1 VAJA IZ MEHANIKE TRDNIH TELES

ponovitev linearne algebre

Naloga

Naloga

Določi rang matrike $[A] = \begin{bmatrix} 3 & -2 & -1 & -1 \\ 6 & -4 & 4 & 3 \\ 3 & -2 & 5 & 4 \\ 9 & -6 & 3 & 2 \end{bmatrix}$.

Naloga

Določi rang matrike $[A] = \begin{bmatrix} 3 & -2 & -1 & -1 \\ 6 & -4 & 4 & 3 \\ 3 & -2 & 5 & 4 \\ 9 & -6 & 3 & 2 \end{bmatrix}$.

Označimo z n_r število linearno neodvisnih vrstic matrike, z n_c pa število linearno neodvisnih stolpcev matrike. Potem velja $n_r = n_c$. Rang linearne preslikave je enak dimenziji slike preslikave. Označimo rang matrike $[A]$ z $\texttt{rank}([A])$. Velja $\texttt{rank}([A]) = n_r = n_c$. Nadalje veljajo trditve: Rang matrike se ne spremeni, če eno od vrstic pomnožim s poljubnim realnim številom. Rang matrike se ne spremeni, če eno od vrstic prištejem drugi. Isto velja za stolpce. Te lastnosti bomo s pridom uporabili pri izračunu ranga.

Rang matrike $[A]$ je enak rangu matrike $[A_1]$, ki jo dobimo tako, da prvo vrstico pomnožimo z (-2) in prištejemo drugi, nato prvo vrstico pomnožimo z (-1) in prištejemo tretji in nazadnje prvo vrstico pomnožimo z (-3) in prištejemo četrti.

$$[A_1] = \begin{bmatrix} 3 & -2 & -1 & -1 \\ 0 & 0 & 6 & 5 \\ 0 & 0 & 6 & 5 \\ 0 & 0 & 6 & 5 \end{bmatrix}.$$

Rang matrike $[A]$ je enak rangu matrike $[A_1]$, ki jo dobimo tako, da prvo vrstico pomnožimo z (-2) in prištejemo drugi, nato prvo vrstico pomnožimo z (-1) in prištejemo tretji in nazadnje prvo vrstico pomnožimo z (-3) in prištejemo četrti.

$$[A_1] = \begin{bmatrix} 3 & -2 & -1 & -1 \\ 0 & 0 & 6 & 5 \\ 0 & 0 & 6 & 5 \\ 0 & 0 & 6 & 5 \end{bmatrix}.$$

Rang matrike $[A_1]$ je enak rangu matrike $[A_2]$, ki jo dobimo tako, da drugo vrstico $[A_1]$ pomnožimo z (-1) in prištejemo tretji in nato drugo vrstico $[A_1]$ pomnožimo z (-1) in prištejemo četrti.

$$[A_2] = \begin{bmatrix} 3 & -2 & -1 & -1 \\ 0 & 0 & 6 & 5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Rang matrike $[A_3]$ je enak rangu matrike $[A_3]$, ki jo dobimo tako, da prvi stolpec $[A_2]$ delimo z 3, drugi stolpec $[A_2]$ delimo z 2, prvi stolpec $[A_2]$ prištejemo drugemu tretjemu in nato še četrtemu, dalje tretji stolpec $[A_2]$ delimo z 6, četrti s (-5), tretji stolpec prištejemo četrtemu in končno preuredimo vrstni red stolpcev.

$$[A_3] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} .$$

Rang matrike $[A_3]$ je enak rangu matrike $[A_3]$, ki jo dobimo tako, da prvi stolpec $[A_2]$ delimo z 3, drugi stolpec $[A_2]$ delimo z 2, prvi stolpec $[A_2]$ prištejemo drugemu tretjemu in nato še četrtemu, dalje tretji stolpec $[A_2]$ delimo z 6, četrti s (-5), tretji stolpec prištejemo četrtemu in končno preuredimo vrstni red stolpcev.

$$[A_3] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Z zaporedjem operacij, ki ohranjajo rang, smo matriko $[A]$ prevedli v matriko $[A_3]$ z enakim rangom. Iz matrike $[A_3]$ neposredno preberemo $\mathrm{rang}([A_3]) = \mathrm{rang}([A]) = 2$.

Rang matrike $[A_3]$ je enak rangu matrike $[A_3]$, ki jo dobimo tako, da prvi stolpec $[A_2]$ delimo z 3, drugi stolpec $[A_2]$ delimo z 2, prvi stolpec $[A_2]$ prištejemo drugemu tretjemu in nato še četrtemu, dalje tretji stolpec $[A_2]$ delimo z 6, četrti s (-5), tretji stolpec prištejemo četrtemu in končno preuredimo vrstni red stolpcev.

$$[A_3] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Z zaporedjem operacij, ki ohranjajo rang, smo matriko $[A]$ prevedli v matriko $[A_3]$ z enakim rangom. Iz matrike $[A_3]$ neposredno preberemo $\mathrm{rang}([A_3]) = \mathrm{rang}([A]) = 2$. V MATLABU vnesemo matriko $[A]$

Rang matrike $[A_3]$ je enak rangu matrike $[A_3]$, ki jo dobimo tako, da prvi stolpec $[A_2]$ delimo z 3, drugi stolpec $[A_2]$ delimo z 2, prvi stolpec $[A_2]$ prištejemo drugemu tretjemu in nato še četrtemu, dalje tretji stolpec $[A_2]$ delimo z 6, četrti s (-5), tretji stolpec prištejemo četrtemu in končno preuredimo vrstni red stolpcev.

$$[A_3] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} .$$

Z zaporedjem operacij, ki ohranjajo rang, smo matriko $[A]$ prevedli v matriko $[A_3]$ z enakim rangom. Iz matrike $[A_3]$ neposredno preberemo $\mathrm{rang}([A_3]) = \mathrm{rang}([A]) = 2$. V MATLABU vnesemo matriko $[A]$

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
```

Rang matrike $[A_3]$ je enak rangu matrike $[A_3]$, ki jo dobimo tako, da prvi stolpec $[A_2]$ delimo z 3, drugi stolpec $[A_2]$ delimo z 2, prvi stolpec $[A_2]$ prištejemo drugemu tretjemu in nato še četrtemu, dalje tretji stolpec $[A_2]$ delimo z 6, četrti s (-5), tretji stolpec prištejemo četrtemu in končno preuredimo vrstni red stolpcev.

$$[A_3] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} .$$

Z zaporedjem operacij, ki ohranjajo rang, smo matriko $[A]$ prevedli v matriko $[A_3]$ z enakim rangom. Iz matrike $[A_3]$ neposredno preberemo $\mathrm{rang}([A_3]) = \mathrm{rang}([A]) = 2$. V MATLABU vnesemo matriko $[A]$

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
```

in vtipkamo ukaz `rank(A)`.

