{\mathsf{T}}\boldsymbol{b}\right| = 3,$$

ta vrednost pa je dosežena pri vektorju $\boldsymbol{x}$, ki je rešitev sistema

$$\begin{bmatrix} -2 & 2 & -2 \\ 0 & -6 & 2 \\ 0 & 0 & -2 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}.$$

Izračunamo, da je $\boldsymbol{x} = (\frac{3}{2}, 0, 0)$.

6. Matrix Eigenvalues

The eigenvalue $\lambda \in \mathbb{C}$ of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is a number satisfying $\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$ for some non-zero vector $\boldsymbol{x} \in \mathbb{C}^n$. The latter is called the eigenvector, and the tuple $(\lambda, \boldsymbol{x})$ is the eigenpair of the matrix $\boldsymbol{A}$.

6.1. Schur Form

The eigenvalues of a matrix can be determined by the Jordan decomposition. In numerical linear algebra a computationally more stable alternative of this decomposition is used, the existence of which is guaranteed by the Schur's theorem: for any $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ there exists a unitary matrix $\boldsymbol{U} \in \mathbb{C}^{n\times n}$ ($\boldsymbol{U}^{\mathsf{H}}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^{\mathsf{H}} = \boldsymbol{I}$) and an upper triangular matrix $\boldsymbol{T} \in \mathbb{C}^{n\times n}$ such that $\boldsymbol{U}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{U} = \boldsymbol{T}$. This implies $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{T}\boldsymbol{U}^{\mathsf{H}}$, which is the Schur decomposition of $\boldsymbol{A}$, and $\boldsymbol{T}$ is called the Schur form of $\boldsymbol{A}$. Since $\boldsymbol{A}$ and $\boldsymbol{T}$ are similar matrices, the eigenvalues of $\boldsymbol{A}$ can be obtained from the diagonal of $\boldsymbol{T}$.

Exercise 6.1. Let λ be a simple eigenvalue of $\boldsymbol{A}$. Find out how the Schur form $\boldsymbol{T}$ can be used to compute the eigenvector corresponding to λ. Demonstrate the procedure by computing the eigenvector of the smalles eigenvalue of the matrix $\boldsymbol{A}$ determined by the Schur decomposition matrices

$$\boldsymbol{U} = \frac{1}{3}\begin{bmatrix} 2 & -2 & 0 & 1 \\ 1 & 2 & 0 & 2 \\ 0 & 0 & 3 & 0 \\ 2 & 1 & 0 & -2 \end{bmatrix}, \qquad \boldsymbol{T} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 2 & 1 & -1 \\ 0 & 0 & -1 & 2 \\ 0 & 0 & 0 & 3 \end{bmatrix}.$$

Solution. Matrika $\boldsymbol{T}$ je zgornja trikotna, zato elementi na diagonali ustrezajo njenim lastnim vrednostim. Ker je matrika $\boldsymbol{A}$ podobna matriki $\boldsymbol{T}$, so to tudi lastne vrednosti $\boldsymbol{A}$. Lastni vektor $\boldsymbol{x}$ za matriko $\boldsymbol{A}$, ki pripada lastni vrednosti λ, lahko izrazimo v odvisnosti od lastnega vektorja $\boldsymbol{y}$ za matriko $\boldsymbol{T}$, ki pripada λ, kot $\boldsymbol{x} = \boldsymbol{U}\boldsymbol{y}$. Premislimo torej, kako poiskati lastni vektor $\boldsymbol{y}$.

Denimo, da λ ustreza i-temu elementu $t_{i,i}$ na diagonali matrike $\boldsymbol{T}$. Pišimo

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_{1,1} & \boldsymbol{T}_{1,2} & \boldsymbol{T}_{1,3} \\ & t_{i,i} & \boldsymbol{T}_{2,3} \\ & & \boldsymbol{T}_{3,3} \end{bmatrix}, \qquad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ y_2 \\ \boldsymbol{y}_3 \end{bmatrix}.$$

Ker je $\boldsymbol{T}\boldsymbol{y} = \lambda\boldsymbol{y}$, mora veljati

$$(\boldsymbol{T}_{1,1} - \lambda\boldsymbol{I})\boldsymbol{y}_1 + \boldsymbol{T}_{1,2}y_2 + \boldsymbol{T}_{1,3}\boldsymbol{y}_3 = \boldsymbol{0}, \qquad \boldsymbol{T}_{2,3}\boldsymbol{y}_3 = \boldsymbol{0}, \qquad (\boldsymbol{T}_{3,3} - \lambda\boldsymbol{I})\boldsymbol{y}_3 = \boldsymbol{0}.$$

Ker je λ enostavna lastna vrednost, je $\boldsymbol{T}_{3,3} - \lambda\boldsymbol{I}$ nesingularna matrika, zato je $\boldsymbol{y}_3 = \boldsymbol{0}$. Ostane nam enačba

$$(\boldsymbol{T}_{1,1} - \lambda\boldsymbol{I})\boldsymbol{y}_1 + \boldsymbol{T}_{1,2}y_2 = \boldsymbol{0}.$$

Ker je lastni vektor $\boldsymbol{y}$ določen le do konstante natačno, lahko vzamemo, da je $y_2 = 1$. Z reševanjem sistema $(\boldsymbol{T}_{1,1} - \lambda\boldsymbol{I})\boldsymbol{y}_1 = -\boldsymbol{T}_{1,2}$ določimo še $\boldsymbol{y}_1$. Vektor $(\boldsymbol{y}_1, 1, \boldsymbol{0})$ je lastni vektor za $\boldsymbol{T}$, ki pripada lastni vrednosti λ.

Za podano Schurovo formo matrike $\boldsymbol{A}$ lahko lastni vektor, ki pripada najmanjši lastni vrednosti -1, dobimo z reševanjem sistema

$$\left(\begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix} - (-1) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \boldsymbol{y}_1 = - \begin{bmatrix} 3 \\ 1 \end{bmatrix}.$$

Izračunamo $\boldsymbol{y}_1 = (-7, -2)/6$, kar pomeni, da je $\boldsymbol{y} = (-7, -2, 6, 0)/6$ lastni vektor za matriko $\boldsymbol{T}$, $\boldsymbol{x} = \boldsymbol{U}\boldsymbol{y} = (-10, -11, 18, -16)/18$ pa lastni vektor za matriko $\boldsymbol{A}$, ki pripada lastni vrednosti -1.

Exercise 6.2. Given is a block matrix

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{A}_2 \end{bmatrix}, \qquad \boldsymbol{A}_1 \in \mathbb{R}^{m\times m},\ \boldsymbol{A}_2 \in \mathbb{R}^{n\times n},\ \boldsymbol{X} \in \mathbb{R}^{m\times n}.$$

1. Suppose the Schur decompositions $\boldsymbol{A}_1 = \boldsymbol{U}_1\boldsymbol{T}_1\boldsymbol{U}_1^{\mathsf{H}}$ and $\boldsymbol{A}_2 = \boldsymbol{U}_2\boldsymbol{T}_2\boldsymbol{U}_2^{\mathsf{H}}$ of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are given. Determine the Schur decomposition of $\boldsymbol{A}$ and prove that the eigenvalues of $\boldsymbol{A}$ correspond to the union of the eigenvalues of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$.

2. Using a Householder reflection compute an upper Hessenberg matrix, which has the form of $\boldsymbol{A}$ and is similar to the matrix

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}.$$

   Then determine the Schur decomposition of the matrix $\boldsymbol{B}$.

Solution.

1. Opazimo, da je

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{U}_1\boldsymbol{T}_1\boldsymbol{U}_1^{\mathsf{H}} & \boldsymbol{X} \\ \boldsymbol{0} & \boldsymbol{U}_2\boldsymbol{T}_2\boldsymbol{U}_2^{\mathsf{H}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U}_2 \end{bmatrix}}_{\boldsymbol{U}} \underbrace{\begin{bmatrix} \boldsymbol{T}_1 & \boldsymbol{U}_1^{\mathsf{H}}\boldsymbol{X}\boldsymbol{U}_2 \\ \boldsymbol{0} & \boldsymbol{T}_2 \end{bmatrix}}_{\boldsymbol{T}} \underbrace{\begin{bmatrix} \boldsymbol{U}_1^{\mathsf{H}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U}_2^{\mathsf{H}} \end{bmatrix}}_{\boldsymbol{U}^{\mathsf{H}}}.$$

Ker je $\boldsymbol{T}$ zgornja trikotna matrika in $\boldsymbol{U}$ unitarna matrika, je $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{T}\boldsymbol{U}^{\mathsf{H}}$ Schurov razcep matrike $\boldsymbol{A}$. Lastne vrednosti matrike $\boldsymbol{A}$ so enake lastnim vrednostim matrike $\boldsymbol{T}$ in se nahajajo na diagonali matrike $\boldsymbol{T}$. Torej so enake lastnim vrednostim matrik $\boldsymbol{T}_1$ in $\boldsymbol{T}_2$, te pa ustrezajo lastnim vrednostim matrik $\boldsymbol{A}_1$ in $\boldsymbol{A}_2$.

2. Konstruirajmo ortogonalno transformacijo

$$\boldsymbol{Q} = \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P} \end{bmatrix}, \qquad \boldsymbol{P} = \boldsymbol{I} - \frac{2}{\boldsymbol{w}^{\mathsf{T}}\boldsymbol{w}}\boldsymbol{w}\boldsymbol{w}^{\mathsf{T}},$$

kjer je $\boldsymbol{P}$ Householderjevo zrcaljenje, določeno z vektorjem $\boldsymbol{w} = (1, 0, -1)$, ki poskrbi, da je $\boldsymbol{P} \cdot (0, 0, -1) = (-1, 0, 0)$. Izračunamo, da je

$$\boldsymbol{Q}\boldsymbol{B}\boldsymbol{Q}^{\mathsf{T}} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \cdot \boldsymbol{Q}^{\mathsf{T}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}.$$

Vidimo, da ima matrika $\boldsymbol{Q}\boldsymbol{B}\boldsymbol{Q}^{\mathsf{T}}$ obliko matrike $\boldsymbol{A}$. Ker je

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}}_{\boldsymbol{U}_1} \cdot \underbrace{\begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix}}_{\boldsymbol{T}_1} \cdot \underbrace{\frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}}_{\boldsymbol{U}_1^{\mathsf{H}}},$$

je Schurov razcep $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{T}\boldsymbol{U}^{\mathsf{H}}$ matrike $\boldsymbol{B}$ podan z matrikama

$$\boldsymbol{T} = \begin{bmatrix} i & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & 0 & -i \end{bmatrix}, \qquad \boldsymbol{U} = \boldsymbol{Q}^{\mathsf{T}}\begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{U}_1 \end{bmatrix} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & i & 0 & 0 \\ 0 & 0 & i & 1 \\ 0 & 0 & 1 & i \\ i & 1 & 0 & 0 \end{bmatrix}.$$

The general Schur form can be replaced by the real Schur form, which corresponds to a real quasi upper triangular matrix. This is an upper triangular matrix, except that blocks of size 2×2 are allowed on the diagonal.

Exercise 6.3. Prove that for any matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ there exist an orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and a quasi upper triangular matrix $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ such that $\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{T}$.

Solution. Dokazujemo z indukcijo na velikost matrike $\boldsymbol{A} \in \mathbb{R}^{n\times n}$. Če je $n = 1$, trditev velja, saj lahko izberemo $\boldsymbol{Q} = 1$ in $\boldsymbol{T} = \boldsymbol{A}$. Denimo, da trditev velja za matrike velikosti manjše ali enake $n-1$, in dokažimo, da velja tudi za matrike velikosti n. Naj bo $(\lambda, \boldsymbol{x})$ lastni par matrike $\boldsymbol{A}$. Če je λ realna lastna vrednost, $\boldsymbol{x}$ razpenja realni enodimenzionalni invariantni podprostor $\mathcal{N} = \{\alpha\boldsymbol{x};\, \alpha \in \mathbb{R}\}$ za $\boldsymbol{A}$, saj je $\boldsymbol{A}\boldsymbol{y} \in \mathcal{N}$ za vsak $\boldsymbol{y} \in \mathcal{N}$. Če je po drugi strani λ kompleksna lastna vrednost, je tudi $(\overline{\lambda}, \overline{\boldsymbol{x}})$ lastni par za $\boldsymbol{A}$. Realni in imaginarni del vektorjev $\boldsymbol{x}$ in $\overline{\boldsymbol{x}}$,

$$\boldsymbol{x}_{\mathrm{R}} = \frac{1}{2}\left(\boldsymbol{x} + \overline{\boldsymbol{x}}\right), \qquad \boldsymbol{x}_{\mathrm{I}} = \frac{1}{2i}\left(\boldsymbol{x} - \overline{\boldsymbol{x}}\right),$$

sta vektorja iz prostora $\mathbb{R}^n$, ki razpenjata realni dvodimenzionalni invariantni podprostor $\mathcal{N} = \{\alpha\boldsymbol{x}_{\mathrm{R}} + \beta\boldsymbol{x}_{\mathrm{I}};\, \alpha, \beta \in \mathbb{R}\}$ za $\boldsymbol{A}$. V obeh primerih obstaja taka ortogonalna matrika $\boldsymbol{Q}_1 \in \mathbb{R}^{n\times n}$, da njen prvi oziroma njena prva dva stolpca določata bazo za $\mathcal{N}$. Iz invariantnosti podprostora $\mathcal{N}$ potem sledi

$$\boldsymbol{A}\boldsymbol{Q}_1 = \boldsymbol{Q}_1 \begin{bmatrix} \boldsymbol{T}_1 & \times \\ \boldsymbol{0} & \boldsymbol{A}_2 \end{bmatrix},$$

pri čemer je $\boldsymbol{T}_1$ skalar ali matrika velikosti 2×2, matrika $\boldsymbol{A}_2$ pa v odvisnosti od tega velikosti $n-1$ ali $n-2$. Po indukcijski predpostavki obstaja ortogonalna matrika $\boldsymbol{Q}_2$, da je $\boldsymbol{T}_2 = \boldsymbol{Q}_2^{\mathsf{T}}\boldsymbol{A}_2\boldsymbol{Q}_2$ kvazi zgornja trikotna. Iz tega sledi

$$\left(\boldsymbol{Q}_1 \begin{bmatrix} \boldsymbol{I} & \\ & \boldsymbol{Q}_2 \end{bmatrix}\right)^{\mathsf{T}} \boldsymbol{A} \left(\boldsymbol{Q}_1 \begin{bmatrix} \boldsymbol{I} & \\ & \boldsymbol{Q}_2 \end{bmatrix}\right) = \begin{bmatrix} \boldsymbol{T}_1 & \times \\ \boldsymbol{0} & \boldsymbol{T}_2 \end{bmatrix},$$

kar potrjuje obstoj razcepa $\boldsymbol{Q}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{T}$.

6.2. Power Method

Usually a certain eigenvalue of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ can be found by performing the iteration

$$\boldsymbol{x}^{(r+1)} = \boldsymbol{A}\boldsymbol{x}^{(r)}, \quad r = 0, 1, \ldots.$$

Here $\boldsymbol{x}^{(0)} \in \mathbb{C}^n$ is a chosen initial approximation for the eigenvector of $\boldsymbol{A}$.

Exercise 6.4. Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a diagonalizable matrix with the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ satisfying

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

This means that λ_1 is the dominant eigenvalue. Prove that

$$\lim_{r\to\infty} \frac{\boldsymbol{x}_i^{(r+1)}}{\boldsymbol{x}_i^{(r)}} = \lambda_1,$$

where $i \in \{1, 2, \ldots, n\}$ is the index, for which it holds that $|\boldsymbol{x}_i^{(r)}| = \|\boldsymbol{x}^{(r)}\|_\infty$.

Solution. Naj bodo $(\lambda_i, \boldsymbol{v}_i)$, $i = 1, 2, \ldots, n$, lastni pari matrike $\boldsymbol{A}$, ki so številčeni tako, da velja

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|\,.$$

Ker se da matriko $\boldsymbol{A}$ diagonalizirati, so lastni vektorji linearno neodvisni in začetni vektor $\boldsymbol{x}^{(0)}$ lahko razvijemo kot

$$\boldsymbol{x}^{(0)} = \alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2 + \ldots + \alpha_n \boldsymbol{v}_n.$$

To pomeni, da za približek $\boldsymbol{x}^{(r)}$ velja

$$\boldsymbol{x}^{(r)} = \boldsymbol{A}^r \boldsymbol{x}^{(0)} = \alpha_1 \lambda_1^r \boldsymbol{v}_1 + \alpha_2 \lambda_2^r \boldsymbol{v}_2 + \ldots + \alpha_n \lambda_n^r \boldsymbol{v}_n$$

in za vsak $i \in \{1, 2, \ldots, n\}$, pri katerem je $\boldsymbol{x}_i^{(r)} \neq 0$, lahko razmerje $\boldsymbol{x}_i^{(r+1)} / \boldsymbol{x}_i^{(r)}$ zapišemo kot

$$\frac{\boldsymbol{x}_i^{(r+1)}}{\boldsymbol{x}_i^{(r)}} = \lambda_1 \frac{\left(\alpha_1 \boldsymbol{v}_1 + \alpha_2 (\frac{\lambda_2}{\lambda_1})^{r+1} \boldsymbol{v}_2 + \ldots + \alpha_n (\frac{\lambda_n}{\lambda_1})^{r+1} \boldsymbol{v}_n\right)_i}{\left(\alpha_1 \boldsymbol{v}_1 + \alpha_2 (\frac{\lambda_2}{\lambda_1})^{r} \boldsymbol{v}_2 + \ldots + \alpha_n (\frac{\lambda_n}{\lambda_1})^{r} \boldsymbol{v}_n\right)_i}.$$

Ker je λ_1 dominantna lastna vrednost ($|\lambda_1| > |\lambda_2|$), od tod sledi, da razmerja konvergirajo k λ_1, ko gre r proti neskončno.

Exercise 6.5. Suppose that the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of the diagonalizable matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ satisfy

$$|\lambda_1| = |\lambda_2| > |\lambda_3| \geq \ldots \geq |\lambda_n|$$

and $\lambda_2 = -\lambda_1$. Compose a procedure for computing the eigenvalues λ_1 and λ_2 together with their corresponding eigenvectors.

Solution. V primeru, ko je $\lambda_2 = -\lambda_1$, lahko na podoben način kot v nalogi 6.4 izpeljemo, da za vsak indeks $i \in \{1, 2, \ldots, n\}$ velja

$$\lim_{r \to \infty} \frac{\boldsymbol{x}_i^{(r+2)}}{\boldsymbol{x}_i^{(r)}} = \lambda_1^2.$$

Če torej vemo, da sta po absolutni vrednosti največji lastni vrednosti matrike $\boldsymbol{A}$ nasprotno predznačeni, ju lahko določimo na podlagi razmerij komponent vsakih dveh drugih zaporednih približkov. Nadalje, ker je

$$\frac{\boldsymbol{x}^{(r)}}{\left\|\boldsymbol{x}^{(r)}\right\|} = \frac{\alpha_1 \boldsymbol{v}_1 + \alpha_2 (-1)^r \boldsymbol{v}_2 + \left(\frac{\lambda_3}{\lambda_1}\right)^r \boldsymbol{v}_3 + \ldots + \left(\frac{\lambda_n}{\lambda_1}\right)^r \boldsymbol{v}_n}{\left\|\alpha_1 \boldsymbol{v}_1 + \alpha_2 (-1)^r \boldsymbol{v}_2 + \left(\frac{\lambda_3}{\lambda_1}\right)^r \boldsymbol{v}_3 + \ldots + \left(\frac{\lambda_n}{\lambda_1}\right)^r \boldsymbol{v}_n\right\|},$$

je $\boldsymbol{x}^{(r)}$ za dovolj velik r približno enak $\beta_1 \boldsymbol{v}_1 + \beta_2 \boldsymbol{v}_2$ za neki konstanti β_1 in β_2. Za naslednji približek potem velja $\boldsymbol{x}^{(r+1)} = \lambda_1 \beta_1 \boldsymbol{v}_1 - \lambda_1 \beta_2 \boldsymbol{v}_2$. Iz tega izrazimo

$$\boldsymbol{v}_1 = \frac{1}{2\lambda_1 \beta_1} \left(\lambda_1 \boldsymbol{x}^{(r)} + \boldsymbol{x}^{(r+1)}\right), \qquad \boldsymbol{v}_2 = \frac{1}{2\lambda_1 \beta_2} \left(\lambda_1 \boldsymbol{x}^{(r)} - \boldsymbol{x}^{(r+1)}\right).$$

Vektorja $\boldsymbol{v}_1$ in $\boldsymbol{v}_2$ lahko torej določimo tako, da na podlagi λ_1 izračunamo vektorja $\lambda_1 \boldsymbol{x}^{(r)} \pm \boldsymbol{x}^{(r+1)}$ in ju normiramo.

With the power method the dominant eigenvalue of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is computed by normalizing the approximation at every step of the iteration:

$$\widetilde{\boldsymbol{x}}^{(r+1)} = \boldsymbol{A}\boldsymbol{x}^{(r)}, \quad \boldsymbol{x}^{(r+1)} = \frac{\widetilde{\boldsymbol{x}}^{(r+1)}}{\left\|\widetilde{\boldsymbol{x}}^{(r+1)}\right\|}, \quad r = 0, 1, \ldots.$$

The initial approximation $\boldsymbol{x}^{(0)} \in \mathbb{C}^n$ is also normalized. By normalizing we avoid overflows in the computation. The best approximation λ_r of the dominant eigenvalue based on the computed approximation $\boldsymbol{x}^{(r)}$ of the eigenvector is the Rayleigh quotient

$$\rho\left(\boldsymbol{A}, \boldsymbol{x}^{(r)}\right) = \frac{{\boldsymbol{x}^{(r)}}^{\mathsf{T}} \boldsymbol{A}\boldsymbol{x}^{(r)}}{{\boldsymbol{x}^{(r)}}^{\mathsf{T}} \boldsymbol{x}^{(r)}} = {\boldsymbol{x}^{(r)}}^{\mathsf{T}} \boldsymbol{A}\boldsymbol{x}^{(r)}.$$

Exercise 6.6. In Matlab prepare a function that performs the power method for the given matrix $\boldsymbol{A}$ and the initial vector $\boldsymbol{x}^{(0)}$. Let the stopping criteria of the iteration be prescribed by the tolerance for the second norm of the error $\boldsymbol{A}\boldsymbol{x}^{(r)} - \lambda_r \boldsymbol{x}^{(r)}$ and the maximal steps. Test the function with the matrix

$$\boldsymbol{A} = \begin{bmatrix} 4 & 1 & -2 & 1 \\ 1 & 3 & -1 & 8 \\ -1 & -1 & 5 & 2 \\ 0 & 3 & 1 & -6 \end{bmatrix}$$

and use the bult-in command `eig` to confirm the quality of the approximation.

Solution. Vhodni podatki implementacije potenčne metode so matrika $\boldsymbol{A}$, začetni približek $\boldsymbol{x}^{(0)}$ ter parametra, ki določata zaustavitveni kriterij. Metoda vrne približek za dominantno lastno vrednost matrike $\boldsymbol{A}$, ki je izračunan kot Rayleighjev kvocient drugega izhodnega podatka: približka za dominantni lastni vektor. Tretji izhodni podatek je število korakov, ki jih je potrebno izvesti, da je izpolnjen zaustavitveni kriterij. Pri implementaciji smo pozorni na to, da v vsakem koraku približek z matriko množimo le enkrat, saj to predstavlja najzahtevnejšo operacijo v metodi.

```
function [e,x,k] = potencna(A,x0,tol,N)

x = x0/norm(x0);
Ax = A*x;
e = x'*Ax;
k = 0;
while norm(Ax-e*x) >= tol && k < N
```

```
    x = Ax/norm(Ax);
    Ax = A*x;
    e = x'*Ax;
    k = k+1;
end

end
```

Metodo preizkusimo s spodnjimi ukazi in ugotovimo, da se zaključi po 146 korakih pri približku -8.3602 za lastno vrednost.

```
x0 = ones(4,1); tol = 1e-12; N = 200;
[e1,~,k] = potencna(A,x0,tol,N);
```

Uporaba vgrajene metode `eig` na matriki $\boldsymbol{A}$ potrdi pravilnost izračuna, saj so približki za lastne vrednosti matrike $\boldsymbol{A}$ enaki -8.3602, 6.6817, 4.8349 in 2.8436.

Exercise 6.7. Adjust the implementation of the power method from Exercise 6.6 in a way that it can accept a function that multiplies the vector by a matrix instead of the matrix itself. Using the multiplication by the inverse $\boldsymbol{A}^{-1}$, compute the absolutely smallest eigenvalue of the given matrix $\boldsymbol{A}$. By suitable shifting of the matrix try to compute all eigenvalues of the matrix $\boldsymbol{A}$.

Solution. Popravek implementacije metode, ki omogoča množenje približka z inverzom matrike $\boldsymbol{A}$, je preprost; treba je le zamenjati ukaz `A*x` z ukazom `A(x)` ter metodi za prvi vhodni podatek namesto matrike podati funkcijo množenja z matriko. Za definicijo funkcije množenja z inverzom matrike $\boldsymbol{A}$ ni treba izračunati inverza matrike $\boldsymbol{A}$. Računanje $\boldsymbol{A}^{-1}\boldsymbol{x}$ interpretiramo kot reševanje sistema $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{x}$. Ker se tekom iteracije matrika $\boldsymbol{A}$ ne spreminja, vnaprej izračunamo LU razcep $\boldsymbol{P}\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ in na vsakem koraku izvedemo le reševanje sistemov $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$ in $\boldsymbol{U}\boldsymbol{z} = \boldsymbol{y}$ s premimi in obratnimi substitucijami. Rezultat potenčne metode je recipročna vrednost približka za absolutno najmanjšo lastno vrednost matrike $\boldsymbol{A}$, saj je λ lastna vrednost matrike $\boldsymbol{A}$ natanko tedaj, ko je λ^{-1} lastna vrednost matrike $\boldsymbol{A}^{-1}$. Iskanju absolutno najmanjše lastne vrednosti matrike $\boldsymbol{A}$ z izvajanjem potenčne metode za $\boldsymbol{A}^{-1}$ pravimo inverzna potenčna metoda. S spodnjimi ukazi dobimo lastno vrednost 2.8436 pri prejšnjih nastavitvah vhodnih podatkov v 49 korakih.

```
[L,U,p] = lu(A,'vector');
[e4,~,k] = potencna(@(x)U\(L\x(p)),x0,tol,N);
e4 = 1/e4;
```

Podoben trik kot pri računanju absolutno najmanjše lastne vrednosti uporabimo tudi za računanje srednjih dveh lastnih vrednosti matrike $\boldsymbol{A}$. Za poljubno realno število μ je η lastna vrednost matrike $\boldsymbol{A}-\mu\boldsymbol{I}$ natanko tedaj, ko je η^{-1} lastna vrednost matrike $(\boldsymbol{A}-\mu\boldsymbol{I})^{-1}$. Če torej vzamemo $\mu = 6.5$, bo lastna vrednost $\lambda \approx 6.6817$ matrike $\boldsymbol{A}$ (določili smo jo z uporabo `eig`) z absolutno najmanjšo lastno vrednostjo η matrike $\boldsymbol{A} - \mu\boldsymbol{I}$ povezana z zvezo $\lambda = \eta + \mu$ in jo lahko zato poiščemo z inverzno potenčno

metodo. Na podoben način dobimo lastno vrednost $\lambda \approx 4.8349$ matrike $\boldsymbol{A}$ z uporabo premika $\mu = 5$. Prvi rezultat dobimo v 14, drugega pa v 13 korakih. Korakov je manj kot pri prejšnjih izvedbah metode, ker smo zaradi poznavanja dejanskih lastnih vrednosti matrike $\boldsymbol{A}$ lahko izbrali primerna premika.

```
[L,U,p] = lu(A-6.5*I,'vector');
[e2,~,k] = potencna(@(x)U\(L\x(p)),x0,tol,N);
e2 = 1/e2 + 6.5;

[L,U,p] = lu(A-5*I,'vector');
[e3,~,k] = potencna(@(x)U\(L\x(p)),x0,tol,N);
e3 = 1/e3 + 5;
```

Exercise 6.8. Compose a function that makes use of the power method and the Householder reductions to compute all eigenvalues of the matrix. Assume that none of the two eigenvalues of the input matrix have the same absolute value. Test the function with the matrix $\boldsymbol{A}$ form Exercise 6.6.

Solution. Pri Householderjevi redukciji s Householderjevim zrcaljenjem matrike izločimo lastno vrednost, ki smo jo izračunali s potenčno metodo, in nadaljujemo z matriko manjše velikosti, ki ima enake lastne vrednosti (z izjemo izločene) kot prvotna matrika. V osnovi je postopek naslednji.

```
n = size(A,1);
A = @(x)A*x;
E = zeros(n,1);
for k = 1:n-1
    % potencna metoda
    [E(k),v] = potencna(A,rand(n-k+1,1),tol,N);

    % Householderjeva redukcija
    A = @(x)redukcija(A,v,x);
end
E(n) = A(1);
```

Postavimo $\boldsymbol{A}_1 = \boldsymbol{A} \in \mathbb{R}^{n\times n}$. V koraku $k \in \{1, 2, \ldots, n-1\}$ s potenčno metodo izračunamo dominantno lastno vrednost λ_k matrike $\boldsymbol{A}_k \in \mathbb{R}^{(n-k+1)\times(n-k+1)}$. Nato funkcijo množenja z matriko popravimo z redukcijo. V manjkajoči metodi `redukcija` se na podlagi dominantnega lastnega vektorja $\boldsymbol{v}_k$ matrike $\boldsymbol{A}_k$ določi vektor $\boldsymbol{w}_k$ Householderjevega zrcaljenja $\boldsymbol{P}_k = \boldsymbol{I} - 2/(\boldsymbol{w}_k^\mathsf{T}\boldsymbol{w}_k)\boldsymbol{w}_k\boldsymbol{w}_k^\mathsf{T}$ z lastnostjo $\boldsymbol{P}_k\boldsymbol{v}_k = \pm\boldsymbol{e}_1$ (kjer je $\boldsymbol{e}_1$ standardni enotski vektor z enko v prvi komponenti), ki zagotavlja, da je

$$\boldsymbol{P}_k\boldsymbol{A}_k\boldsymbol{P}_k = \begin{bmatrix} \lambda_k & \times \\ \boldsymbol{0} & \boldsymbol{A}_{k+1} \end{bmatrix}.$$

Matrika $\boldsymbol{A}_{k+1} \in \mathbb{R}^{(n-k)\times(n-k)}$ ima enake lastne vrednosti kot $\boldsymbol{A}_k$ z izjemo λ_k. Množenje matrike $\boldsymbol{A}_{k+1}$ z vektorjem $\boldsymbol{x} \in \mathbb{R}^{n-k}$ se izvede brez dejanskega izračuna

matrik $\boldsymbol{P}_k$ in $\boldsymbol{A}_{k+1}$ na podlagi opazke

$$\boldsymbol{P}_k\boldsymbol{A}_k\boldsymbol{P}_k \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \lambda_k & \times \\ \boldsymbol{0} & \boldsymbol{A}_{k+1} \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \times \\ \boldsymbol{A}_{k+1}\boldsymbol{x} \end{bmatrix}.$$

Absolutno najmanjša lastna vrednost matrike $\boldsymbol{A}$ je kar $\boldsymbol{A}_n$. Z izvedbo opisanega postopka na matriki iz naloge 6.6 dobimo dobre približke za vse lastne vrednosti.

Exercise 6.9. Given is a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ with eigenpairs $(\lambda_i, \boldsymbol{v}_i)$, $i = 1, 2, \ldots, n$, where $\boldsymbol{v}_i \in \mathbb{R}^n$ are normalized vectors and the eigenvalues $\lambda_i \in \mathbb{R}$ satisfy

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|\,.$$

The eigenpair $(\lambda_1, \boldsymbol{v}_1)$ can be computed by power method. Consider the eigenpairs of the matrix $\boldsymbol{A} - \lambda_1\boldsymbol{v}_1\boldsymbol{v}_1^{\mathsf{T}}$ and based on the observations compose an algorithm for the computation of all eigenpairs of the matrix $\boldsymbol{A}$.