Rang matrike $[A_3]$ je enak rangu matrike $[A_3]$, ki jo dobimo tako, da prvi stolpec $[A_2]$ delimo z 3, drugi stolpec $[A_2]$ delimo z 2, prvi stolpec $[A_2]$ prištejemo drugemu tretjemu in nato še četrtemu, dalje tretji stolpec $[A_2]$ delimo z 6, četrti s (-5), tretji stolpec prištejemo četrtemu in končno preuredimo vrstni red stolpcev.

$$[A_3] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Z zaporedjem operacij, ki ohranjajo rang, smo matriko $[A]$ prevedli v matriko $[A_3]$ z enakim rangom. Iz matrike $[A_3]$ neposredno preberemo $\mathrm{rang}([A_3]) = \mathrm{rang}([A]) = 2$. V MATLABU vnesemo matriko $[A]$

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
```

in vtipkamo ukaz `rank(A)`.

Program vrne 2.

Če s `help rank` pogledamo razlago ukaza `rank` dobimo

```
RANK   Matrix rank.
  RANK(A) provides an estimate of the number of linearly
  independent rows or columns of a matrix A.
  RANK(A,tol) is the number of singular values of A
  that are larger than tol.
  RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.
```

Če s `help rank` pogledamo razlago ukaza `rank` dobimo

```
RANK   Matrix rank.
  RANK(A) provides an estimate of the number of linearly
  independent rows or columns of a matrix A.
  RANK(A,tol) is the number of singular values of A
  that are larger than tol.
  RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.
```

To nas utegne zavesti, da uporabimo ukaz `rank(A,tol)` s posebej izbrano predpisano toleranco `tol`, manjšo od tiste, ki jo ponuja program, misleč, da bomo tako dobili bolj natančno rešitev.

Če s `help rank` pogledamo razlago ukaza `rank` dobimo

```
RANK   Matrix rank.
  RANK(A) provides an estimate of the number of linearly
  independent rows or columns of a matrix A.
  RANK(A,tol) is the number of singular values of A
  that are larger than tol.
  RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.
```

To nas utegne zavesti, da uporabimo ukaz `rank(A,tol)` s posebej izbrano predpisano toleranco `tol`, manjšo od tiste, ki jo ponuja program, misleč, da bomo tako dobili bolj natančno rešitev.

Pa poskusimo. Izberimo `tol = 1e-16`.

Če s `help rank` pogledamo razlago ukaza `rank` dobimo

```
RANK   Matrix rank.
  RANK(A) provides an estimate of the number of linearly
  independent rows or columns of a matrix A.
  RANK(A,tol) is the number of singular values of A
  that are larger than tol.
  RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.
```

To nas utegne zavesti, da uporabimo ukaz `rank(A,tol)` s posebej izbrano predpisano toleranco `tol`, manjšo od tiste, ki jo ponuja program, misleč, da bomo tako dobili bolj natančno rešitev.

Pa poskusimo. Izberimo `tol = 1e-16`. Vtipkamo ukaz `rank(A,tol)`.

Če s `help rank` pogledamo razlago ukaza `rank` dobimo

```
RANK   Matrix rank.
  RANK(A) provides an estimate of the number of linearly
  independent rows or columns of a matrix A.
  RANK(A,tol) is the number of singular values of A
  that are larger than tol.
  RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.
```

To nas utegne zavesti, da uporabimo ukaz `rank(A,tol)` s posebej izbrano predpisano toleranco `tol`, manjšo od tiste, ki jo ponuja program, misleč, da bomo tako dobili bolj natančno rešitev.

Pa poskusimo. Izberimo `tol = 1e-16`. Vtipkamo ukaz `rank(A,tol)`. Program vrne 3.

Če s `help rank` pogledamo razlago ukaza `rank` dobimo

```
RANK   Matrix rank.
  RANK(A) provides an estimate of the number of linearly
  independent rows or columns of a matrix A.
  RANK(A,tol) is the number of singular values of A
  that are larger than tol.
  RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.
```

To nas utegne zavesti, da uporabimo ukaz `rank(A,tol)` s posebej izbrano predpisano toleranco `tol`, manjšo od tiste, ki jo ponuja program, misleč, da bomo tako dobili bolj natančno rešitev.

Pa poskusimo. Izberimo `tol = 1e-16`. Vtipkamo ukaz `rank(A,tol)`. Program vrne 3.

Težava se pojavi, ker predpisana toleranca `tol` ni vsklajena z zaokrožitvenimi napakami izračuna.

Če s `help rank` pogledamo razlago ukaza `rank` dobimo

```
RANK   Matrix rank.
  RANK(A) provides an estimate of the number of linearly
  independent rows or columns of a matrix A.
  RANK(A,tol) is the number of singular values of A
  that are larger than tol.
  RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.
```

To nas utegne zavesti, da uporabimo ukaz `rank(A,tol)` s posebej izbrano predpisano toleranco `tol`, manjšo od tiste, ki jo ponuja program, misleč, da bomo tako dobili bolj natančno rešitev.

Pa poskusimo. Izberimo `tol = 1e-16`. Vtipkamo ukaz `rank(A,tol)`. Program vrne 3.

Težava se pojavi, ker predpisana toleranca `tol` ni vsklajena z zaokrožitvenimi napakami izračuna. Obravnava zaokrožitvenih napak krepko presega okvir te naloge in se zato z njo ne bomo ukvarjali.

Če s `help rank` pogledamo razlago ukaza `rank` dobimo