Solution. Za produkt matrike $\boldsymbol{A}_1 = \boldsymbol{A} - \lambda_1\boldsymbol{v}_1\boldsymbol{v}_1^{\mathsf{T}}$ in vektorja $\boldsymbol{v}_i$, $i \in \{1, 2, \ldots, n\}$, velja

$$\boldsymbol{A}_1\boldsymbol{v}_i = \boldsymbol{A}\boldsymbol{v}_i - \lambda_1(\boldsymbol{v}_1^{\mathsf{T}}\boldsymbol{v}_i)\boldsymbol{v}_1 = \lambda_1(1 - \boldsymbol{v}_1^{\mathsf{T}}\boldsymbol{v}_i)\boldsymbol{v}_1.$$

Obravnavajmo najprej $i = 1$. Ker je $\boldsymbol{v}_1$ normiran vektor, je $1 - \boldsymbol{v}_1^{\mathsf{T}}\boldsymbol{v}_1 = 0$. Torej je $\boldsymbol{A}_1\boldsymbol{v}_1 = \boldsymbol{0} = 0 \cdot \boldsymbol{v}_1$, kar pomeni, da je $(0, \boldsymbol{v}_1)$ lastni par matrike $\boldsymbol{A}_1$. Za $i > 1$ iz simetrije matrike $\boldsymbol{A}$ sledi

$$\boldsymbol{v}_1^{\mathsf{T}}\boldsymbol{v}_i = \frac{1}{\lambda_1}\boldsymbol{v}_1^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{v}_i = \frac{1}{\lambda_1}\boldsymbol{v}_1^{\mathsf{T}}\boldsymbol{A}\boldsymbol{v}_i = \frac{\lambda_i}{\lambda_1}\boldsymbol{v}_1^{\mathsf{T}}\boldsymbol{v}_i.$$

Lastni vrednosti λ_1 in λ_i sta različni, zato je $\boldsymbol{v}_1^{\mathsf{T}}\boldsymbol{v}_i = 0$. Potemtakem je $\boldsymbol{A}_1\boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i$ in $(\lambda_i, \boldsymbol{v}_i)$ je lastni par matrike $\boldsymbol{A}_1$.

Po tem premisleku je $(\lambda_2, \boldsymbol{v}_2)$ dominantni lastni par matrike $\boldsymbol{A}_1$. Izračunamo ga lahko s potenčno metodo. Nato definiramo matriko $\boldsymbol{A}_2 = \boldsymbol{A}_1 - \lambda_2\boldsymbol{v}_2\boldsymbol{v}_2^{\mathsf{T}}$. Zanjo po zgornjih sklepih velja $\boldsymbol{A}_2\boldsymbol{v}_1 = \boldsymbol{A}_2\boldsymbol{v}_2 = \boldsymbol{0}$ ter $\boldsymbol{A}_2\boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i$, $i = 3, 4, \ldots, n$. Lastni par $(\lambda_3, \boldsymbol{v}_3)$ izračunamo s potenčno metodo in enak postopek nadaljujemo rekurzivno z matrikami $\boldsymbol{A}_i = \boldsymbol{A}_{i-1} - \lambda_i\boldsymbol{v}_i\boldsymbol{v}_i^{\mathsf{T}}$. Na ta način dobimo vse lastne pare matrike $\boldsymbol{A}$. Ta postopek se imenuje Hotellingova redukcija.

Exercise 6.10. Let λ be a simple eigenvalue of a matrix $\boldsymbol{A}$ with a (right) eigenvector $\boldsymbol{v}$ (that is $\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}$) and a left eigenvector $\boldsymbol{u}$ (that is $\boldsymbol{u}^{\mathsf{H}}\boldsymbol{A} = \lambda\boldsymbol{u}^{\mathsf{H}}$). Prove that the eigenvalues of the matrix

$$\boldsymbol{B} = \boldsymbol{A} - \frac{\lambda}{\boldsymbol{u}^{\mathsf{H}}\boldsymbol{v}}\boldsymbol{v}\boldsymbol{u}^{\mathsf{H}}$$

agree with the eigenvalues of $\boldsymbol{A}$, except that instead of λ it has an eigenvalue equal to 0. Based on this observation describe how one can use the power method to compute the second absolutely largest eigenvalue μ of $\boldsymbol{A}$ provided that $|\lambda| > |\mu|$ and that μ is absolutely larger than the rest of the eigenvalues of $\boldsymbol{A}$. Demonstrate the procedure in Matlab on the example from Exercise 6.6.

Solution. Ker je

$$\boldsymbol{B}\boldsymbol{v} = \boldsymbol{A}\boldsymbol{v} - \frac{\lambda}{\boldsymbol{u}^{\mathsf{H}}\boldsymbol{v}}\boldsymbol{v}\boldsymbol{u}^{\mathsf{H}}\boldsymbol{v} = \lambda\boldsymbol{v} - \lambda\boldsymbol{v} = \boldsymbol{0} = 0 \cdot \boldsymbol{v},$$

je 0 lastna vrednost matrike $\boldsymbol{B}$. Naj bo $(\nu, \boldsymbol{w})$ poljuben lastni par matrike $\boldsymbol{A}$, različen od $(\lambda, \boldsymbol{v})$. Tedaj velja

$$\lambda\boldsymbol{u}^{\mathsf{H}}\boldsymbol{w} = \boldsymbol{u}^{\mathsf{H}}\boldsymbol{A}\boldsymbol{w} = \nu\boldsymbol{u}^{\mathsf{H}}\boldsymbol{w}$$

in ker je $\lambda \neq \nu$, je $\boldsymbol{u}^{\mathsf{H}}\boldsymbol{w} = 0$. To pomeni, da je $\boldsymbol{B}\boldsymbol{w} = \nu\boldsymbol{w}$, zato je ν tudi lastna vrednost matrike $\boldsymbol{B}$.

Če je λ dominantna lastna vrednost matrike $\boldsymbol{A}$, lahko s potenčno metodo izračunamo približek za par $(\lambda, \boldsymbol{v})$. Poleg tega lahko s potenčno metodo za matriko $\boldsymbol{A}^{\mathsf{T}}$ izračunamo približek za $\boldsymbol{u}$, saj je $\boldsymbol{u}$ (desni) lastni vektor za $\boldsymbol{A}^{\mathsf{T}}$:

$$\boldsymbol{A}^{\mathsf{T}}\boldsymbol{u} = \left(\boldsymbol{u}^{\mathsf{H}}\boldsymbol{A}\right)^{\mathsf{H}} = (\lambda\boldsymbol{u})^{\mathsf{H}} = \overline{\lambda}\boldsymbol{u}^{\mathsf{H}}.$$

S tem je določena matrika $\boldsymbol{B}$. Če je μ po absolutni vrednosti druga največja lastna vrednost matrike $\boldsymbol{A}$, je μ dominantna lastna vrednost matrike $\boldsymbol{B}$, ki jo lahko izračunamo z novo izvedbo potenčne metode.

Na podlagi teh ugotovitev lahko v Matlabu največjo in drugo največjo lastno vrednost matrike $\boldsymbol{A}$ iz naloge 6.6 izračunamo z naslednjimi ukazi.

```
x0 = ones(4,1); N = 200; tol = 1e-12;

[lam,v] = potencna(A,x0,tol,N);    % lam = 8.3602
[~,u] = potencna(A',x0,tol,N);

s = lam/(u'*v)*v;
Bx = @(x) A*x-(u'*x)*s;
mu = potencna(Bx,x0,tol,N);        % mu = 6.6817
```

Pri izvedbi potenčne metode za matriko $\boldsymbol{B}$ podamo le funkcijo $\boldsymbol{x} \mapsto \boldsymbol{B}\boldsymbol{x}$, ki zahteva manj operacij od izračuna matrike $\boldsymbol{B}$. Vidimo, da s tem postopkom dobimo dober približek za μ. Omeniti velja še, da je ob izračunu natančnejšega približka za λ vektor $\boldsymbol{u}$ učinkoviteje poiskati z inverzno potenčno metodo za matriko $\boldsymbol{A}^{\mathsf{T}} - \lambda\boldsymbol{I}$, saj lahko na ta način približek za $\boldsymbol{u}$ dobimo z bistveno manj koraki iteracije.

Exercise 6.11. Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a matrix with the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ satisfying

$$|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|\,.$$

1. Prove that for a randomly chosen vector $\boldsymbol{x} \in \mathbb{R}^n$ the Rayleigh quotients in the power method with the initial approximation $(\boldsymbol{A} - \lambda_1\boldsymbol{I})\boldsymbol{x}$ in theory converge to the eigenvalue λ_2. What troubles are expected in practice?

2. Suppose that by using the power method with the initial approximation $\boldsymbol{x}$ we compute an approximation for the eigenvalue λ_1 and then with the initial approximation $(\boldsymbol{A} - \lambda_1\boldsymbol{I})\boldsymbol{x}$ the eigenvalue λ_2. How would you continue the procedure in the manner of the first item to compute the remaining eigenvalues of $\boldsymbol{A}$?

Solution.

1. Po predpostavki so lastne vrednosti paroma različne, zato so pripadajoči lastni vektorji $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ linearno neodvisni. Naj bodo $\alpha_1, \alpha_2, \ldots, \alpha_n$ take realne vrednosti, da je
$$\boldsymbol{x} = \alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2 + \ldots + \alpha_n \boldsymbol{v}_n.$$
Za začetni približek $\boldsymbol{x}_0 = (\boldsymbol{A} - \lambda_1 \boldsymbol{I})\boldsymbol{x}$ potem velja
$$(\boldsymbol{A} - \lambda_1 \boldsymbol{I})\boldsymbol{x} = (\lambda_2 - \lambda_1)\alpha_2 \boldsymbol{v}_2 + \ldots + (\lambda_n - \lambda_1)\alpha_n \boldsymbol{v}_n.$$
Pišimo $\beta_i = (\lambda_i - \lambda_1)\alpha_i$, $i = 2, 3, \ldots, n$. Ker je $\boldsymbol{x}$ naključno izbran, lahko predpostavimo, da je $\alpha_2 \neq 0$ oziroma $\beta_2 \neq 0$. Za približek $\boldsymbol{x}_k$ na k-tem koraku potenčne metode potem velja
$$\boldsymbol{x}_k = \frac{\boldsymbol{A}^k \boldsymbol{x}_0}{\|\boldsymbol{A}^k \boldsymbol{x}_0\|} = \frac{\beta_2 \boldsymbol{v}_2 + \beta_2 (\frac{\lambda_3}{\lambda_2})^k \boldsymbol{v}_3 + \ldots + \beta_n (\frac{\lambda_n}{\lambda_2})^k \boldsymbol{v}_n}{\left\| \beta_2 \boldsymbol{v}_2 + \beta_2 (\frac{\lambda_3}{\lambda_2})^k \boldsymbol{v}_3 + \ldots + \beta_n (\frac{\lambda_n}{\lambda_2})^k \boldsymbol{v}_n \right\|}$$
in po smeri konvergira k lastnemu vektorju $\boldsymbol{v}_2$, zato Rayleighjevi kvocienti konvergirajo k λ_2. V praksi zaradi numeričnih napak koeficient razvoja vektorja $(\boldsymbol{A} - \lambda_1 \boldsymbol{I})\boldsymbol{x}$ pri vektorju $\boldsymbol{v}_1$ ni enak 0, zato lahko Rayleighjevi kvocienti v potenčni metodi sčasoma skonvergirajo k lastni vrednosti λ_1.

2. Če za začetni približek pri potenčni metodi vzamemo $(\boldsymbol{A} - \lambda_2 \boldsymbol{I})(\boldsymbol{A} - \lambda_1 \boldsymbol{I})\boldsymbol{x}$, po podobnem razmisleku kot v prvi točki Rayleighjevi kvocienti konvergirajo k λ_3. Če torej po izračunu lastne vrednosti λ_k začetni približek, uporabljen pri izvedbi potenčne metode, pomnožimo z $(\boldsymbol{A} - \lambda_k \boldsymbol{I})$, z naslednjo izvedbo potenčne metode dobimo λ_{k+1}. Na ta način nadaljujemo, dokler ne dobimo lastne vrednosti λ_n.

6.3. QR Iteration

With the QR iteration the eigenvalues of a matrix $\boldsymbol{A}$ are computed by iteratively transforming the matrix into a real Schur form, from which the eigenvalues of $\boldsymbol{A}$ can be determined. The method is usually performed by first reducing the matrix into upper Hessenberg matrix $\boldsymbol{H}$ by applying a similiarity transform. Then, we set $\boldsymbol{A}_0 = \boldsymbol{H}$ and iteratively compute
$$\boldsymbol{A}_{k+1} = \boldsymbol{R}_k \boldsymbol{Q}_k + \sigma_k \boldsymbol{I}, \qquad k = 0, 1, \ldots,$$
where $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are matrices of the QR decomposition $\boldsymbol{A}_k - \sigma_k \boldsymbol{I} = \boldsymbol{Q}_k \boldsymbol{R}_k$ for a chosen shift σ_k. The reduction to Hessenberg matrix is used to decrease the computational complexity of a step of iteration, while the shifts σ_k are introduced to speed up the convergence of the matrices $\boldsymbol{A}_k$ to the Schur form of $\boldsymbol{A}$.

Exercise 6.12. Let

$$\boldsymbol{A} = \begin{bmatrix} 6 & 1 & 1 & 1 \\ 1 & 6 & 1 & 1 \\ 1 & 1 & 6 & 1 \\ 1 & 1 & 1 & 6 \end{bmatrix}.$$

1. Use Householder reflections to reduce the matrix $\boldsymbol{A}$ to an upper Hessenberg matrix $\boldsymbol{H}$.
2. Perform a step of the QR iteration for $\boldsymbol{H}$ with the shift $\sigma_0 = 5$. Use a Givens rotation to compute the QR decomposition.

Solution.

1. Poskrbimo, da se zadnja dva elementa v prvem stolpcu matrike $\boldsymbol{A}$ spremenita v 0. To lahko dosežemo s Householderjevim zrcaljenjem

$$\boldsymbol{P} = \boldsymbol{I} - \frac{2}{\boldsymbol{w}^\mathsf{T}\boldsymbol{w}}\boldsymbol{w}\boldsymbol{w}^\mathsf{T}, \qquad \boldsymbol{w} = (1+\sqrt{3}, 1, 1).$$

   Z nekaj računanja v enem koraku redukcije dobimo Hessenbergovo matriko

$$\boldsymbol{H} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{P} \end{bmatrix} \cdot \boldsymbol{A} \cdot \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{P} \end{bmatrix}^\mathsf{T} = \begin{bmatrix} 6 & -\sqrt{3} & 0 & 0 \\ -\sqrt{3} & 8 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}.$$

   Ker je matrika $\boldsymbol{A}$ simetrična, je matrika $\boldsymbol{H}$ po pričakovanjih tridiagonalna.

2. Hessenbergova matrika $\boldsymbol{H}$ iz prejšnje točke je razcepna. Sklepamo lahko, da je 5 vsaj dvojna lastna vrednost matrike $\boldsymbol{H}$, preostali pa sta določeni s podmatriko

$$\boldsymbol{A}_0 = \begin{bmatrix} 6 & -\sqrt{3} \\ -\sqrt{3} & 8 \end{bmatrix}.$$

   S pomočjo Givensove rotacije določimo QR razcep matrike $\boldsymbol{A}_0 - 5\boldsymbol{I}$. Rotacija, ki izniči drugi element v stolpcu $(1, -\sqrt{3})$, je podana z

$$\boldsymbol{Q}_0^\mathsf{T} = \frac{1}{2}\begin{bmatrix} 1 & -\sqrt{3} \\ \sqrt{3} & 1 \end{bmatrix}.$$

   Velja

$$\boldsymbol{R}_0 = \boldsymbol{Q}_0^\mathsf{T}(\boldsymbol{A}_0 - 5\boldsymbol{I}) = \begin{bmatrix} 2 & -2\sqrt{3} \\ 0 & 0 \end{bmatrix}$$

   in kot rezultat prvega koraka QR iteracije dobimo

$$\boldsymbol{A}_1 = \boldsymbol{R}_0\boldsymbol{Q}_0 + 5\boldsymbol{I} = \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 0 & 5 \end{bmatrix}.$$

   Izkazalo se je, da premik $\sigma_0 = 5$ ustreza lastni vrednosti matrike $\boldsymbol{A}_0$, zato smo že po prvem koraku QR iteracije dobili obe lastni vrednosti matrike $\boldsymbol{A}_0$: 5 in 9. Zaključimo, da je 5 trojna, 9 pa enojna lastna vrednost matrike $\boldsymbol{A}$.

Exercise 6.13. Let $\boldsymbol{A}$ be a symmetric positive definite matrix.

1. Argue that there exists a matrix $\sqrt{\boldsymbol{A}}$ satisfying $\sqrt{\boldsymbol{A}}^2 = \boldsymbol{A}$.
2. Let $\boldsymbol{B}_0 = \sqrt{\boldsymbol{A}}$, and let $\boldsymbol{B}_k$, $k = 1, 2, \ldots$, be the matrices obtained by performing the QR iteration for the matrix $\sqrt{\boldsymbol{A}}$. Prove that $\boldsymbol{B}_k^2$ corresponds to the matrix $\boldsymbol{A}_k$ obtained by the iteration

$$\boldsymbol{A}_0 = \boldsymbol{A}, \qquad \boldsymbol{A}_k = \boldsymbol{V}_{k-1}^{\mathsf{T}} \boldsymbol{V}_{k-1}, \quad k = 1, 2, \ldots,$$

where $\boldsymbol{A}_{k-1} = \boldsymbol{V}_{k-1} \boldsymbol{V}_{k-1}^{\mathsf{T}}$ is the Cholesky decomposition of $\boldsymbol{A}_{k-1}$. From this deduce that the eigenvalues of $\boldsymbol{A}$ can be computed with the above iteration, where QR decomposition is replaced by the Cholesky decomposition.

Solution.

1. Ker je matrika $\boldsymbol{A}$ simetrična, ima realne lastne vrednosti in ortogonalne lastne vektorje, zato jo lahko predstavimo z razcepom $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\mathsf{T}}$, kjer je $\boldsymbol{U}$ ortogonalna matrika s stolpci, ki ustrezajo normiranim lastnim vektorjem matrike $\boldsymbol{A}$, $\boldsymbol{D}$ pa diagonalna matrika, ki je določena z lastnimi vrednostmi matrike $\boldsymbol{A}$. Ker je $\boldsymbol{A}$ pozitivno definitna, so njene lastne vrednosti pozitivne in matriko $\boldsymbol{D}$ lahko razcepimo v $\boldsymbol{D} = \sqrt{\boldsymbol{D}}\sqrt{\boldsymbol{D}}$, kjer je $\sqrt{\boldsymbol{D}}$ diagonalna matrika, v kateri nastopajo koreni lastnih vrednosti matrike $\boldsymbol{A}$. Sedaj lahko iz

$$\left(\boldsymbol{U}\sqrt{\boldsymbol{D}}\boldsymbol{U}^{\mathsf{T}}\right)\left(\boldsymbol{U}\sqrt{\boldsymbol{D}}\boldsymbol{U}^{\mathsf{T}}\right) = \boldsymbol{U}\left(\sqrt{\boldsymbol{D}}\left(\boldsymbol{U}^{\mathsf{T}}\boldsymbol{U}\right)\sqrt{\boldsymbol{D}}\right)\boldsymbol{U}^{\mathsf{T}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\mathsf{T}}$$

sklepamo, da je $\sqrt{\boldsymbol{A}} = \boldsymbol{U}\sqrt{\boldsymbol{D}}\boldsymbol{U}^{\mathsf{T}}$ matrika z želeno lastnostjo.

2. Pri QR iteraciji za matriko $\sqrt{\boldsymbol{A}}$ v koraku $k \in \{1, 2, \ldots\}$ izračunamo QR razcep $\boldsymbol{B}_{k-1} = \boldsymbol{Q}_{k-1}\boldsymbol{R}_{k-1}$ matrike $\boldsymbol{B}_{k-1}$ in nadaljujemo z matriko $\boldsymbol{B}_k = \boldsymbol{R}_{k-1}\boldsymbol{Q}_{k-1}$. Zanjo velja

$$\boldsymbol{B}_k = \boldsymbol{R}_{k-1}\boldsymbol{Q}_{k-1} = \boldsymbol{Q}_{k-1}^{\mathsf{T}}\boldsymbol{B}_{k-1}\boldsymbol{Q}_{k-1},$$

zato iz dejstva, da je $\boldsymbol{B}_0$ simetrična, po indukciji sledi, da je tudi $\boldsymbol{B}_k$ simetrična. Na podlagi tega lahko sklepamo, da je

$$\boldsymbol{B}_k^2 = \boldsymbol{B}_k\boldsymbol{B}_k^{\mathsf{T}} = \boldsymbol{R}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{Q}_{k-1}^{\mathsf{T}}\boldsymbol{R}_{k-1}^{\mathsf{T}} = \boldsymbol{R}_{k-1}\boldsymbol{R}_{k-1}^{\mathsf{T}},$$

in ker je $\boldsymbol{R}_{k-1}$ zgornje trikotna matrika s pozitivnimi diagonalnimi elementi, matrika $\boldsymbol{R}_{k-1}^{\mathsf{T}}$ ustreza (enolično določenemu) faktorju Choleskega za matriko $\boldsymbol{B}_k^2$. Tudi matrika $\boldsymbol{B}_{k-1}$ je simetrična, zato zanjo velja

$$\boldsymbol{B}_{k-1}^2 = \boldsymbol{B}_{k-1}^{\mathsf{T}}\boldsymbol{B}_{k-1} = \boldsymbol{R}_{k-1}^{\mathsf{T}}\boldsymbol{Q}_{k-1}^{\mathsf{T}}\boldsymbol{Q}_{k-1}\boldsymbol{R}_{k-1} = \boldsymbol{R}_{k-1}^{\mathsf{T}}\boldsymbol{R}_{k-1}$$

in po definiciji matrike $\boldsymbol{A}_k$ je $\boldsymbol{A}_k = \boldsymbol{B}_k^2$, kot smo želeli dokazati. Iz lastnosti QR iteracije zaključimo, da zaporedje matrik $\boldsymbol{B}_0, \boldsymbol{B}_1, \boldsymbol{B}_2, \ldots$ konvergira k Schurovi formi za $\sqrt{\boldsymbol{A}}$, torej zaporedje matrik $\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2$ konvergira k Schurovi formi za $\boldsymbol{A}$. Pri tem velja dodati, da je zaporedje matrik $\boldsymbol{A}_0, \boldsymbol{A}_1, \boldsymbol{A}_2, \ldots$ dobro definiramo, saj je $\boldsymbol{A}_k = \boldsymbol{V}_{k-1}^{-1}\boldsymbol{A}_{k-1}\boldsymbol{V}_{k-1}$ za vsak $k > 0$ in so zato po indukciji vse matrike v zaporedju simetrične pozitivno definitne.

7. Polynomial Interpolation

The principal problem of polynomial interpolation is to find for a given function f a polynomial of degree $n \in \mathbb{N}_0$ that agrees with the function f in $n+1$ distinct points. For such a polynomial we say that it interpolates the function or that it is the interpolation polynomial for f. In a broader sense, the theory of interpolation deals with the construction and analysis of polynomials that match the function in certain properties.

7.1. Lagrange Form

A convenient way to represent the polynomial of degree n that interpolates the values of a function in distinct points x_i, $i = 0, 1, \ldots, n$, is based on Lagrange basis polynomials $\ell_{n,i}$. They are defined by

$$\ell_{n,i}(x) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{x - x_k}{x_i - x_k}, \qquad i = 0, 1, \ldots, n,$$

from which it is clear that they are all polynomials of degree n.

Exercise 7.1. Argue that $\ell_{n,i}(x_i) = 1$ and $\ell_{n,i}(x_j) = 0$ for every $j \neq i$. From this deduce that

$$p = \sum_{k=0}^{n} f(x_k)\ell_{n,k}$$

is a polynomial of degree at most n satisfying $p(x_i) = f(x_i)$, $i = 0, 1, \ldots, n$.

Solution. Če izvrednotimo $\ell_{n,i}$ v točki x_i, so vsi členi produkta, ki določa $\ell_{n,i}$, enaki 1, zato je $\ell_{n,i}(x_i) = 1$. Če po drugi strani $\ell_{n,i}$ izvrednotimo v točki x_j, $j \neq i$, je eden izmed členov produkta enak 0, zato je $\ell_{n,i}(x_j) = 0$. Za vrednost polinoma p v poljubni interpolacijski točki x_i torej velja

$$p(x_i) = \sum_{k=0}^{n} f(x_k)\ell_{n,k}(x_i) = f(x_i).$$

Ker so vsi Lagrangeeevi bazni polinomi $\ell_{n,k}$ stopnje n, je tudi p, ki je določen kot njihova linearna kombinacija, največ stopnje n.

Exercise 7.2. Prove that the Lagrange basis polynomials $\ell_{n,i}$, $i = 0, 1, \ldots, n$, constitute a basis for the space of polynomials of degree at most n.

Solution. Število Lagrangeevih baznih polinomov $\ell_{n,i}$ je enako $n + 1$, kar ustreza dimenziji prostora polinomov stopnje največ n. Zato zadošča dokaz, da so funkcije $\ell_{n,i}$ linearno neodvisne. Denimo, da za neka realna števila a_k, $k = 0, 1, \ldots, n$, velja

$$\sum_{k=0}^{n} a_k \ell_{n,k} = 0.$$

Ker po nalogi 7.1 za vsak $i \in \{0, 1, \ldots, n\}$ velja $\ell_{n,i}(x_i) = 1$ in $\ell_{n,i}(x_j) = 0$, $j \neq i$, z izračunom vrednosti linearne kombinacije v točki x_i dobimo $a_i = 0$.

Exercise 7.3. Use the Lagrange basis polynomials to determine the polynomial of degree at most 3 that interpolates the function $f(x) = 40/(x + 1)$ at the points 0, $\frac{1}{3}$, $\frac{2}{3}$, 1. Express the polynomial also in the power basis.

Solution. Lagrangeeevi bazni polinomi so za ta primer podani z

$$\ell_{3,0}(x) = -\frac{1}{2}(3x - 1)(3x - 2)(x - 1), \qquad \ell_{3,1}(x) = \frac{9}{2}x(3x - 2)(x - 1),$$
$$\ell_{3,2}(x) = -\frac{9}{2}x(3x - 1)(x - 1), \qquad \ell_{3,3}(x) = \frac{1}{2}x(3x - 1)(3x - 2).$$

Prikazani so na sliki 7.1a. Izračunamo $f(0) = 40$, $f(\frac{1}{3}) = 30$, $f(\frac{2}{3}) = 24$, $f(1) = 20$ in polinom p, ki interpolira funkcijo, predstavimo v obliki

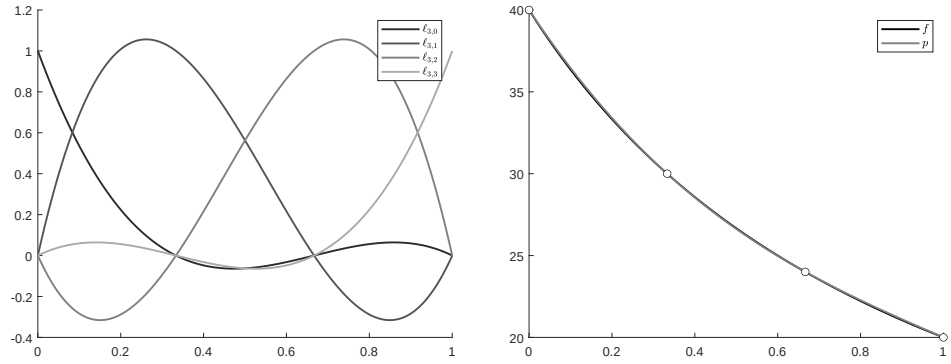
$$p = 40\ell_{3,0} + 30\ell_{3,1} + 24\ell_{3,2} + 20\ell_{3,3}.$$

Z upoštevanjem predpisov za Lagrangeeve bazne polinome lahko p v potenčni bazi izrazimo kot $p(x) = -9x^3 + 27x^2 - 38x + 40$. Polinom p je, skupaj s funkcijo f, prikazan na sliki 7.1b.

Exercise 7.4. Prove that the polynomial p of degree at most n that interpolates the values of f at points $x_0, x_1, \ldots, x_n$ can be represented in the barycentric form

$$p(x) = \frac{\displaystyle\sum_{k=0}^{n} \frac{w_k}{x - x_k} f(x_k)}{\displaystyle\sum_{k=0}^{n} \frac{w_k}{x - x_k}},$$

where $w_k = 1/\omega'(x_k)$ for $\omega(x) = (x - x_0)(x - x_1)\ldots(x - x_n)$. Argue that for precomputed values w_k, $k = 0, 1, \ldots, n$, the value of p at any point x can be computed with $\mathcal{O}(n)$ basic computational operations, independently of the values $f(x_k)$.



(A) *Lagrangeevi bazni polinomi.* (B) *Interpolacijski polinom.*

Slika 7.1: *Primer polinomske interpolacije iz naloge 7.3.*

Solution. Za odvod polinoma ω velja

$$\omega'(x) = \sum_{k=0}^{n} \prod_{\substack{i=0 \\ i \neq k}}^{n} (x - x_i),$$

torej je

$$w_k = \frac{1}{\prod\limits_{\substack{i=0 \\ i \neq k}} (x_k - x_i)}, \quad k = 0, 1, \ldots, n,$$

in po definiciji Lagrangeevih baznih polinomov lahko p zapišemo v obliki

$$p(x) = \sum_{k=0}^{n} f(x_k)\ell_{n,k}(x) = \sum_{k=0}^{n} \frac{\omega(x)}{x - x_k} w_k f(x_k).$$

Ta zveza velja za vsako funkcijo f, tudi za konstanto $x \mapsto 1$, iz česar sledi enakost

$$1 = \sum_{k=0}^{n} \frac{\omega(x)}{x - x_k} w_k.$$

Ob upoštevanju $p(x) = p(x)/1$ iz teh dveh zvez dobimo iskano obliko polinoma p.

Za izračun vrednosti w_k, $k = 0, 1, \ldots, n$, je potrebnih $(n+1)(2n-1) = \mathcal{O}(n^2)$ osnovnih računskih operacij. Ko te vrednosti, ki so odvisne le od interpolacijskih točk, imamo, lahko ob podanih vrednostih $f(x_k)$, $k = 0, 1, \ldots, n$, vrednost polinoma p v poljubni točki x izračunamo le s $5n+4 = \mathcal{O}(n)$ osnovnimi računskimi operacijami. Pri tem moramo biti pazljivi v primeru, ko je x enak eni izmed interpolacijskih točk x_k. Tedaj velja $p(x_k) = f(x_k)$.

The value of an interpolation polynomial can be computed without its explicit construction. Such procedures are called iterative interpolation. One of them is the computation of values by the Neville's scheme presented in the following.

Exercise 7.5. Let $x_0, x_1, \ldots$ be distinct points and $p_{i,k}$, $i \geq 0$, $k \geq 0$, the polynomial of degree at most k that interpolates the values of a function f at the points $x_i, x_{i+1}, \ldots, x_{i+k}$.

1. Suppose the values of the polynomials $p_{i,k-1}$ and $p_{i+1,k-1}$ at a point x are known. Prove that the value of $p_{i,k}$ at x is

$$p_{i,k}(x) = \frac{1}{x_{i+k} - x_i} \begin{vmatrix} x - x_i & p_{i,k-1}(x) \\ x - x_{i+k} & p_{i+1,k-1}(x) \end{vmatrix}.$$

2. Compute the value of the polynomial that interpolates the points $(0, 2)$, $(2, 4)$ and $(4, 8)$ at the point $x = 1$. Use the relation from the previous item starting with constant polynomials and proceeding with two steps to compute the value of the parabola that interpolates the given points.

3. Compute the value of the polynomial that interpolates the points $(0, 2)$, $(2, 4)$, $(4, 8)$ and $(6, 10)$ at the point $x = 1$. Use the calculations from the previous item.

Solution.

1. Ker je

$$\begin{aligned} p_{i,k-1}(x_j) &= f(x_j), & j &= i, i+1, \ldots, i+k-1, \\ p_{i+1,k-1}(x_j) &= f(x_j), & j &= i+1, \ldots, i+k-1, i+k, \end{aligned}$$

   za polinom

$$q(x) = \frac{x - x_i}{x_{i+k} - x_i} p_{i+1,k-1}(x) + \frac{x_{i+k} - x}{x_{i+k} - x_i} p_{i,k-1}(x)$$

   velja

$$q(x_j) = f(x_j), \qquad j = i, i+1, \ldots, i+k.$$

   Iz enoličnosti interpolacijskega polinoma sledi $p_{i,k} = q$, kar zaključuje dokaz.