```
RANK   Matrix rank.
  RANK(A) provides an estimate of the number of linearly
  independent rows or columns of a matrix A.
  RANK(A,tol) is the number of singular values of A
  that are larger than tol.
  RANK(A) uses the default tol = max(size(A)) * norm(A) * eps.
```

To nas utegne zavesti, da uporabimo ukaz `rank(A,tol)` s posebej izbrano predpisano toleranco `tol`, manjšo od tiste, ki jo ponuja program, misleč, da bomo tako dobili bolj natančno rešitev.

Pa poskusimo. Izberimo `tol = 1e-16`. Vtipkamo ukaz `rank(A,tol)`. Program vrne 3.

Težava se pojavi, ker predpisana toleranca `tol` ni vsklajena z zaokrožitvenimi napakami izračuna. Obravnava zaokrožitvenih napak krepko presega okvir te naloge in se zato z njo ne bomo ukvarjali.

Lahko pa povzamemo splošen poduk: “Če ne poznaš vpliva določenih parametrov na izračun, raje privzemi vgrajene vrednosti.”

Naloga 17

Naloga 17

Dani sta matriki

$$[A] = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}, \qquad [B] = \begin{bmatrix} a & b \\ -1 & 1 \end{bmatrix}.$$

Kakšna zveza mora veljati med parametroma a in b, da bosta matriki $[A]$ in $[B]$ ekvivalentni? Določi parametra a in b tako, da bosta matriki $[A]$ in $[B]$ podobni. Izračunaj tudi matriko $[P]$, za katero je $[A] = [P]^{-1} \cdot [B] \cdot [P]$.

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q])$$

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\,\det([B])\,\det([Q]).$$

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\ \det([B])\ \det([Q]).$$

Opazimo, da je $\det([A]) = 0$.

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\ \det([B])\ \det([Q]).$$

Opazimo, da je $\det([A]) = 0$. Od tu sledi

$$0 = \det([A]) = \det([P])\ \det([B])\ \det([Q]).$$

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\,\det([B])\,\det([Q]).$$

Opazimo, da je $\det([A]) = 0$. Od tu sledi

$$0 = \det([A]) = \det([P])\,\det([B])\,\det([Q]).$$

Vendar sta po definiciji $\det([P]) \neq 0$ in $\det([Q]) \neq 0$.

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\,\det([B])\,\det([Q]).$$

Opazimo, da je $\det([A]) = 0$. Od tu sledi

$$0 = \det([A]) = \det([P])\,\det([B])\,\det([Q]).$$

Vendar sta po definiciji $\det([P]) \neq 0$ in $\det([Q]) \neq 0$. Torej mora biti $\det([B]) = 0$.

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\,\det([B])\,\det([Q]).$$

Opazimo, da je $\det([A]) = 0$. Od tu sledi

$$0 = \det([A]) = \det([P])\,\det([B])\,\det([Q]).$$

Vendar sta po definiciji $\det([P]) \neq 0$ in $\det([Q]) \neq 0$. Torej mora biti $\det([B]) = 0$. Iz zadnje zahteve sledi $a = -b$.

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\,\det([B])\,\det([Q]).$$

Opazimo, da je $\det([A]) = 0$. Od tu sledi

$$0 = \det([A]) = \det([P])\,\det([B])\,\det([Q]).$$

Vendar sta po definiciji $\det([P]) \neq 0$ in $\det([Q]) \neq 0$. Torej mora biti $\det([B]) = 0$. Iz zadnje zahteve sledi $a = -b$. Ugotovili smo: Če takšni matriki $[P]$ in $[Q]$ obstajata potem mora biti $a = -b$.

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\,\det([B])\,\det([Q]).$$

Opazimo, da je $\det([A]) = 0$. Od tu sledi

$$0 = \det([A]) = \det([P])\,\det([B])\,\det([Q]).$$

Vendar sta po definiciji $\det([P]) \neq 0$ in $\det([Q]) \neq 0$. Torej mora biti $\det([B]) = 0$. Iz zadnje zahteve sledi $a = -b$. Ugotovili smo: Če takšni matriki $[P]$ in $[Q]$ obstajata potem mora biti $a = -b$.

Sestavimo še obratno vprašanje.

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\,\det([B])\,\det([Q]).$$

Opazimo, da je $\det([A]) = 0$. Od tu sledi

$$0 = \det([A]) = \det([P])\,\det([B])\,\det([Q]).$$

Vendar sta po definiciji $\det([P]) \neq 0$ in $\det([Q]) \neq 0$. Torej mora biti $\det([B]) = 0$. Iz zadnje zahteve sledi $a = -b$. Ugotovili smo: Če takšni matriki $[P]$ in $[Q]$ obstajata potem mora biti $a = -b$.

Sestavimo še obratno vprašanje. Naj bo $a = -b$.

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\,\det([B])\,\det([Q]).$$

Opazimo, da je $\det([A]) = 0$. Od tu sledi

$$0 = \det([A]) = \det([P])\,\det([B])\,\det([Q]).$$

Vendar sta po definiciji $\det([P]) \neq 0$ in $\det([Q]) \neq 0$. Torej mora biti $\det([B]) = 0$. Iz zadnje zahteve sledi $a = -b$. Ugotovili smo: Če takšni matriki $[P]$ in $[Q]$ obstajata potem mora biti $a = -b$.

Sestavimo še obratno vprašanje. Naj bo $a = -b$. Ali potem res obstajata takšni matriki $[P]$ in $[Q]$?

Matriki $[A]$ in $[B]$ sta *ekvivalentni*, če obstajata dve nesingularni matriki $[P]$ in $[Q]$, da velja

$$[A] = [P]\,[B]\,[Q]$$

Zato velja

$$\det([A]) = \det([P]\,[B]\,[Q]) = \det([P])\,\det([B])\,\det([Q]).$$

Opazimo, da je $\det([A]) = 0$. Od tu sledi

$$0 = \det([A]) = \det([P])\,\det([B])\,\det([Q]).$$

Vendar sta po definiciji $\det([P]) \neq 0$ in $\det([Q]) \neq 0$. Torej mora biti $\det([B]) = 0$. Iz zadnje zahteve sledi $a = -b$. Ugotovili smo: Če takšni matriki $[P]$ in $[Q]$ obstajata potem mora biti $a = -b$.

Sestavimo še obratno vprašanje. Naj bo $a = -b$. Ali potem res obstajata takšni matriki $[P]$ in $[Q]$? Tudi odgovor na to vprašanje je pritrdilen, **saj imata matriki $[A]$ in $[B]$ enak rang**.

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1}\,[B]\,[P]. \tag{1}$$

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1} [B] [P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni.

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1} [B] [P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$.

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1}\,[B]\,[P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti!

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1}\,[B]\,[P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti! Matrika $[A]$ je simetrična. Ali mora biti tudi matrika $[B]$ simetrična?

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1}\,[B]\,[P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti! Matrika $[A]$ je simetrična. Ali mora biti tudi matrika $[B]$ simetrična? Izkaže se, da to ni bistveno!

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1} [B] [P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti! Matrika $[A]$ je simetrična. Ali mora biti tudi matrika $[B]$ simetrična? Izkaže se, da to ni bistveno! **Matriki $[A]$ in $[B]$ morata imeti enake lastne vrednosti!**