2. Naj bo $x_0 = 0$, $x_1 = 2$ in $x_3 = 4$. Vemo, da je

$$p_{0,0}(x) = 2, \qquad p_{1,0}(x) = 4, \qquad p_{2,0}(x) = 8.$$

   Nato za $x = 1$ izračunamo

$$p_{0,1}(x) = \frac{1}{x_1 - x_0} \begin{vmatrix} x - x_0 & p_{0,0}(x) \\ x - x_1 & p_{1,0}(x) \end{vmatrix} = \frac{1}{2} \begin{vmatrix} 1 & 2 \\ -1 & 4 \end{vmatrix} = 3,$$

$$p_{1,1}(x) = \frac{1}{x_2 - x_1} \begin{vmatrix} x - x_1 & p_{1,0}(x) \\ x - x_2 & p_{2,0}(x) \end{vmatrix} = \frac{1}{2} \begin{vmatrix} -1 & 4 \\ -3 & 8 \end{vmatrix} = 2$$

ter še

$$p_{0,2}(x) = \frac{1}{x_2 - x_0} \begin{vmatrix} x - x_0 & p_{0,1}(x) \\ x - x_2 & p_{1,1}(x) \end{vmatrix} = \frac{1}{4} \begin{vmatrix} 1 & 3 \\ -3 & 2 \end{vmatrix} = \frac{11}{4}.$$

3. Naj bo še $x_3 = 6$. Izračune iz prejšnje točke lahko zberemo v naslednji shemi, ki jo dopolnimo z izračunom vrednosti $p_{2,1}(x)$, $p_{1,2}(x)$ in $p_{0,3}(x)$ pri $x = 1$.

$x_\cdot$	$x - x_\cdot$	$p_{\cdot,0}(x)$	$p_{\cdot,1}(x)$	$p_{\cdot,2}(x)$	$p_{\cdot,3}(x)$
0	1	2			
			3		
2	−1	4		$\frac{11}{4}$	
			2		$\frac{5}{2}$
4	−3	8		$\frac{5}{4}$	
			5		
6	−5	10			

Sledi torej, da je $p_{0,3}(1) = \frac{5}{2}$.

7.2. Newton's Form

The Lagrange form of the interpolation polynomial is simple and intelligible but computationally ineffective. More appropriate and versatile is the Newton's form, which is based on divided differences. The divided difference $[x_0, x_1, \ldots, x_n]f$ represents the leading coefficient of the polynomial of degree at most n that interpolates the function f at the points $x_0, x_1, \ldots, x_n$.

Exercise 7.6. Let $x_0, x_1, \ldots, x_n$ be pairwise distinct points. Prove that

$$[x_0, x_1, \ldots, x_n]f = \sum_{k=0}^{n} w_k f(x_k),$$

where the values w_k, $k = 0, 1, \ldots, n$, are defined as in Exercise 7.4.

Solution. Polinom p stopnje največ n, ki interpolira vrednosti funkcije f v točkah $x_0, x_1, \ldots, x_n$, lahko zapišemo v Lagrangeeevi obliki kot

$$p = \sum_{k=0}^{n} f(x_k)\ell_{n,k}.$$

Vodilni koeficient je koeficient pri funkciji $x \mapsto x^n$ in je po definiciji Lagrangeevih baznih polinomov enak

$$\sum_{k=0}^{n} f(x_k) \frac{1}{\prod_{\substack{j=0 \\ j \neq k}}^{n} (x_i - x_k)} = \sum_{k=0}^{n} \frac{f(x_k)}{\omega'(x_k)},$$

kar po definiciji w_k potrjuje dokazovano formulo.

Exercise 7.7. Let x_0 and x_1 be distinct points. Argue that for a function f defined at x_0 and x_1 it holds that $[x_0]f = f(x_0)$, $[x_1]f = f(x_1)$, and

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

To what does the value of $\lim_{x_1 \to x_0} [x_0, x_1]f$ correspond?

Solution. Uporabimo lahko rezultat naloge 7.6. Če je $\omega(x) = x - x_0$ ali $\omega(x) = x - x_1$, je odvod funkcije ω enak 1, zato je $[x_0]f = f(x_0)$ in $[x_1]f = f(x_1)$. V primeru dveh točk je funkcija ω podana s predpisom $\omega(x) = (x - x_0)(x - x_1)$, njen odvod ω' pa s predpisom $\omega'(x) = 2x - x_0 - x_1$. Velja torej

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

Limita tega izraza, ko pošljemo x_1 proti x_0, ustreza odvodu funkcije f v točki x_0.

The attractiveness of the divided differences is in the recurrence relation that they satisfy. For a given function f and the sequence of interpolation points $x_0, x_1, \ldots, x_{k-1}, x_k$ the value $[x_0, x_1, \ldots, x_k]f$ can be expressed as

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

Since the divided difference $[x]f$ is equal to the value of f at the point x, we can start with the values $f(x_0), f(x_1), \ldots, f(x_k)$ and compute $[x_0, x_1, \ldots, x_k]f$ in $k - 1$ steps of recursion. An interpolation point can appear in the sequence multiple times, by which we express that not only the value of f but also the value of one or more of its derivatives is interpolated. For the validity of the recurrence relation it is sufficient that at least two points in the sequence are distinct since the divided difference is independent of the order of the points and can thus be assumed that they are ordered increasingly. If all the points are equal, i. e. $x_k = x_0$, the divided difference can be computed by the formula

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

under the assumption that the function is k-times differentiable at x_0.

The interpolation polynomial p of f for a sequence of the interpolation points $x_0, x_1, \ldots, x_n$ is in the Newton's form given by

$$p(x) = \sum_{k=0}^{n} (x - x_0)(x - x_1) \ldots (x - x_{k-1})[x_0, x_1, \ldots, x_k]f.$$

The polynomial satisfies $p(x_i) = f(x_i)$, $i = 0, 1, \ldots, n$. Moreover, if the point x_j appears m-times, it holds that $p^{(r)}(x_j) = f^{(r)}(x_j)$, $r = 0, 1, \ldots, m - 1$.

Exercise 7.8. Let p be polynomial of degree 3 that interpolates the values of the function $f(x) = 40/(x+1)$ at the points 0, $\frac{1}{3}$, $\frac{2}{3}$, 1. Express p in the Newton's form.

Solution. Polinom p določimo na podlagi naslednje sheme. Vrednosti deljenih diferenc so izračunane s pomočjo rekurzivne formule.

$\cdot$	$[\cdot]f$	$[\cdot,\cdot]f$	$[\cdot,\cdot,\cdot]f$	$[\cdot,\cdot,\cdot,\cdot]f$
0	40			
		-30		
$\frac{1}{3}$	30		18	
		-18		-9
$\frac{2}{3}$	24		9	
		-12		
1	20			

S pomočjo podatkov v zgornji diagonali sheme določimo interpolacijski polinom

$$p(x) = 40 - 30x + 18x\left(x - \frac{1}{3}\right) - 9x\left(x - \frac{1}{3}\right)\left(x - \frac{2}{3}\right).$$

Razvidno je, da se p v potenčni bazi izraža kot $p(x) = -9x^3 + 27x^2 - 38x + 40$. Zato se (kot je zaradi enoličnosti interpolacijskega polinoma pričakovano) ujema s polinomom v Lagrangeevi obliki, izpeljanim v nalogi 7.3.

Exercise 7.9. Determine the polynomial q of degree 3 that interpolates the values and derivatives of the function $f(x) = 40/(x+1)$ at the points 0 and 1.

Solution. Polinom q določimo na podoben način kot polinom p v nalogi 7.8, le da tokrat uporabimo zaporedje interpolacijskih točk $0, 0, 1, 1$ namesto $0, \frac{1}{3}, \frac{2}{3}, 1$. S tem, da točki 0 in 1 v zaporedju podamo dvakrat, ponazarimo, da interpoliramo tako vrednosti kot vrednosti odvoda funkcije f v točkah 0 in 1. V shemi deljenih diferenc v stolpcu, ki določa deljeno diferenco f na dveh točkah, dvakrat uporabimo odvod $f'(x) = -40/(x+1)^2$ funkcije f.

$\cdot$	$[\cdot]f$	$[\cdot,\cdot]f$	$[\cdot,\cdot,\cdot]f$	$[\cdot,\cdot,\cdot,\cdot]f$
0	40			
		-40		
0	40		20	
		-20		-10
1	20		10	
		-10		
1	20			

Koeficiente polinoma q razberemo iz zgornje diagonale sheme deljenih diferenc. Dobimo

$$q(x) = 40 - 40x + 20x^2 - 10x^2(x-1),$$

kar se razlikuje od polinoma p v nalogi 7.8.

The Newton's form of the interpolation polynomial can be interpreted as a generalization of the Taylor expansion of f. Under suitable assumptions on the function f it holds that

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

where p is the interpolation polynomial of f for the sequence of interpolation points $x_0, x_1, \dots, x_n$. It follows from this relation and the Hermite–Genocchi formula that

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)(x - x_1)\dots(x - x_n)$$

for some $\xi \in (\min(x, x_0), \max(x, x_n))$, which is a convenient equality that can be used for estimation of the interpolation error on a closed interval.

Exercise 7.10. Using the fourth derivative of the function $f(x) = 40/(x + 1)$ estimate the errors

$$\max_{x\in[0,1]} |f(x) - p(x)| \quad \text{in} \quad \max_{x\in[0,1]} |f(x) - q(x)|$$

of interpolation of f by the polynomials p and q from Exercises 7.8 and 7.9 on $[0, 1]$.

Solution. Za polinom p velja

$$\max_{x\in[0,1]} |f(x) - p(x)| \le \frac{1}{4!} \max_{x\in[0,1]} \left|f^{(4)}(x)\right| \max_{x\in[0,1]} \left|x(x - \tfrac{1}{3})(x - \tfrac{2}{3})(x - 1)\right|.$$

Ker je $f^{(4)}(x) = 40 \cdot 4!/(x + 1)^5$, lahko prvi del zgornje ocene navzgor omejimo s 40. Za omejitev drugega dela ocene obravnavamo maksimalni absolutni vrednosti parabol $x(x - 1)$ in $(x - \frac{1}{3})(x - \frac{2}{3})$ na intervalu $[0, 1]$. Prva je omejena z $\frac{1}{4}$, druga pa z $\frac{2}{9}$, kar pomeni, da je drugi maksimum v oceni gotovo manjši od $\frac{1}{18}$. Na podlagi tega ocenimo

$$\max_{x\in[0,1]} |f(x) - p(x)| \le \frac{40}{18} = 2.\overline{2}.$$

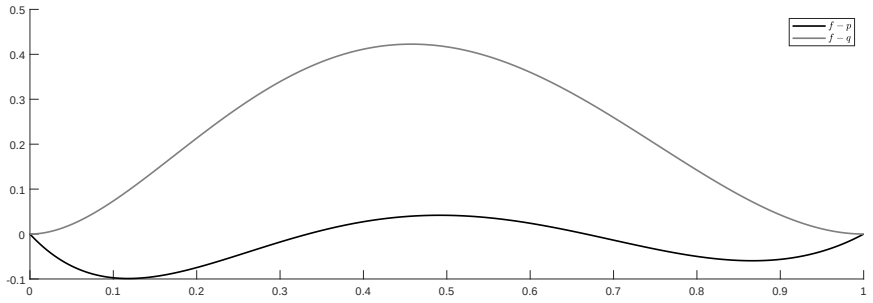
Napako interpolacije s polinom q ocenimo z

$$\max_{x\in[0,1]} |f(x) - q(x)| \le \frac{1}{4!} \max_{x\in[0,1]} \left|f^{(4)}(x)\right| \max_{x\in[0,1]} \left|x^2(x - 1)^2\right|.$$

Ker je absolutna vrednost parabole $x(x - 1)$ na intervalu $[0, 1]$ omejena z $\frac{1}{4}$, velja

$$\max_{x\in[0,1]} |f(x) - q(x)| \le \frac{40}{16} = 2.5.$$

Oceni namigujeta, da vzdolž intervala $[0, 1]$ polinom p bolje aproksimira funkcijo f kot polinom q. To potrjujeta tudi grafa razlik $f - p$ in $f - q$, prikazana na sliki 7.2.



SLIKA 7.2: *Graf razlik med funkcijo in interpolacijskima polinomoma iz naloge 7.10.*

Exercise 7.11. Prove that for the function $f(x) = 40/(x+1)$ and any sequence of interpolation points $x_0, x_1, \ldots, x_k$ it holds that

$$[x_0, x_1, \ldots, x_k]f = \frac{40(-1)^k}{(x_0+1)(x_1+1)\ldots(x_k+1)}.$$

Use this result to improve the error estimate from Exercise 7.10.

Solution. Dokažimo najprej, da izražava deljene diference velja, če so vse točke enake. Z indukcijo preverimo, da za k-ti odvod funkcije f velja

$$f^{(k)}(x) = \frac{40(-1)^k k!}{(x+1)^{k+1}},$$

kar pomeni, da je

$$[\underbrace{x_0, x_0, \ldots, x_0}_{k+1}]f = \frac{f^{(k)}(x_0)}{k!} = \frac{40(-1)^k}{(x_0+1)^{k+1}}.$$

Predpostavimo sedaj, da sta med točkami $x_0, x_1, \ldots, x_k$ vsaj dve različni (brez škode za splošnost sta to x_0 in x_k) in da izražava velja za deljenje diference, v katerih nastopa manj kot $k+1$ točk. Potem po rekurzivni zvezi za deljene diference velja

$$[x_0, x_1, \ldots, x_k]f = \frac{\frac{40(-1)^{k-1}}{(x_1+1)\ldots(x_{k-1}+1)(x_k+1)} - \frac{40(-1)^{k-1}}{(x_0+1)(x_1+1)\ldots(x_{k-1}+1)}}{x_k - x_0},$$

kar se poenostavi ravno v želeno izražavo.

Iz dokazanega sledi, da za poljuben $x \in [0,1]$ velja

$$\left|[0, \tfrac{1}{3}, \tfrac{2}{3}, 1, x]f\right| = \frac{9}{x+1} \leq 9, \qquad |[0,0,1,1,x]f| = \frac{10}{x+1} \leq 10.$$

S pomočjo ocen, izpeljanih v nalogi 7.10, sklepamo, da je

$$|f(x) - p(x)| \leq \frac{1}{2}, \qquad |f(x) - q(x)| \leq \frac{5}{8}.$$

Tudi ti dve oceni, ki sta precej boljši od ocen v nalogi 7.10, kažeta na to, da je polinom p boljša aproksimacija za f kot q.

Exercise 7.12. Adjust the Horner's method in order to evaluate a polynomial in the Newton's form. Compute the values of the polynomials p and q from Exercises 7.8 and 7.9 at the point $x = \frac{1}{2}$. Which of these values is a better estimate for $f(\frac{1}{2})$?

Solution. Vrednost polinoma, ki je predstavljen v obliki

$$x \mapsto a_0 + a_1(x - x_0) + \ldots + a_n(x - x_0) \cdot \ldots \cdot (x - x_{n-1}),$$

lahko v točki x izračunamo s $3n$ računskimi operacijami na podoben način kot s Hornerjevim postopkom. Edina razlika je, da moramo x v vsakem koraku postopka ustrezno zamakniti. Če podatke predstavimo s seznamoma $\boldsymbol{a} = (a_0, a_1, \ldots, a_n)$ in $\boldsymbol{X} = (x_0, x_1, \ldots, x_n)$, v Matlabu to opravimo na naslednji način.

```
n = length(a)-1;
b = a(n+1);
for i = n:-1:1
    b = a(i) + (x-X(i))*b;
end
```

Končna vrednost b ustreza vrednosti polinoma v točki x. Za izračun vrednosti polinomov p in q v točki $x = \frac{1}{2}$ so potrebni trije koraki postopka. Pri polinomu p spremenljivka b po vrsti zavzame vrednosti -9, 19.5, -26.75, 26.625, pri polinomu q pa -10, 25, -27.5, 26.25. Torej je $p(\frac{1}{2}) = 26.625$ in $q(\frac{1}{2}) = 26.25$. Ker je $f(\frac{1}{2}) = \frac{80}{3} = 26.\overline{6}$, polinom p v točki $\frac{1}{2}$ predstavlja boljši približek za funkcijo f kot polinom q.

Exercise 7.13. In Matlab prepare a function that accepts arrays of interpolation points and function values at the interpolation points and returns the values of the interpolation polynomial at abscissas given by a third input array. Implement the function by constructing a scheme of divided differences and evaluate the polynomial by the adjusted Horner's method. Test the function with the data provided in Exercise 7.8 that determines the interpolation polynomial p of f, and compute

$$\max_{i=0,1,\ldots,1000} \left| f\left(\tfrac{i}{1000}\right) - p\left(\tfrac{i}{1000}\right) \right|$$

to obtain a precise estimate of the actual interpolation error on the interval $[0, 1]$.

Solution. Izsek kode, podan v nalogi 7.12, za izračun vrednosti polinoma, ki interpolira vrednosti funkcije f v $n+1$ točkah, dopolnimo s konstrukcijo tabele `F` velikosti $(n+1) \times (n+1)$, v katero shranjujemo vrednosti deljenih diferenc. V prvi stolpec te tabele postavimo funkcijske vrednosti iz `fX` v interpolacijskih točkah iz `X`. Spremenljivki `X` in `fX` predstavljata vrstici dolžine $n+1$, ki ju metoda prejme kot vhodna podatka. Nato v zanki dopolnimo tabelo `F` tako, da v koraku j, $j = 2, 3, \ldots, n+1$, zapolnimo prvih $n-j+2$ elementov v j-tem stolpcu z vrednostmi, ki jih izračunamo s pomočjo rekurzivne formule za deljene diference na podlagi vrednosti v prejšnjem stolpcu tabele `F` in interpolacijskih točk iz `X`.

```
n = length(X)-1;
F = [fX' NaN(n+1,n)];
for j = 2:n+1
    for i = 1:n-j+2
        F(i,j) = (F(i+1,j-1)-F(i,j-1))/(X(i+j-1)-X(i));
    end
end
```

Prva vrstica tabele `F` predstavlja koeficiente interpolacijskega polinoma v Newtonovi obliki. Za izračun vrednosti tega polinoma lahko uporabimo prilagojeni Hornerjev postopek. Pri podatkih `X = [0 1/3 2/3 1]` in `fX = [40 30 24 20]` izračunamo vrednosti polinoma v točkah `x = 0:1e-3:1`. Maksimalna absolutna razlika med vrednostmi funkcije in polinoma p v teh točkah je 0.0990.

Exercise 7.14. Customize the function from Exercise 7.13 so that it additionally accepts the array of derivatives at the interpolation points and computes the values of the polynomial that interpolates the function and derivative values. Test the method with the data from Exercise 7.9 that determines the polynomial q for the function f. Compute

$$\max_{i=0,1,\ldots,1000} \left| f\left(\tfrac{i}{1000}\right) - q\left(\tfrac{i}{1000}\right) \right|$$

and compare the result to the error estimate from Exercise 7.13.

Solution. Metoda za izračun vrednosti polinoma, ki poleg vrednosti funkcije v interpolacijskih točkah interpolira tudi vrednosti njenih odvodov, mora sprejeti še eno dodatno vrstico `dfX` dolžine $n+1$, v kateri so shranjene vrednosti odvodov funkcije v interpolacijskih točkah. Shema deljenih diferenc je v tem primeru dolžine $(2n+2)\times(2n+2)$. V prvi stolpec `F` je treba na začetku vstaviti funkcijske vrednosti iz `fX` tako, da se vsaka zaporedoma pojavi dvakrat. V drugi stolpec v lihe vrstice `F` postavimo vrednosti odvodov iz `dfX`, v sode vrstice (razen zadnje) pa vrednosti, ki jih dobimo s pomočjo rekurzivne formule za deljene diference. Pred nadaljevanjem razširimo še seznam `X` tako, da vsako interpolacijsko točko podvojimo.

```
F = NaN(2*n+2);
F([1:2:2*n+1 2:2:2*n+2],1) = [fX fX]';
F([1:2:2*n+1 2:2:2*n],2) = [dfX diff(fX)./diff(X)]';
X([1:2:2*n+1 2:2:2*n+2]) = [X X]';
```

Preostanek postopka za izračun sheme deljenih diferenc je v večji meri enak prejšnjemu, treba je le ustrezno prilagoditi indekse v zanki. Začnemo s tretjim stolpcem in vztrajamo do stolpca $2n+2$.

Ocena za napako interpolacije vrednosti in odvodov funkcije f v točkah 0 in 1 s polinomom q, ki jo izračunamo v 1001 točkah na intervalu $[0,1]$, je enaka 0.4226 in nakazuje, da se polinom q slabše prilega funkciji f kot p. Z izrisom grafa f ter grafov p in q se lahko prepričamo, da se p enakomerno prilega funkciji f vzdolž celotnega intervala, medtem ko se q s funkcijo posebej dobro ujema v okolici točk 0 in 1, slabše pa v sredini intervala.

8. Differentiation and Integration

The basic idea in approximation of a derivative value or an integral of a function is to replace the function by its interpolation polynomial and derive or integrate the polynomial instead. With this approach, explicit rules for differentation and integration that depend on function values can be derived. One can also reverse the process: a rule is expressed as a weighted combination of function values, and the weights are determined by the requirement that the approximation is exact for polynomials of degree as high as possible. This is the so-called method of undetermined coefficients.

8.1. Differentation Rules

It makes sense to set an approximation for a derivative value of a function f at a point x_0 based on values of f in the neighborhood of x_0. In dependence of the number of used function values one can expect better or worse approximations.

Exercise 8.1. For a given function f, we would like to derive a formula for the derivative of f at the point x_0 of the form

$$F(x_0) = Af(x_0 - 2h) + Bf(x_0 - h) + Cf(x_0) + Df(x_0 + h) + Ef(x_0 + 2h),$$

where $h > 0$ is a chosen offset. Determine the constants A, B, C, D and E so that the formula is exact for all polynomials of degree at most 4. Use the system of equations obtained by replacing f by the functions $x \mapsto (x - x_0)^i$, $i = 0, 1, 2, 3, 4$.

Solution. Poljuben polinom stopnje največ 4 lahko predstavimo v predlagani potenčni bazi z zamikom x_0. Zato bo formula zanj točna, če konstante A, B, C, D in E zadoščajo enačbam

$$\begin{aligned}
0 &= A + B + C + D + E,\\
1 &= -2hA - hB + hD + 2hE,\\
0 &= 4h^2A + h^2B + h^2D + 4h^2E,\\
0 &= -8h^3A - h^3B + h^3D + 8h^3E,\\
0 &= 16h^4A + h^4B + h^4D + 16h^4E,
\end{aligned}$$

ki jih dobimo tako, da v formuli funkcijo f po vrsti nadomestimo z baznimi funkcijami $x \mapsto (x - x_0)^i$, $i = 0, 1, 2, 3, 4$. Od tod izračunamo

$$A = \frac{1}{12h}, \quad B = -\frac{2}{3h}, \quad C = 0, \quad D = \frac{2}{3h}, \quad E = -\frac{1}{12h},$$

torej je

$$F(x_0) = \frac{1}{12h}\left(f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)\right)$$

iskana formula za približek $f'(x_0)$.

Exercise 8.2. Assume f is five times continuously differentiable and for the rule F from Exercise 8.1 derive the remainder $f'(x_0) - F(x_0)$ in the form $Kf^{(5)}(\xi)$, where K is a constant depending on h abd ξ is a number from $(x_0 - 2h, x_0 + 2h)$.

Solution. Pravilo $F(x_0)$ je enolično določeno z zahtevo, da je točno za polinome stopnje manjše ali enake 4. Potemtakem zaradi enoličnosti polinoma p stopnje največ 4, ki interpolira vrednosti f v točkah $x_0 - 2h$, $x_0 - h$, x_0, $x_0 + h$, $x_0 + 2h$, ustreza vrednosti $p'(x_0)$. Za vsak $x \in [x_0 - 2h, x_0 + 2h]$ obstaja število $\xi \in (x_0 - 2h, x_0 + 2h)$, odvisno od x, da velja

$$f(x) - p(x) = \frac{f^{(5)}(\xi)}{5!}(x - x_0 + 2h)(x - x_0 + h)(x - x_0)(x - x_0 - h)(x - x_0 - 2h).$$

Če zgornji izraz odvajamo in izvrednotimo v x_0, dobimo

$$f'(x_0) - F(x_0) = f'(x_0) - p'(x_0) = \frac{f^{(5)}(\xi)}{5!} 4h^4 = \frac{h^4}{30} f^{(5)}(\xi),$$

kar predstavlja ostanek v iskani obliki.

Exercise 8.3. Using the rule F derived in Exercise 8.1, we would like to find a precise approximation for the derivative of the function $f(x) = x/(x + 1)$ at the points $x_0 = \frac{1}{2}$. The approximation error depends on h and consists of two parts, the error of method that is determined by the remainder and the round-off error that occurs in the evaluation of f. Assume the evaluation error is smaller than ε and determine the offset h, at which the approximation error is the smallest possible.

Solution. Napaka metode za dano funkcijo f ustreza absolutni vrednosti ostanka. Za majhen h bo približno enaka

$$\frac{h^4}{30} \frac{5!}{(1 + 1/2)^6} = \frac{2^8}{3^6} h^4.$$

Zaokrožitveno napako lahko na podlagi formule za odvajanje ocenimo z

$$\frac{1}{h}\left|\frac{1}{12}\varepsilon_1 - \frac{2}{3}\varepsilon_2 + \frac{2}{3h}\varepsilon_3 - \frac{1}{12}\varepsilon_4\right| \le \frac{3\varepsilon}{2h}, \qquad |\varepsilon_i| \le \varepsilon, \ i = 1, 2, 3, 4.$$

Celotna napaka je torej oblike

$$\frac{2^8}{3^6} h^4 + \frac{3\varepsilon}{2h}$$

in bo najmanjša pri $h \approx \sqrt[5]{3^7/2^{11}\varepsilon}$.

8.2. Integration Rules

Interpolation integration rules for a function f on an interval $[a, b]$ are in general represented as

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{i=0}^{n} w_i f(x_i) + R(f).$$

The points $x_0 < x_1 < \ldots < x_n$ are assumed to be distinct, and we address them as nodes of the integration rule. The coefficients w_i, $i = 0, 1, \ldots, n$, are the weights, which according to the Lagrange form of the interpolation polynomial for f can be expressed as the integrals

$$w_i = \int_a^b \ell_{n,i}(x)\,\mathrm{d}x$$

of the Lagrange basis polynomials $\ell_{n,i}$ of degree n defined based on the interpolation points x_i. The expression $R(f)$ denotes the remainder of the integration rule, which under the assumption that f is sufficiently times continously differential satisfies

$$R(f) = \int_a^b \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega(x)\,\mathrm{d}x,$$

where $\omega(x) = (x - x_0)(x - x_1)\ldots(x - x_n)$ and $\xi_x \in (a, b)$ is a number depending on x.

Exercise 8.4. Derive the interpolation integration rule for the integral of a function f on the interval $[a, b]$ with nodes $x_0 = a$ and $x_1 = b$. Assuming f is twice continuously differentiable, prove that there exits $\xi \in (a, b)$ such that the remainder satisfies

$$R(f) = -\frac{(b-a)^3}{12} f''(\xi),$$

and argue that the rule is exact for all linear functions. What does the approximation of the integral correspond to geometrically?

Solution. Uteži integracijskega pravila

$$\int_a^b f(x)\,\mathrm{d}x = w_0 f(a) + w_1 f(a) + R(f)$$

sta podani z

$$w_0 = \int_a^b \frac{x-b}{a-b}\,\mathrm{d}x = \frac{b-a}{2}, \quad w_1 = \int_a^b \frac{x-a}{b-a}\,\mathrm{d}x = \frac{b-a}{2},$$

ostanek pa z

$$R(f) = \frac{1}{2}\int_a^b f''(\xi_x)(x-a)(x-b)\,\mathrm{d}x.$$

Slednjega lahko poenostavimo. Ker je funkcija f'' zvezna na intervalu $[a,b]$, je omejena, zato obstajata realni konstanti k in K, da velja $k \le f''(\xi_x) \le K$ za vsak $x \in [a,b]$. Nadalje velja

$$K(x-a)(x-b) \le f''(\xi_x)(x-a)(x-b) \le k(x-a)(x-b),$$

saj je parabola $x \mapsto (x-a)(x-b)$ na intervalu $[a,b]$ negativnega predznaka. Po integraciji zgornjih neenačb dobimo

$$K\int_a^b (x-a)(x-b)\,\mathrm{d}x \le \int_a^b f''(\xi_x)(x-a)(x-b)\,\mathrm{d}x \le k\int_a^b (x-a)(x-b)$$

in iz

$$\int_a^b (x-a)(x-b)\,\mathrm{d}x = -\frac{1}{6}(b-a)^3$$

sledi

$$k \le -\frac{6}{(b-a)^3}\int_a^b f''(\xi_x)(x-a)(x-b)\,\mathrm{d}x \le K.$$

Ponovno upoštevamo, da je f'' zvezna na intervalu $[a,b]$, kar pomeni, da zavzame vse vrednosti med k in K. Torej obstaja tako število $\xi \in (a,b)$, da je

$$f''(\xi) = -\frac{6}{(b-a)^3}\int_a^b f''(\xi_x)(x-a)(x-b).$$

Integral funkcije f lahko torej predstavimo kot

$$\int_a^b f(x)\,\mathrm{d}x = \frac{b-a}{2}\left(f(a)+f(b)\right) - \frac{(b-a)^3}{12}f''(\xi).$$

Če je funkcija f linearna, je odvod f'' enak 0 in $R(f)=0$. To pomeni, da je pravilo točno. Geometrijsko vrednost $\frac{b-a}{2}\left(f(a)+f(b)\right)$ predstavlja ploščino trapeza z oglišči $(a,0)$, $(b,0)$, $(b,f(b))$, $(a,f(a))$, zato se je pravila prijelo ime trapezno pravilo.

A special class of interpolation rules are the Newton–Cotes rules defined by equidistant nodes x_i ($x_i = x_{i-1} + h$ for $h = (b-a)/n$). There are two types of these rules: the closed rules for which the first and the last node correspond to the endpoints of the interval ($x_0 = a$ and $x_n = b$) and the open rules for which the points x_0 and x_n are left out (the first and the last node are $x_1 = a + h$ and $x_{n-1} = b - h$). The latter type is used primarily when f is not bounded at the endpoints of the interval. The closed rule for $n = 1$ is the trapezoidal rule

$$\int_a^b f(x)\,\mathrm{d}x = \frac{h}{2}\left(f(x_0)+f(x_1)\right) - \frac{1}{12}h^3 f^{(2)}(\xi),$$

and the rule for $n = 2$ is the Simpson's rule:

$$\int_a^b f(x)\,\mathrm{d}x = \frac{1}{3}h\left(f(x_0)+4f(x_1)+f(x_2)\right) - \frac{1}{90}h^5 f^{(4)}(\xi).$$

Here $\xi \in (a,b)$. The rule for $n = 3$ is called the 3/8-rule.

Exercise 8.5. Derive the Newton–Cotes integration rule of the closed type with four nodes ($n = 3$), which takes the form

$$\int_a^b f(x)\,\mathrm{d}x = Af(x_0) + Bf(x_1) + Cf(x_2) + Df(x_3) + Ef^{(r)}(\xi), \qquad \xi \in (a,b).$$

Determine the constants A, B, C and D so that the rule is exact for polynomials of degree as high as possible. Use the functions $x \mapsto (x-a)^i$, $i = 0,1,2,3$, that form a basis of the space of polynomials of degree 3. Then, by a similar approach, determine the constants E and r assuming f is sufficiently times continuously differentiable.

Solution. Ko integracijsko pravilo uporabimo za funkcije $x \mapsto (x-a)^i$, $i = 0,1,2,3$, dobimo

$$\begin{aligned} 3h &= A + B + C + D, \\ \tfrac{1}{2}(3h)^2 &= hB + 2hC + 3hD, \\ \tfrac{1}{3}(3h)^3 &= h^2B + 4h^2C + 9h^2D, \\ \tfrac{1}{4}(3h)^4 &= h^3B + 8h^3C + 27h^3D. \end{aligned}$$

Konstante, ki zadoščajo zgornjemu sistemu enačb, so

$$A = \frac{3}{8}h, \quad B = \frac{9}{8}h, \quad C = \frac{9}{8}h, \quad D = \frac{3}{8}h.$$

Oglejmo si še ostanek integracijskega pravila za funkcijo $x \mapsto (x-x_0)^4$. Ker je

$$\frac{(3h)^5}{5} - \frac{3h}{8}\left(0 + 3h^4 + 3(2h)^4 + (3h)^4\right) = -\frac{9}{10}h^5,$$

je $r = 4$ in $E = -3/80h^5$. S tem je 3/8-pravilo

$$\int_a^b f(x)\,\mathrm{d}x = \frac{3}{8}\left(f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right) - \frac{3}{80}h^5 f^{(4)}(\xi).$$

v celoti določeno.

Among the Newton–Cotes rules of the open type the simplest is the midpoint rule with one node ($n = 2$):

$$\int_a^b f(x)\,\mathrm{d}x = 2hf(x_1) + \frac{1}{3}h^3 f^{(2)}(\xi).$$

Frequently used is the Milne's rule with three nodes corresponding to $n = 4$.