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1}[B][P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti! Matrika $[A]$ je simetrična. Ali mora biti tudi matrika $[B]$ simetrična? Izkaže se, da to ni bistveno! **Matriki $[A]$ in $[B]$ morata imeti enake lastne vrednosti!** Pa preverimo, če je to res potrebno! Naj ima matrika $[A]$ lastno vrednost λ in naj velja enakost (1).

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1} [B] [P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti! Matrika $[A]$ je simetrična. Ali mora biti tudi matrika $[B]$ simetrična? Izkaže se, da to ni bistveno! **Matriki $[A]$ in $[B]$ morata imeti enake lastne vrednosti!** Pa preverimo, če je to res potrebno! Naj ima matrika $[A]$ lastno vrednost λ in naj velja enakost (1). Potem velja

$$\det([A] - \lambda [I]) = 0$$

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1}\,[B]\,[P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti! Matrika $[A]$ je simetrična. Ali mora biti tudi matrika $[B]$ simetrična? Izkaže se, da to ni bistveno! **Matriki $[A]$ in $[B]$ morata imeti enake lastne vrednosti!** Pa preverimo, če je to res potrebno! Naj ima matrika $[A]$ lastno vrednost λ in naj velja enakost (1). Potem velja

$$\det([A] - \lambda\,[I]) = 0 \;=\; \det([P]^{-1}\,[B]\,[P] - \lambda[I])$$

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1}\,[B]\,[P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti! Matrika $[A]$ je simetrična. Ali mora biti tudi matrika $[B]$ simetrična? Izkaže se, da to ni bistveno! **Matriki $[A]$ in $[B]$ morata imeti enake lastne vrednosti!** Pa preverimo, če je to res potrebno! Naj ima matrika $[A]$ lastno vrednost λ in naj velja enakost (1). Potem velja

$$\begin{aligned}\det([A] - \lambda\,[I]) = 0 \;&=\; \det([P]^{-1}\,[B]\,[P] - \lambda[I]) \\ &=\; \det\left([P]^{-1}\,([B] - \lambda\,[I])\,[P]\right)\end{aligned}$$

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1}\,[B]\,[P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti! Matrika $[A]$ je simetrična. Ali mora biti tudi matrika $[B]$ simetrična? Izkaže se, da to ni bistveno! **Matriki $[A]$ in $[B]$ morata imeti enake lastne vrednosti!** Pa preverimo, če je to res potrebno! Naj ima matrika $[A]$ lastno vrednost λ in naj velja enakost (1). Potem velja

$$\begin{aligned}
\det([A] - \lambda\,[I]) = 0 \;&=\; \det([P]^{-1}\,[B]\,[P] - \lambda[I]) \\
&=\; \det\left([P]^{-1}\,([B] - \lambda\,[I])\,[P]\right) \\
&=\; \det([P]^{-1})\,\det([B] - \lambda\,[I])\,\det([P]).
\end{aligned}$$

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1}\,[B]\,[P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti! Matrika $[A]$ je simetrična. Ali mora biti tudi matrika $[B]$ simetrična? Izkaže se, da to ni bistveno! **Matriki $[A]$ in $[B]$ morata imeti enake lastne vrednosti!** Pa preverimo, če je to res potrebno! Naj ima matrika $[A]$ lastno vrednost λ in naj velja enakost (1). Potem velja

$$\begin{aligned}
\det([A] - \lambda\,[I]) = 0 \;&=\; \det([P]^{-1}\,[B]\,[P] - \lambda[I]) \\
&=\; \det\left([P]^{-1}\,([B] - \lambda\,[I])\,[P]\right) \\
&=\; \det([P]^{-1})\,\det([B] - \lambda\,[I])\,\det([P]).
\end{aligned}$$

Matrika $[P]$ je nesingularna, torej mora biti $\det([B]-\lambda\,[I]) = 0$.

Matriki $[A]$ in $[B]$ sta *podobni*, če obstaja taka nesingularna matrika $[P]$, da velja

$$[A] = [P]^{-1}\,[B]\,[P]. \tag{1}$$

Opazimo: Če sta matriki podobni, potem sta tudi ekvivalentni. Torej mora biti $\det([B]) = 0$. Kot bomo videli v nadaljevanju pa to še ni zadosti! Matrika $[A]$ je simetrična. Ali mora biti tudi matrika $[B]$ simetrična? Izkaže se, da to ni bistveno! **Matriki $[A]$ in $[B]$ morata imeti enake lastne vrednosti!** Pa preverimo, če je to res potrebno! Naj ima matrika $[A]$ lastno vrednost λ in naj velja enakost (1). Potem velja

$$\begin{aligned}
\det([A] - \lambda\,[I]) = 0 \;&=\; \det([P]^{-1}\,[B]\,[P] - \lambda[I]) \\
&=\; \det\left([P]^{-1}\,([B] - \lambda\,[I])\,[P]\right) \\
&=\; \det([P]^{-1})\,\det([B] - \lambda\,[I])\,\det([P]).
\end{aligned}$$

Matrika $[P]$ je nesingularna, torej mora biti $\det([B]-\lambda\,[I]) = 0$. Ugotovili smo, da imata matriki $[A]$ in $[B]$ enake lastne vrednosti.

Izračunajmo lastne vrednosti in lastne vektorje matrike $[A]$. Rešujemo enačbo

$$[A]\,\boldsymbol{v} = \lambda\,\boldsymbol{v}.$$

Izračunajmo lastne vrednosti in lastne vektorje matrike $[A]$. Rešujemo enačbo

$$[A]\,\boldsymbol{v} = \lambda\,\boldsymbol{v}.$$

Enačbo prepišemo v ekvivalentno obliko

$$0 = [A]\,\boldsymbol{v} - \lambda\,\boldsymbol{v}$$

Izračunajmo lastne vrednosti in lastne vektorje matrike $[A]$. Rešujemo enačbo

$$[A]\,\boldsymbol{v} = \lambda\,\boldsymbol{v}.$$

Enačbo prepišemo v ekvivalentno obliko

$$0 = [A]\,\boldsymbol{v} - \lambda\,\boldsymbol{v} = ([A] - \lambda\,[I])\,\boldsymbol{v}.$$

Izračunajmo lastne vrednosti in lastne vektorje matrike $[A]$. Rešujemo enačbo

$$[A]\,\boldsymbol{v} = \lambda\,\boldsymbol{v}.$$

Enačbo prepišemo v ekvivalentno obliko

$$0 = [A]\,\boldsymbol{v} - \lambda\,\boldsymbol{v} = ([A] - \lambda\,[I])\,\boldsymbol{v}.$$

Enačba bo imela netrivialno rešitev $\boldsymbol{v} \neq \mathbf{0}$ natanko takrat, ko bo determinanta sistema enačb $\det([A] - \lambda\,[I]) = 0$.

Izračunajmo lastne vrednosti in lastne vektorje matrike $[A]$. Rešujemo enačbo

$$[A]\,\boldsymbol{v} = \lambda\,\boldsymbol{v}.$$

Enačbo prepišemo v ekvivalentno obliko

$$0 = [A]\,\boldsymbol{v} - \lambda\,\boldsymbol{v} = ([A] - \lambda\,[I])\,\boldsymbol{v}.$$

Enačba bo imela netrivialno rešitev $\boldsymbol{v} \neq \mathbf{0}$ natanko takrat, ko bo determinanta sistema enačb $\det([A] - \lambda\,[I]) = 0$. Rešujemo enačbo

$$0 = \det([A] - \lambda\,[I])$$

Izračunajmo lastne vrednosti in lastne vektorje matrike $[A]$. Rešujemo enačbo

$$[A]\,\boldsymbol{v} = \lambda\,\boldsymbol{v}.$$

Enačbo prepišemo v ekvivalentno obliko

$$0 = [A]\,\boldsymbol{v} - \lambda\,\boldsymbol{v} = ([A] - \lambda\,[I])\,\boldsymbol{v}.$$

Enačba bo imela netrivialno rešitev $\boldsymbol{v} \neq \mathbf{0}$ natanko takrat, ko bo determinanta sistema enačb $\det([A] - \lambda\,[I]) = 0$. Rešujemo enačbo

$$0 = \det([A] - \lambda\,[I]) = \left| \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right|$$

Izračunajmo lastne vrednosti in lastne vektorje matrike $[A]$. Rešujemo enačbo

$$[A]\,\boldsymbol{v} = \lambda\,\boldsymbol{v}.$$

Enačbo prepišemo v ekvivalentno obliko

$$0 = [A]\,\boldsymbol{v} - \lambda\,\boldsymbol{v} = ([A] - \lambda\,[I])\,\boldsymbol{v}.$$

Enačba bo imela netrivialno rešitev $\boldsymbol{v} \neq \boldsymbol{0}$ natanko takrat, ko bo determinanta sistema enačb $\det([A] - \lambda\,[I]) = 0$. Rešujemo enačbo

$$0 = \det([A] - \lambda\,[I]) = \left| \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = \begin{vmatrix} 1-\lambda & 2 \\ 2 & 4-\lambda \end{vmatrix}.$$

Izračunajmo lastne vrednosti in lastne vektorje matrike $[A]$. Rešujemo enačbo

$$[A]\,\boldsymbol{v} = \lambda\,\boldsymbol{v}.$$

Enačbo prepišemo v ekvivalentno obliko

$$0 = [A]\,\boldsymbol{v} - \lambda\,\boldsymbol{v} = ([A] - \lambda\,[I])\,\boldsymbol{v}.$$

Enačba bo imela netrivialno rešitev $\boldsymbol{v} \neq \mathbf{0}$ natanko takrat, ko bo determinanta sistema enačb $\det([A] - \lambda\,[I]) = 0$. Rešujemo enačbo

$$0 = \det([A] - \lambda\,[I]) = \left| \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} - \lambda\, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = \begin{vmatrix} 1-\lambda & 2 \\ 2 & 4-\lambda \end{vmatrix}.$$

Rešitvi kvadratne enačbe

$$(1-\lambda)\,(4-\lambda) - 4 = 0$$

sta $\lambda_1 = 0$ in $\lambda_2 = 5$.

Izračunajmo še pripadajoča lastna vektorja.

Izračunajmo še pripadajoča lastna vektorja.

Lastni vektor $\boldsymbol{v}_1 = [v_{11}\, v_{21}]^T$ je rešitev enačbe $[A]\,\boldsymbol{v}_1 = \lambda_1\,\boldsymbol{v}_1$.

Izračunajmo še pripadajoča lastna vektorja.

Lastni vektor $\boldsymbol{v}_1 = [v_{11}\, v_{21}]^T$ je rešitev enačbe $[A]\, \boldsymbol{v}_1 = \lambda_1\, \boldsymbol{v}_1$.

Oziroma rešitev enačbe $[A]\, \boldsymbol{v}_1 - \lambda_1\, \boldsymbol{v}_1 = \mathbf{0}$.

Izračunajmo še pripadajoča lastna vektorja.

Lastni vektor $\boldsymbol{v}_1 = [v_{11}\, v_{21}]^T$ je rešitev enačbe $[A]\,\boldsymbol{v}_1 = \lambda_1\,\boldsymbol{v}_1$.

Oziroma rešitev enačbe $[A]\,\boldsymbol{v}_1 - \lambda_1\,\boldsymbol{v}_1 = \mathbf{0}$.

Ali napisano širše

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Izračunajmo še pripadajoča lastna vektorja.

Lastni vektor $\boldsymbol{v}_1 = [v_{11}\; v_{21}]^T$ je rešitev enačbe $[A]\,\boldsymbol{v}_1 = \lambda_1\,\boldsymbol{v}_1$.

Oziroma rešitev enačbe $[A]\,\boldsymbol{v}_1 - \lambda_1\,\boldsymbol{v}_1 = \mathbf{0}$.

Ali napisano širše

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Splošna rešitev enačbe se glasi

$$\boldsymbol{v}_1 = \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = t \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \qquad t \in R.$$

Izračunajmo še pripadajoča lastna vektorja.

Lastni vektor $\boldsymbol{v}_1 = [v_{11}\; v_{21}]^T$ je rešitev enačbe $[A]\, \boldsymbol{v}_1 = \lambda_1\, \boldsymbol{v}_1$.

Oziroma rešitev enačbe $[A]\, \boldsymbol{v}_1 - \lambda_1\, \boldsymbol{v}_1 = \mathbf{0}$.

Ali napisano širše

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Splošna rešitev enačbe se glasi

$$\boldsymbol{v}_1 = \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = t \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \qquad t \in R.$$

Iz množice vseh rešitev bomo izbrali eno, to je enotski vektor

$$\boldsymbol{v}_1 = \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} -2 \\ 1 \end{bmatrix}.$$

Izračunajmo še pripadajoča lastna vektorja.

Lastni vektor $\boldsymbol{v}_1 = [v_{11}\, v_{21}]^T$ je rešitev enačbe $[A]\,\boldsymbol{v}_1 = \lambda_1\,\boldsymbol{v}_1$.

Oziroma rešitev enačbe $[A]\,\boldsymbol{v}_1 - \lambda_1\,\boldsymbol{v}_1 = \mathbf{0}$.

Ali napisano širše

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Splošna rešitev enačbe se glasi

$$\boldsymbol{v}_1 = \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = t \begin{bmatrix} -2 \\ 1 \end{bmatrix}, \qquad t \in R.$$

Iz množice vseh rešitev bomo izbrali eno, to je enotski vektor

$$\boldsymbol{v}_1 = \begin{bmatrix} v_{11} \\ v_{21} \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} -2 \\ 1 \end{bmatrix}.$$

Podobno bi izračunali drugi lastni vektor (ki pripada lastni vrednosti λ_2)

$$\boldsymbol{v}_2 = \begin{bmatrix} v_{21} \\ v_{21} \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Pri matriki $[B]$ postopamo podobno.

Pri matriki $[B]$ postopamo podobno. Najprej izračunamo lastni vrednosti in dobimo $\nu_1 = 0$ in $\nu_2 = a + 1$.

Pri matriki $[B]$ postopamo podobno. Najprej izračunamo lastni vrednosti in dobimo $\nu_1 = 0$ in $\nu_2 = a + 1$. Iz primerjave lastnih vrednosti $\lambda_1 = \nu_1$ in $\lambda_2 = \nu_2$, zaključimo, da mora biti $a = 4$.

Pri matriki $[B]$ postopamo podobno. Najprej izračunamo lastni vrednosti in dobimo $\nu_1 = 0$ in $\nu_2 = a + 1$. Iz primerjave lastnih vrednosti $\lambda_1 = \nu_1$ in $\lambda_2 = \nu_2$, zaključimo, da mora biti $a = 4$. Matriko $[B]$ sedaj lahko zapišemo

$$[B] = \begin{bmatrix} 4 & -4 \\ -1 & 1 \end{bmatrix}.$$

Pri matriki $[B]$ postopamo podobno. Najprej izračunamo lastni vrednosti in dobimo $\nu_1 = 0$ in $\nu_2 = a + 1$. Iz primerjave lastnih vrednosti $\lambda_1 = \nu_1$ in $\lambda_2 = \nu_2$, zaključimo, da mora biti $a = 4$. Matriko $[B]$ sedaj lahko zapišemo

$$[B] = \begin{bmatrix} 4 & -4 \\ -1 & 1 \end{bmatrix}.$$

Izračunamo še pripadajoča lastna vektorja

$$\boldsymbol{u}_1 = \begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{u}_2 = \begin{bmatrix} u_{12} \\ u_{22} \end{bmatrix} = \frac{1}{\sqrt{17}} \begin{bmatrix} 4 \\ -1 \end{bmatrix}.$$

Opazimo, da sta lastna vektorja $\boldsymbol{v}_1$ in $\boldsymbol{v}_2$ pravokotna med seboj.

Pri matriki $[B]$ postopamo podobno. Najprej izračunamo lastni vrednosti in dobimo $\nu_1 = 0$ in $\nu_2 = a + 1$. Iz primerjave lastnih vrednosti $\lambda_1 = \nu_1$ in $\lambda_2 = \nu_2$, zaključimo, da mora biti $a = 4$. Matriko $[B]$ sedaj lahko zapišemo

$$[B] = \begin{bmatrix} 4 & -4 \\ -1 & 1 \end{bmatrix}.$$

Izračunamo še pripadajoča lastna vektorja

$$\boldsymbol{u}_1 = \begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{u}_2 = \begin{bmatrix} u_{12} \\ u_{22} \end{bmatrix} = \frac{1}{\sqrt{17}} \begin{bmatrix} 4 \\ -1 \end{bmatrix}.$$

Opazimo, da sta lastna vektorja $\boldsymbol{v}_1$ in $\boldsymbol{v}_2$ pravokotna med seboj. Velja splošnejša trditev: Lastni vektorji realne simetrične matrike, ki ustrezajo različnim lastnim vrednostim so pravokotni med seboj.