Exercise 8.6. Derive the Newton–Cotes integration rule of the open type with three nodes ($n = 4$), which takes the form

$$\int_a^b f(x)\,\mathrm{d}x = Af(x_1) + Bf(x_2) + Cf(x_3) + Df^{(r)}(\xi), \quad \xi \in (a,b).$$

Determine the constants A, B and C so that the rule is exact for polynomials of degree as high as possible. By determining the constants D and r express the remainder of the rule assuming f is sufficiently times continuously differentiable.

Solution. Z ustrezno izbiro treh prostih konstant lahko zagotovimo, da bo pravilo točno za vse parabole. To je res natanko tedaj, ko je pravilo točno za funkcije $x \mapsto (x-a)^i$, $i = 0, 1, 2$, oziroma ko konstante A, B in C zadoščajo zvezam

$$4h = A + B + C, \quad 8h^2 = hA + 2hB + 3hC, \quad 64h^3/3 = h^2A + 4h^2B + 9h^2C.$$

Rešitev dobljenega sistema enačb je $A = 8h/3$, $B = -4h/3$ in $C = 8h/3$. Nadalje opazimo, da je

$$\int_a^b (x-a)^3 \,\mathrm{d}x = 64h^4 = \frac{4h}{3}\left(2h^3 - 8h^3 + 54h^3\right)$$

in

$$\int_a^b (x-a)^4 \,\mathrm{d}x = \frac{1024h^5}{5} \neq \frac{4h}{3}\left(2h^4 - 16h^4 + 162h^4\right),$$

od kjer sledi, da je pravilo točno za polinome stopnje manjše ali enake 3, ne pa tudi za polinome stopnje 4. To pomeni, da je $r = 4$. Iz druge enačbe potem sledi še, da je $4!D = 1024h^5/5 - 592h^5/3 = 112h^5/15$, torej je $D = 14h^5/45$. Na ta način smo izpeljali Milnovo pravilo

$$\int_a^b f(x) \,\mathrm{d}x = \frac{4}{3}h\left(2f(x_1) - f(x_2) + 2f(x_3)\right) + \frac{14}{45}h^5 f^{(4)}(\xi).$$

s štirimi vozli.

Adding nodes to integration rules (i.e. increasing n) does not necessarily improve approximations of the integral. A more suitable extension of the basic Newton–Cotes rules are the so-called composite rules obtained by dividing the interval $[a, b]$ to m subintervals of equal length, on each of which we use a basic rule determined by a small n.

Exercise 8.7. Let $S_1 f$ denote the Simpson's rule on the interval $[a, b]$, and $S_2 f$ the integration rule composed of two Simpson's rules on the interval $[a, b]$. Prove that the Milne's rule derived in Exercise 8.6 can be expressed as a combination of the rules $S_1 f$ and $S_2 f$.

Solution. Integracijski pravili $S_1 f$ in $S_2 f$ sta podani z

$$S_1 f = \frac{2h}{3}\left(f(x_0) + 4f(x_2) + f(x_4)\right),$$
$$S_2 f = \frac{h}{3}\left(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + f(x_4)\right),$$

kjer je $h = (b-a)/4$ in $x_i = a + ih$, $i = 0, 1, \ldots, 4$. Ker v Milnovem pravilu ne nastopata funkcijski vrednosti v točkah x_0 in x_4, vzamemo

$$2S_2 f - S_1 f = \frac{4h}{3}\left(2f(x_1) - f(x_2) + 2f(x_3)\right),$$

kar ravno ustreza formuli, izpeljani v nalogi 8.6.

Exercise 8.8. Let

$$x_j = a + jh, \quad h = \frac{b-a}{m}, \quad j = 0, 1, \ldots, m,$$

where m is a chosen even natural number. Assume that the composite trapezoidal rule

$$T_h f = \frac{h}{2}\left(f(x_0) + 2f(x_1) + \ldots + 2f(x_{m-1}) + f(x_m)\right)$$

satisfies

$$\int_a^b f(x)\,\mathrm{d}x = T_h f + c_2 h^2 + c_4 h^4 + \ldots, \quad c_2, c_4, \ldots \in \mathbb{R}.$$

Argue why $T_h^1 f = (4T_h f - T_{2h} f)/3$ is a better integral approximation than $T_h f$ and find out to which composite rule $T_h^1 f$ corresponds.

Solution. Na podlagi izražave ostanka za sestavljeno trapezno pravilo sklepamo, da je

$$\int_a^b f(x)\,\mathrm{d}x - T_h^1 f = \frac{1}{3}\left(4c_2 h^2 + 4c_4 h^4 + \ldots - c_2(2h)^2 - c_4(2h)^4 - \ldots\right) = \mathcal{O}(h^4),$$

kar pomeni, da je $T_h^1 f$ približek za integral, ki je za dva reda boljši od $T_h f$. Z upoštevanjem izražav $T_h f$ in T_{2hf} lahko pravilo $T_h^1 f$ zapišemo kot

$$T_h^1 f = \frac{h}{3}\left(f(x_0) + 4\sum_{i=1}^{\frac{m}{2}} f(x_{2i-1}) + 2\sum_{i=1}^{\frac{m}{2}-1} f(x_{2i}) + f(x_m)\right),$$

kar ustreza ravno sestavljenemu Simpsonovemu pravilu za korak h.

Exercise 8.9. Let

$$x_j = a + jh, \quad h = \frac{b-a}{2m}, \quad j = 0, 1, \ldots, 2m,$$

where m is a chosen natural number. In Matlab implement the composite Simpson's rule

$$S_m f = \frac{h}{3}\left(f(x_0) + 4\sum_{i=1}^{m} f(x_{2i-1}) + 2\sum_{i=1}^{m-1} f(x_{2i}) + f(x_{2m})\right).$$

Test it by computing the integral of the function $f(x) = e^{-x^2}$ on the interval $[0, 3]$ for choices $m = 3k + 1$, $k = 0, 1, 2, 3, 4, 5$. Compare the results to the exact value of the integral. What is the smallest m, at which $S_m f$ absolutely differs from the exact solution $\sqrt{\pi}\,\mathrm{erf}(3)/2$ for less than 10^{-10}?

Solution. Funkcijo, ki izračuna približek po sestavljenem Simpsonovem pravilu, lahko zasnujemo tako, da sprejme seznam `fX`, ki vsebuje vrednosti funkcije f v točkah iz seznama `X = a:h:b`, kjer je `h = (b-a)/(2*m)`. Približek za integral `Sf` funkcije f na intervalu $[a, b]$ lahko potem izračunamo z naslednjim ukazom.

```
Sf = h*(fX(1) + 4*sum(fX(2:2:end-1)) + ...
                2*sum(fX(3:2:end-2)) + fX(end))/3;
```

Tabela 8.1 vsebuje napake $e_m = |\sqrt{\pi}\,\mathrm{erf}(3)/2 - S_m f|$, dobljene pri numeričnem integriranju funkcije $f(x) = e^{-x^2}$ na intervalu $[0, 3]$ pri različnih izbirah m. Točna vrednost je izračunana s pomočjo vgrajene funkcije `erf`. Nadalje izračunamo, da je $m = 50$ najmanjši parameter, pri katerem je napaka sestavljenega pravila manjša od 10^{-10}.

m	1	4	7	10	13	16
e_m	$1.8 \cdot 10^{-1}$	$1.7 \cdot 10^{-6}$	$2.3 \cdot 10^{-7}$	$5.9 \cdot 10^{-8}$	$2.1 \cdot 10^{-8}$	$9.3 \cdot 10^{-9}$

TABELA 8.1: *Napake pri integraciji s sestavljenim Simpsonovim pravilom v nalogi 8.9.*

Exercise 8.10. In Matlab implement the adaptive Simpson's rule. Test it by integrating the function $g(x) = 1/\sqrt{x + 10^{-6}}$ on the interval $[0, 1]$, which starting with 1000 at the left end of the interval quickly decreases to 0. Estimate the tolerance ensuring that the result absolutely differs from the exact solution $(\sqrt{10^6 + 1} - 1)/500$ for less that 10^{-15}, and compare the computational time to the time required by the composite Simpson's rule with $m = 10^8$ that provides a comparably good approximation.

Solution. Funkcijo za računanje integrala po adaptivnem Simpsonovem pravilu zasnujemo rekurzivno. Sprejme funkcijo f ter interval $[a, b]$, na katerem jo integriramo, poleg tega pa še toleranco (`tol`) za absolutno napako približka. V funkciji najprej izračunamo približek `Sf1` po osnovnem Simpsonovem pravilu na intervalu $[a, b]$ in približek `Sf2` po sestavljenem Simpsonovem pravilu na dveh podintervalih intervala $[a, b]$. Na podlagi teh dveh vrednosti lahko napako integracije ocenimo z `abs(Sf2-Sf1)/15`, kar sledi iz izpeljave Richardsonove ekstrapolacije. Če je ocenjena napaka manjša od tolerance, zaključimo in približek Richardsonove integracije prištejemo približku za integral, sicer pa rekurzivno nadaljujemo na levem in desnem podintervalu intervala $[a, b]$ z razpolovljeno toleranco ter rezultata rekurzivnih klicev seštejemo.

```
function Sf = adsimpson(f,a,b,tol)

c = (a+b)/2;
d = (a+c)/2;
e = (c+b)/2;

fa = f(a);
```

```
fc = f(c);
fb = f(b);

Sf1 = (b-a)*(fa + 4*fc + fb)/6;
Sf2 = (b-a)*(fa + 4*f(d) + 2*fc + 4*f(e) + fb)/12;

r = (Sf2-Sf1)/15;
if abs(r) <= tol
    Sf = Sf2 + r;
else
    Sfa = adsimpson(f,a,c,tol/2);
    Sfb = adsimpson(f,c,b,tol/2);
    Sf = Sfa + Sfb;
end

end
```

Izkaže se, da je tak postopek izredno učinkovit pri integraciji funkcij, ki na enem delu intervala strmo rastejo ali padajo, na drugem pa so zelo položne oziroma se blago spreminjajo. V našem primeru lahko z adaptivnim Simpsonovim pravilom približek za integral, ki se od točne rešitve absolutno razlikuje za manj kot 10^{-15}, izračunamo približno 20-krat hitreje kot s sestavljenim Simpsonovim pravilom.

A special type of interpolation rules are the Gaussian rules obtained by finding the nodes such that the rule is exact for the polynomials of the highest possible degree. Consequently such rules are usually applicable only when the function values are known along the entire interval of integration, but the approach is very effective due to the fact that a rule with $n+1$ nodes is exact for polynomials of degree $2n+1$.

Exercise 8.11. Let the function f be defined on the interval $[-1,1]$. Determine the weights α_0, α_1 and the nodes x_0, x_1 in the Gaussian rule

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = \alpha_0 f(x_0) + \alpha_1 f(x_1) + K f^{(4)}(\xi), \quad \xi \in (-1,1).$$

Also, determine the constant K that appears in the remainder of the rule.

Solution. V zastavljenem integracijskem pravilu proste parametre predstavljata uteži α_0 in α_1 ter vozla x_0 in x_1. Zato domnevamo, da jih lahko določimo tako, da bo pravilo točno za polinome stopnje manjše ali enake 3 (vsak tak polinom ima namreč 4 proste parametre). Uporabimo integracijsko pravilo za funkcije $x \mapsto x^i$, $i = 0, 1, 2, 3$,

in zahtevajmo, da je za te funkcije pravilo točno. Dobimo sistem nelinearnih enačb

$$\begin{aligned} 2 &= \alpha_0 + \alpha_1, \\ 0 &= \alpha_0 x_0 + \alpha_1 x_1, \\ 2/3 &= \alpha_0 x_0^2 + \alpha_1 x_1^2, \\ 0 &= \alpha_0 x_0^3 + \alpha_1 x_1^3. \end{aligned}$$

Iz teh enačb lahko hitro sklepamo, da so uteži in vozla neničelni. Če drugo enačbo množimo z x_0^2 in od nje odštejemo četrto, dobimo

$$\alpha_1 x_1 (x_0^2 - x_1^2) = 0,$$

torej mora biti $x_0^2 = x_1^2$. Po prvi in tretji enačbi je potem

$$2x_0^2 = (\alpha_0 + \alpha_1) x_0^2 = \frac{2}{3}.$$

Brez škode za splošnost privzamemo, da je $x_0 < x_1$, torej je $x_0 = -\sqrt{3}/3$ in $x_1 = \sqrt{3}/3$. To tudi pomeni, da sta uteži α_0 in α_1 enaki 1. Ostanek izpeljanega integracijskega pravila določimo s pomočjo funkcije $x \mapsto x^4$. Po predpostavki, da je za splošno funkcijo f oblike $Kf^{(4)}(\xi)$, iz

$$\int_{-1}^{1} x^4 \,\mathrm{d}x - \left(-\sqrt{3}/3\right)^4 - \left(\sqrt{3}/3\right)^4 = \frac{2}{5} - \frac{2}{9} = \frac{8}{45},$$

sledi $K = 1/135$. Integral lahko torej zapišemo kot

$$\int_{-1}^{1} f(x) \,\mathrm{d}x = f(-\sqrt{3}/3) + f(\sqrt{3}/3) + \frac{1}{135} f^{(4)}(\xi).$$

The derivation of the Gaussian rules by the method of undetermined coefficients is in general difficult since the calculation of nodes and weights requires solving a system of non-linear equations. Fortunately, it turns out that the nodes can be determined independently from the weights as zeros of a polynomial of degree $n+1$ that is orthogonal to all polynomials of degree at most n. Here, the inner product $\langle g, h \rangle$ of square-integrable functions g and h is defined by

$$\langle g, h \rangle = \int_{-1}^{1} g(x) h(x) \,\mathrm{d}x.$$

After the nodes are determined, the weights can be derived by the standard procedure with the method of undetermined coefficients.

Exercise 8.12. Derive the Gaussian integration rule

$$\int_{-1}^{1} f(x) \,\mathrm{d}x = \alpha_0 f(x_0) + \alpha_1 f(x_1) + \alpha_2 f(x_2) + K f^{(6)}(\xi), \quad \xi \in (-1, 1),$$

that is exact for polynomials of degree at most 5. Determine the nodes x_0, x_1, x_2 based on a cubic polynomial orthogonal to all polynomials of degree at most 2 and the weights α_0, α_1, α_2 by the method of undetermined coefficients. Determine also the constant K in the remainder of the rule, the expression of which is based on the assumption that f is six times continuously differentiable.

Solution. Kubični polinom, ortogonalen na polinome stopnje manjše ali enake 2, lahko določimo z Gram–Schmidtovim postopkom za ortogonalizacijo baznih funkcij $x \mapsto x^i$, $i = 0, 1, 2, 3$. Skalarni produkt funkcij g in h je podan s predpisom

$$\langle g, h \rangle = \int_{-1}^{1} g(x)h(x)\,\mathrm{d}x.$$

Po standardnem postopku dobimo bazne polinome

$$x \mapsto \frac{1}{\sqrt{2}}, \quad x \mapsto \frac{\sqrt{3}}{\sqrt{2}}x, \quad x \mapsto \frac{3\sqrt{5}}{2\sqrt{2}}\left(x^2 - \frac{1}{3}\right), \quad x \mapsto \left(x + \frac{\sqrt{3}}{\sqrt{5}}\right)x\left(x - \frac{\sqrt{3}}{\sqrt{5}}\right).$$

Pri tem zadnji polinom ni normiran, saj nas zanimajo le njegove ničle, ki ustrezajo vozlom $x_0 = -\sqrt{3}/\sqrt{5}$, $x_1 = 0$, $x_2 = \sqrt{3}/\sqrt{5}$. Sedaj lahko uteži določimo z reševanjem sistema linearnih enačb

$$\alpha_0 + \alpha_1 + \alpha_2 = 2, \quad -\frac{\sqrt{3}}{\sqrt{5}}\alpha_0 + \frac{\sqrt{3}}{\sqrt{5}}\alpha_2 = 0, \quad \frac{3}{5}\alpha_0 + \frac{3}{5}\alpha_2 = \frac{2}{3},$$

ki jih dobimo, če v nastavek za integracijsko formulo po vrsti vstavimo funkcije $x \mapsto x^i$, $i = 0, 1, 2$, za katere zahtevamo, da je pravilo točno. Iz tega sledi $\alpha_0 = 5/9$, $\alpha_1 = 8/9$ in $\alpha_2 = 5/9$. Izpeljano pravilo

$$\int_{-1}^{1} f(x)\,\mathrm{d}x = \frac{1}{9}\left(5f\left(-\frac{\sqrt{3}}{\sqrt{5}}\right) + 8f(0) + 5f\left(\frac{\sqrt{3}}{\sqrt{5}}\right)\right) + Kf^{(6)}(\xi)$$

je po konstrukciji točno za polinome stopnje manjše ali enake 5. Konstanto K v ostanku pravila določimo tako, da vstavimo funkcijo $x \mapsto x^6$. Iz

$$\frac{2}{7} = \frac{1}{9}\left(5\frac{27}{125} + 5\frac{27}{125}\right) + 6!K$$

sledi $K = 1/15750$.