Pri matriki $[B]$ postopamo podobno. Najprej izračunamo lastni vrednosti in dobimo $\nu_1 = 0$ in $\nu_2 = a + 1$. Iz primerjave lastnih vrednosti $\lambda_1 = \nu_1$ in $\lambda_2 = \nu_2$, zaključimo, da mora biti $a = 4$. Matriko $[B]$ sedaj lahko zapišemo

$$[B] = \begin{bmatrix} 4 & -4 \\ -1 & 1 \end{bmatrix}.$$

Izračunamo še pripadajoča lastna vektorja

$$\boldsymbol{u}_1 = \begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{u}_2 = \begin{bmatrix} u_{12} \\ u_{22} \end{bmatrix} = \frac{1}{\sqrt{17}} \begin{bmatrix} 4 \\ -1 \end{bmatrix}.$$

Opazimo, da sta lastna vektorja $\boldsymbol{v}_1$ in $\boldsymbol{v}_2$ pravokotna med seboj. Velja splošnejša trditev: Lastni vektorji realne simetrične matrike, ki ustrezajo različnim lastnim vrednostim so pravokotni med seboj. Za nesimetrične matrike, ta trditev ne velja, saj vektorja $\boldsymbol{u}_1$ in $\boldsymbol{u}_2$ nista pravokotna.

Pri matriki $[B]$ postopamo podobno. Najprej izračunamo lastni vrednosti in dobimo $\nu_1 = 0$ in $\nu_2 = a + 1$. Iz primerjave lastnih vrednosti $\lambda_1 = \nu_1$ in $\lambda_2 = \nu_2$, zaključimo, da mora biti $a = 4$. Matriko $[B]$ sedaj lahko zapišemo

$$[B] = \begin{bmatrix} 4 & -4 \\ -1 & 1 \end{bmatrix}.$$

Izračunamo še pripadajoča lastna vektorja

$$\boldsymbol{u}_1 = \begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{u}_2 = \begin{bmatrix} u_{12} \\ u_{22} \end{bmatrix} = \frac{1}{\sqrt{17}} \begin{bmatrix} 4 \\ -1 \end{bmatrix}.$$

Opazimo, da sta lastna vektorja $\boldsymbol{v}_1$ in $\boldsymbol{v}_2$ pravokotna med seboj. Velja splošnejša trditev: Lastni vektorji realne simetrične matrike, ki ustrezajo različnim lastnim vrednostim so pravokotni med seboj. Za nesimetrične matrike, ta trditev ne velja, saj vektorja $\boldsymbol{u}_1$ in $\boldsymbol{u}_2$ nista pravokotna.

Enačbi

$$
\begin{aligned}
[A]\,\boldsymbol{v}_1 &= \lambda_1\,\boldsymbol{v}_1 \\
[A]\,\boldsymbol{v}_2 &= \lambda_2\,\boldsymbol{v}_2
\end{aligned}
$$

Enačbi

$$
\begin{aligned}
[A]\,\boldsymbol{v}_1 &= \lambda_1\,\boldsymbol{v}_1 \\
[A]\,\boldsymbol{v}_2 &= \lambda_2\,\boldsymbol{v}_2
\end{aligned}
$$

zapišemo lahko v matrični obliki

$$
[A]\,[V] = [V]\,[D], \tag{2}
$$

Enačbi

$$\begin{aligned}[A]\,\boldsymbol{v}_1 &= \lambda_1\,\boldsymbol{v}_1\\ [A]\,\boldsymbol{v}_2 &= \lambda_2\,\boldsymbol{v}_2\end{aligned}$$

zapišemo lahko v matrični obliki

$$[A]\,[V] = [V]\,[D], \tag{2}$$

kjer smo uvedli matriki

$$[V] = \begin{bmatrix}\boldsymbol{v}_1 \ \boldsymbol{v}_2\end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22}\end{bmatrix}, \qquad [D] = \begin{bmatrix}\lambda_1 & 0 \\ 0 & \lambda_2\end{bmatrix}.$$

Enačbi

$$[A]\,\boldsymbol{v}_1 \;=\; \lambda_1\,\boldsymbol{v}_1$$

$$[A]\,\boldsymbol{v}_2 \;=\; \lambda_2\,\boldsymbol{v}_2$$

zapišemo lahko v matrični obliki

$$[A]\,[V] = [V]\,[D], \tag{2}$$

kjer smo uvedli matriki

$$[V] = \begin{bmatrix}\boldsymbol{v}_1 \; \boldsymbol{v}_2\end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}, \qquad [D] = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}.$$

Z uvedbo matrike

$$[U] = \begin{bmatrix}\boldsymbol{u}_1 \; \boldsymbol{u}_2\end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$$

Enačbi

$$
\begin{aligned}
[A]\,\boldsymbol{v}_1 &= \lambda_1\,\boldsymbol{v}_1 \\
[A]\,\boldsymbol{v}_2 &= \lambda_2\,\boldsymbol{v}_2
\end{aligned}
$$

zapišemo lahko v matrični obliki

$$
[A]\,[V] = [V]\,[D], \tag{2}
$$

kjer smo uvedli matriki

$$
[V] = \begin{bmatrix} \boldsymbol{v}_1 \ \boldsymbol{v}_2 \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}, \qquad [D] = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}.
$$

Z uvedbo matrike

$$
[U] = \begin{bmatrix} \boldsymbol{u}_1 \ \boldsymbol{u}_2 \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}
$$

lahko pišemo podobno

$$
[B]\,[U] = [U]\,[D]. \tag{3}
$$

Iz enačbe (2) dobimo

$$[A] = [V]\,[D]\,[V]^{-1},$$

Iz enačbe (2) dobimo

$$[A] = [V]\,[D]\,[V]^{-1},$$

iz enačbe (3) pa

$$[D] = [U]^{-1}\,[B]\,[U].$$

Iz enačbe (2) dobimo

$$[A] = [V]\,[D]\,[V]^{-1},$$

iz enačbe (3) pa

$$[D] = [U]^{-1}\,[B]\,[U].$$

Izraz iz druge enačbe vstavimo v prvo in dobimo

$$[A] = [V]\,[U]^{-1}\,[B]\,[U]\,[V]^{-1}$$

Iz enačbe (2) dobimo

$$[A] = [V]\,[D]\,[V]^{-1},$$

iz enačbe (3) pa

$$[D] = [U]^{-1}\,[B]\,[U].$$

Izraz iz druge enačbe vstavimo v prvo in dobimo

$$[A] = [V]\,[U]^{-1}\,[B]\,[U]\,[V]^{-1} = [P]^{-1}\,[B]\,[P].$$

Iz enačbe (2) dobimo

$$[A] = [V]\,[D]\,[V]^{-1},$$

iz enačbe (3) pa

$$[D] = [U]^{-1}\,[B]\,[U].$$

Izraz iz druge enačbe vstavimo v prvo in dobimo

$$[A] = [V]\,[U]^{-1}\,[B]\,[U]\,[V]^{-1} = [P]^{-1}\,[B]\,[P].$$

Iskana matrika $[P]$ je torej

$$[P] = [U]\,[V]^{-1} = \begin{bmatrix} -0.1986 & 1.1839 \\ -0.7409 & 0.0993 \end{bmatrix}.$$

Iz enačbe (2) dobimo

$$[A] = [V]\,[D]\,[V]^{-1},$$

iz enačbe (3) pa

$$[D] = [U]^{-1}\,[B]\,[U].$$

Izraz iz druge enačbe vstavimo v prvo in dobimo

$$[A] = [V]\,[U]^{-1}\,[B]\,[U]\,[V]^{-1} = [P]^{-1}\,[B]\,[P].$$

Iskana matrika $[P]$ je torej

$$[P] = [U]\,[V]^{-1} = \begin{bmatrix} -0.1986 & 1.1839 \\ -0.7409 & 0.0993 \end{bmatrix}.$$

Lastnih vektorjev ne bi bilo potrebno normirati, zato je to samo ena izmed množice rešitev.

Iz enačbe (2) dobimo

$$[A] = [V]\,[D]\,[V]^{-1},$$

iz enačbe (3) pa

$$[D] = [U]^{-1}\,[B]\,[U].$$

Izraz iz druge enačbe vstavimo v prvo in dobimo

$$[A] = [V]\,[U]^{-1}\,[B]\,[U]\,[V]^{-1} = [P]^{-1}\,[B]\,[P].$$

Iskana matrika $[P]$ je torej

$$[P] = [U]\,[V]^{-1} = \begin{bmatrix} -0.1986 & 1.1839 \\ -0.7409 & 0.0993 \end{bmatrix}.$$

Lastnih vektorjev ne bi bilo potrebno normirati, zato je to samo ena izmed množice rešitev. Lastne vektorje in lastne vrednosti matrike $[A]$ lahko v programu MATLAB izračunamo z ukazoma

Iz enačbe (2) dobimo

$$[A] = [V]\,[D]\,[V]^{-1},$$

iz enačbe (3) pa

$$[D] = [U]^{-1}\,[B]\,[U].$$

Izraz iz druge enačbe vstavimo v prvo in dobimo

$$[A] = [V]\,[U]^{-1}\,[B]\,[U]\,[V]^{-1} = [P]^{-1}\,[B]\,[P].$$

Iskana matrika $[P]$ je torej

$$[P] = [U]\,[V]^{-1} = \begin{bmatrix} -0.1986 & 1.1839 \\ -0.7409 & 0.0993 \end{bmatrix}.$$

Lastnih vektorjev ne bi bilo potrebno normirati, zato je to samo ena izmed množice rešitev. Lastne vektorje in lastne vrednosti matrike $[A]$ lahko v programu MATLAB izračunamo z ukazoma