9. Differential Equations

A classical problem of mathematical analysis is solving the ordinary differential equation

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

where f is a given function of two variables and y is an unkown function that we are looking for. Under suitable assumptions on the function f, the function y exists and is unique in the neighborhood of x_0 if an initial condition $y(x_0) = y_0$ is prescribed. This is called an initial or a Cauchy problem.

9.1. The Runge–Kutta Methods

An important class of numerical methods for solving initial problems are Runge–Kutta methods. These are discrete methodsused to compute an approximation for y only at some specific points on the right-hand side of x_0. The approximation y_n for the exact value $y(x_n)$ at the point $x_n = x_0 + nh$, $h > 0$, is determined by

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

for a chosen stage s and chosen coefficients γ_i, $i = 1, 2, \ldots, s$. The latter usually sum to 1, which means that the sum $\sum_{i=1}^{s} \gamma_i k_i$ is an average of the increments k_i. These are in general defined as

$$k_i = hf\left(x_{n-1} + \alpha_i h, y_{n-1} + \sum_{j=1}^{s} \beta_{i,j} k_j\right)$$

and depend on the choice of the parameters α_i and $\beta_{i,j}$. The parameters of a Runge–Kutta method can be economically expressed by the Butche tableau

$$\begin{array}{c|cccc} \alpha_1 & \beta_{1,1} & \beta_{1,2} & \cdots & \beta_{1,s} \\ \alpha_2 & \beta_{2,1} & \beta_{2,2} & \cdots & \beta_{2,s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_s & \beta_{s,1} & \beta_{s,2} & \cdots & \beta_{s,s} \\ \hline & \gamma_1 & \gamma_2 & \cdots & \gamma_s \end{array} .$$

Since any such method uses only the approximation y_{n-1} to compute the approximation y_n, these methods are considered as one-step methods.

Exercise 9.1. Argue that the explicit Euler method

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

and the implicit Euler method

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

are one-stage Runge–Kutta methods and determine their Butcher tableaus.

Solution. Iz formule za eksplicitno Eulerjevo metodo je razvidno, da ustreza enostopenjski Runge–Kutta metodi s parametri $\alpha_1 = 0$, $\beta_{1,1} = 0$ in $\gamma_1 = 1$. Implicitno Eulerjevo metodo pa dobimo, če vzamemo $\alpha_1 = 1$, $\beta_{1,1} = 1$ in $\gamma_1 = 1$, saj je

$$y_n = y_{n-1} + hf(x_n, y_n) = y_{n-1} + hf\left(x_n, y_{n-1} + hf(x_n, y_n)\right).$$

Pripadajoči Butcherjevi shemi sta

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}, \qquad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}.$$

Exercise 9.2. The modified Euler method is given by the Butcher tableau

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

By the Taylor expansion of y prove that the local error of the method is of order 3, that is $y(x_n) - y_n = \mathcal{O}(h^3)$ under the assumption that $y_{n-1} = y(x_{n-1})$.

Solution. Oglejmo si Taylorjev razvoj

$$y(x_n) = y(x_{n-1}) + hy'(x_{n-1}) + \tfrac{1}{2}h^2y''(x_{n-1}) + \mathcal{O}(h^3)$$

funkcije y okoli točke x_{n-1}. Ker je y rešitev diferencialne enačbe, velja

$$\begin{aligned} y'(x_{n-1}) &= f\left(x_{n-1}, y(x_{n-1})\right), \\ y''(x_{n-1}) &= f_x\left(x_{n-1}, y(x_{n-1})\right) + f_y\left(x_{n-1}, y(x_{n-1})\right) y'(x_{n-1}), \end{aligned}$$

zato lahko razvoj ob predpostavki $y_{n-1} = y(x_{n-1})$ zapišemo kot

$$y(x_n) = y_{n-1} + hf_{n-1} + \tfrac{1}{2}h^2\left(f_x(x_{n-1}, y_{n-1}) + f_y(x_{n-1}, y_{n-1})f_{n-1}\right) + \mathcal{O}(h^3),$$

pri čemer smo z f_{n-1} označili vrednost $f(x_{n-1}, y_{n-1})$. Primerjajmo to s približkom y_n, ki ga dobimo z izboljšano Eulerjevo metodo. Po Butcherjevi shemi je

$$y_n = y_{n-1} + hf\left(x_{n-1} + \tfrac{1}{2}h, y_{n-1} + \tfrac{1}{2}hf_{n-1}\right)$$

in iz Taylorjevega razvoja

$$f(x_{n-1}+\tfrac{1}{2}h, y_{n-1}+\tfrac{1}{2}hf_{n-1}) \approx f_{n-1}+\tfrac{1}{2}hf_x(x_{n-1},y_{n-1})+\tfrac{1}{2}hf_{n-1}f_y(x_{n-1},y_{n-1})$$

funkcije f z napako $\mathcal{O}(h^2)$ sledi, da je $y(x_n)-y_n=\mathcal{O}(h^3)$. Kot je posredno razvidno iz izpeljav, je to za red bolje kot pri eksplicitni in implicitni Eulerjevi metodi.

Exercise 9.3. In Matlab implement functions that perform the explicit, implicit and modified Euler method for solving an initial problem. Test the functions by solving the equation

$$y'(x) = 2xy(x) - \frac{y(x)}{x+1}, \quad x > 0,$$

with the initial condition $y(0) = 1$. Find approximations for the values of y at the points $x_n = \frac{n}{20}$, $n = 1, 2, \ldots, 30$, and analyze the error graphs (the exact solution to the initial problem is $y(x) = e^{x^2}/(x+1)$).

Solution. Funkcije za izvedbe metod zasnujemo tako, da sprejmejo funkcijo `f` dveh spremenljivk, ki določa diferencialno enačbo, seznam točk `x`, ki vsebuje točke, v katerih iščemo približek za rešitev, in vrednost `y0`, ki predstavlja začetno vrednost (v prvi točki iz seznama `x`). Funkcija za izvedbo eksplicitne Eulerjeve metode je preprosta in v vsakem koraku zanke izračuna približek v naslednji točki.

```
m = length(x) - 1;
h = diff(x);
y = [y0 zeros(1,m)];
for n = 2:m+1
    y(n) = y(n-1) + h(n-1)*f(x(n-1),y(n-1));
end
```

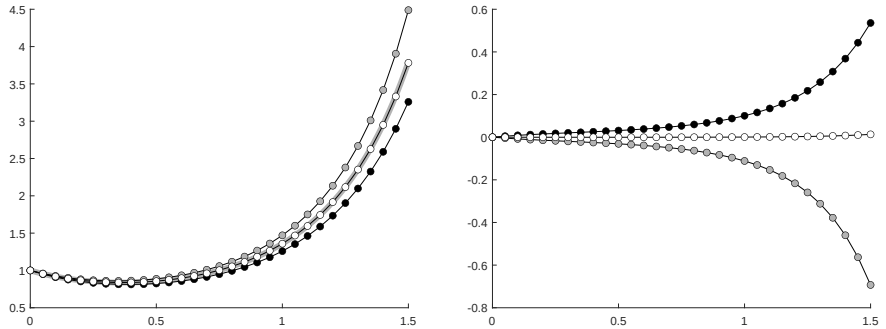
Funkcija, ki izvede implicitno Eulerjevo metodo, v vsakem koraku metode približek za rešitev najprej določi po eksplicitni Eulerjevi metodi, nato pa iterativno računa približek po implicitni Eulerjevi metodi, dokler se približka v iteraciji absolutno ne razlikujeta za manj od `tol`, ki je dodaten parameter funkcije.

```
m = length(x) - 1;
h = diff(x);
y = [y0 zeros(1,m)];
for n = 2:m+1
    y(n) = y(n-1) + h(n-1)*f(x(n-1),y(n-1));
    while true
        yns = y(n);
        y(n) = y(n-1) + h(n-1)*f(x(n),y(n));
        if abs(yns-y(n)) < tol
            break;
        end
    end
end
```

Pri implementaciji izboljšane Eulerjeve metode v vsakem koraku najprej izračunamo premika `k1` in `k2`, nato pa še približek metode.

```
m = length(x) - 1;
h = diff(x);
y = [y0 zeros(1,m)];
for n = 2:m+1
    k1 = f(x(n-1),y(n-1));
    k2 = f(x(n-1)+h(n-1)/2, y(n-1)+h(n-1)*k1/2);
    y(n) = y(n-1) + h(n-1)*k2;
end
```

Opisane funkcije preizkusimo z vhodnimi podatki `f = @(x,y) 2*x.*y - y./(x+1)`, `x = linspace(0,1.5,31)`, `y0 = 1` in `tol = 1e-10`. Rezultate podaja slika 9.1. Slika 9.1a prikazuje izračunane približke, slika 9.1b pa napake (razlike med točno vrednostjo in približki). Črne točke označuje eksplicitno Eulerjevo metodo, sive točke implicitno Eulerjevo metodo, bele točke pa izboljšano Eulerjevo metodo. Siv pas na sliki 9.1a je graf točne rešitve začetnega problema Maksimalna absolutna razlika med točnimi vrednostmi in približki v točkah iz `x` je pri eksplicitni Eulerjevi metodi 0.5358, pri implicitni Eulerjevi metodi 0.6929, pri izboljšani Eulerjevi metodi pa 0.0132. To jasno kaže, da je izboljšana Eulerjeva metoda natančnejša od drugih dveh, kot nakazuje rezultat v nalogi 9.2.



(A) (B)

Slika 9.1: *Primerjava Eulerjevih metod na primeru iz naloge 9.3.*

Exercise 9.4. The Heun's method is given by the Butcher tableau

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

Express the approximation y_n for the value of the solution of the differential equation $y'(x) = -2y(x)$ in dependence of the initial condition $y(0) = y_0$ and find out how

large can be the offset h so that the numerical solution y_n when n goes to the infinity behaves in the same way as the exact solution when x goes to the infinity.

Solution. Po Butcherjevi shemi je približek y_n pri Heunovi metodi podan z

$$y_n = y_{n-1} + \tfrac{1}{2}(k_1 + k_2), \qquad k_1 = hf(x_{n-1}, y_{n-1}), \quad k_2 = hf(x_n, y_{n-1} + hk_1).$$

Če vzamemo $f(x, y) = -2y$, je $k_1 = -2hy_{n-1}$ in $k_2 = -2y_{n-1}h(1-2h)$ ter posledično

$$y_n = y_{n-1} + \tfrac{1}{2}h\left(-2hy_{n-1} - 2y_{n-1}h(1-2h)\right) = y_{n-1}\left(1 - 2h + 2h^2\right).$$

Zato lahko y_n v odvisnosti od y_0 izrazimo kot

$$y_n = y_0\left(1 - 2h + 2h^2\right)^n.$$

Točna rešitev enačbe $y'(x) = -2y(x)$ pri začetnem pogoju $y(0) = y_0$ je $y(x) = y_0e^{-2x}$ in gre proti 0, ko pošljemo x proti neskončno. Če želimo, da se tako obnaša tudi y_n, ko pošljemo n proti neskončno, mora biti $\left|1 - 2h + 2h^2\right| < 1$. Ker je $1-2h+2h^2 > 0$ za vsak $h > 0$, je to ekvivalentno pogoju $h(h-1) < 0$ oziroma $h < 1$.

Exercise 9.5. Derive the trapezoidal method for solving an initial problem by integrating the equation $y'(x) = f(x, y(x))$ on the interval $[x_{n-1}, x_n]$, replacing the integral of $f(x, y(x))$ by the trapezoidal rule, and expressing $y(x_n)$. Determine the Butcher tableau to prove that this is a two-stage Runge–Kuta method. Similarly as in Exercise 9.4 analyze the approximation y_n for the solution of the equation $y'(x) = -2y(x)$ in dependence of the initial condition $y(0) = y_0$.

Solution. Z integracijo leve strani diferencialne enačbe dobimo

$$\int_{x_{n-1}}^{x_n} y'(x)\,\mathrm{d}x = y(x_n) - y(x_{n-1}),$$

z integracijo desne strani pa ob uporabi trapeznega pravila

$$\int_{x_{n-1}}^{x_n} f(x, y(x))\,\mathrm{d}x = \frac{1}{2}h\left(f(x_{n-1}, y(x_{n-1})) + f(x_n, y(x_n))\right) + \mathcal{O}(h^3).$$

Če nadomestimo $y(x_{n-1})$ z y_{n-1} in $y(x_n)$ z y_n ter pozabimo na člen $\mathcal{O}(h^3)$, pridemo do formule

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

ki določa trapezno metodo. Iz izpeljave je razvidno, da je podobno kot pri izboljšani Eulerjevi metodi (obravnavani v nalogi 9.2) tudi pri tej metodi lokalna napaka reda 3. Pripadajoča Butcherjeva shema je

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

Oglejmo si še, kako se y_n izraža v odvisnosti od y_0 v primeru, ko je $f(x,y) = -2y$. Iz

$$y_n = y_{n-1} + \frac{1}{2}h(-2y_{n-1} - 2y_n) = (1-h)y_{n-1} - hy_n$$

sledi

$$y_n = \frac{1-h}{1+h}y_{n-1} = \left(\frac{1-h}{1+h}\right)^n y_0,$$

kar pomeni, da je $\lim_{n\to\infty} y_n = 0$ neodvisno od $h > 0$. Zaključimo lahko, da je trapezna metoda bolj stabilna od Heunove metode (naloga 9.4), saj pri reševanju tega začetnega problema nimamo omejitve na dolžino koraka. Izkaže se, da je to tudi v splošnem prednost implicitnih metod pred eksplicitnimi.

9.2. Multistep Methods

When solving the initial problem numerically by the recurrence relation that determines an approximation y_n for the value of y at x_n based on y_{n-1}, one could also take the advantage of the previous approximations $(y_{n-2}, y_{n-3}, \ldots)$. This is the principal idea of multistep methods.

Exercise 9.6. Derive the two-step Adams–Bashforth method for solving the initial problem by integrating the equation $y'(x) = f(x, y(x))$ on the interval $[x_{n-1}, x_n]$ and replacing the function $x \mapsto f(x, y(x))$ by a linear function that inteprolates it at the points x_{n-2} and x_{n-1} (by this we achieve that the method is explicit).

Solution. Linearno funkcijo, ki interpolira funkcijo $x \mapsto f(x, y(x))$ v točkah x_{n-2} in x_{n-1}, lahko zapišemo kot

$$x \mapsto f\left(x_{n-1}, y(x_{n-1})\right) + \frac{f\left(x_{n-1}, y(x_{n-1})\right) - f\left(x_{n-2}, y(x_{n-2})\right)}{x_{n-1} - x_{n-2}}(x - x_{n-1}).$$

Če slednjo integriramo na intervalu $[x_{n-1}, x_n]$, dobimo

$$hf\left(x_{n-1}, y(x_{n-1})\right) + \frac{1}{2}h^2\frac{f\left(x_{n-1}, y(x_{n-1})\right) - f\left(x_{n-2}, y(x_{n-2})\right)}{x_{n-1} - x_{n-2}}.$$

To pomeni, da je

$$y(x_n) = y(x_{n-1}) + \frac{1}{2}h\left(3f\left(x_{n-1}, y(x_{n-1})\right) - f\left(x_{n-2}, y(x_{n-2})\right)\right) + \mathcal{O}(h^3)$$

in približek za $y(x_n)$ v tej metodi izračunamo kot

$$y_n = y_{n-1} + \frac{1}{2}h(3f(x_{n-1}, y_{n-1}) - f(x_{n-2}, y_{n-2})).$$

References

[1] B. Plestenjak, *Razširjen uvod v numerične metode*, DMFA – založništvo, Ljubljana 2015.

[2] J. Grošelj, *Pozdravljen, Matlab: osnove Matlaba za študente numerične matematike*, Elektronski vir: `https://www.fmf.uni-lj.si/~groseljj/matlab/pozdravljen_matlab.pdf`, Ljubljana 2018.