```
A = [1 2; 2 4]; [V,D] = eig(A);
```

Naloga 22

Naloga 22

Poišči splošno rešitev sistema enačb

$$\begin{aligned} 3\,x_1 - 2\,x_2 - x_3 - x_4 &= 1 \\ 6\,x_1 - 4\,x_2 + 4\,x_3 + 3\,x_4 &= 3 \\ 3\,x_1 - 2\,x_2 + 5\,x_3 + 4\,x_4 &= 2 \\ 9\,x_1 - 6\,x_2 + 3\,x_3 + 2\,x_4 &= 4 \end{aligned}$$

Nalogo lahko rešimo peš, ali z enimo od programov za simbolično računanje kot so npr. MATHEMATICA, MAPLE, AKSIOM, DERIVE ...

Nalogo lahko rešimo peš, ali z enimo od programov za simbolično računanje kot so npr. MATHEMATICA, MAPLE, AKSIOM, DERIVE ...

Nalogo bomo približno rešili z uporabo programa MATLAB.

Nalogo lahko rešimo peš, ali z enimo od programov za simbolično računanje kot so npr. MATHEMATICA, MAPLE, AKSIOM, DERIVE ...
Nalogo bomo približno rešili z uporabo programa MATLAB.
S stališča teoretičnega matematika je taka rešitev seveda sporna, saj z MATLABOM, vsaj brez podpore paketa za simbolično računanje, naloge ne moremo točno rešiti. Vendar je praksa pokazala, da od simbolike vsaj pri večjih sistemih enačb ni kaj prida koristi, zato avtor želi, da bodoči inženir spozna učinkovito numerično orodje za reševanje takšnih in podobnih nalog. To pa MATLAB, vsaj po mnenju večine strokovnjakov, ki se ukvarjajo z numeriko, tudi je.

Nalogo lahko rešimo peš, ali z enimo od programov za simbolično računanje kot so npr. MATHEMATICA, MAPLE, AKSIOM, DERIVE ...

Nalogo bomo približno rešili z uporabo programa MATLAB.

S stališča teoretičnega matematika je taka rešitev seveda sporna, saj z MATLABOM, vsaj brez podpore paketa za simbolično računanje, naloge ne moremo točno rešiti. Vendar je praksa pokazala, da od simbolike vsaj pri večjih sistemih enačb ni kaj prida koristi, zato avtor želi, da bodoči inženir spozna učinkovito numerično orodje za reševanje takšnih in podobnih nalog. To pa MATLAB, vsaj po mnenju večine strokovnjakov, ki se ukvarjajo z numeriko, tudi je.

Kakorkoli že! Omenjeni postopek lahko uporabimo tudi v programih za simbolično računanje.

Najprej nalogo ob uvedbi oznak

$$[A] = \begin{bmatrix} 3 & -2 & -1 & -1 \\ 6 & -4 & 4 & 3 \\ 3 & -2 & 5 & 4 \\ 9 & -6 & 3 & 2 \end{bmatrix}, \qquad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 4 \end{bmatrix} \tag{4}$$

Najprej nalogo ob uvedbi oznak

$$[A] = \begin{bmatrix} 3 & -2 & -1 & -1 \\ 6 & -4 & 4 & 3 \\ 3 & -2 & 5 & 4 \\ 9 & -6 & 3 & 2 \end{bmatrix}, \qquad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 4 \end{bmatrix} \tag{4}$$

prepišemo v matrično obliko

$$[A]\,\boldsymbol{x} = \boldsymbol{b}. \tag{5}$$

Najprej nalogo ob uvedbi oznak

$$[A] = \begin{bmatrix} 3 & -2 & -1 & -1 \\ 6 & -4 & 4 & 3 \\ 3 & -2 & 5 & 4 \\ 9 & -6 & 3 & 2 \end{bmatrix}, \qquad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 4 \end{bmatrix} \tag{4}$$

prepišemo v matrično obliko

$$[A]\,\boldsymbol{x} = \boldsymbol{b}. \tag{5}$$

Nalogo bomo rešili z uporabo algoritma na naslednji strani.

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev.

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev. Pojdi na korak (4).

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev. Pojdi na korak (4).
Če ji ne zadošča, rešitve ni.

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev. Pojdi na korak (4).
Če ji ne zadošča, rešitve ni. KONEC.

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev. Pojdi na korak (4).
Če ji ne zadošča, rešitve ni. KONEC.

(4) Izračunaj jedro matrike $[A]$.

```
Ker = null(A);
```

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev. Pojdi na korak (4).
Če ji ne zadošča, rešitve ni. KONEC.

(4) Izračunaj jedro matrike $[A]$.

```
Ker = null(A);
```

Če je jedro prazna matrika, potem obstaja natanko ena rešitev $\boldsymbol{x} = \boldsymbol{x}_0$.

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev. Pojdi na korak (4).
Če ji ne zadošča, rešitve ni. KONEC.

(4) Izračunaj jedro matrike $[A]$.

```
Ker = null(A);
```

Če je jedro prazna matrika, potem obstaja natanko ena rešitev $\boldsymbol{x} = \boldsymbol{x}_0$. KONEC.

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev. Pojdi na korak (4).
Če ji ne zadošča, rešitve ni. KONEC.

(4) Izračunaj jedro matrike $[A]$.

```
Ker = null(A);
```

Če je jedro prazna matrika, potem obstaja natanko ena rešitev $\boldsymbol{x} = \boldsymbol{x}_0$. KONEC.
Če jedro ni prazna matrika in je oblike `Ker` $= [\boldsymbol{x}_1\, \boldsymbol{x}_2 \ldots \boldsymbol{x}_k]$, potem obstaja neskončno mnogo rešitev.

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev. Pojdi na korak (4).
Če ji ne zadošča, rešitve ni. KONEC.

(4) Izračunaj jedro matrike $[A]$.

```
Ker = null(A);
```

Če je jedro prazna matrika, potem obstaja natanko ena rešitev $\boldsymbol{x} = \boldsymbol{x}_0$. KONEC.
Če jedro ni prazna matrika in je oblike $\texttt{Ker} = [\boldsymbol{x}_1\,\boldsymbol{x}_2 \ldots \boldsymbol{x}_k]$, potem obstaja neskončno mnogo rešitev. Splošna rešitev se glasi

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev. Pojdi na korak (4).
Če ji ne zadošča, rešitve ni. KONEC.

(4) Izračunaj jedro matrike $[A]$.

```
Ker = null(A);
```

Če je jedro prazna matrika, potem obstaja natanko ena rešitev $\boldsymbol{x} = \boldsymbol{x}_0$. KONEC.
Če jedro ni prazna matrika in je oblike $\texttt{Ker} = [\boldsymbol{x}_1\,\boldsymbol{x}_2 \ldots \boldsymbol{x}_k]$, potem obstaja neskončno mnogo rešitev. Splošna rešitev se glasi

$$\boldsymbol{x} = \boldsymbol{x}_0 + \alpha_1\,\boldsymbol{x}_1 + \alpha_2\,\boldsymbol{x}_2 + \cdots + \alpha_k\,\boldsymbol{x}_k, \qquad \alpha_1, \alpha_2\, \ldots, \alpha_k \in \mathbb{R}.$$

Algoritem:

(1) Vnesi podatka $[A]$ in $\boldsymbol{b}$. Npr.

```
A = [3 -2 -1 -1; 6 -4 4 3; 3 -2 5 4; 9 -6 3 2];
b = [1 3 2 4]';
```

(2) Izračunaj rešitev $\boldsymbol{x}_0$ po metodi najmajših kvadratov.

```
x0 = pinv(A) * b;
```

(3) Preveri, če rešitev $\boldsymbol{x}_0$, dobljena po metodi najmajših kvadratov zadošča enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.
Če ji zadošča, potem obstaja vsaj ena rešitev. Pojdi na korak (4).
Če ji ne zadošča, rešitve ni. KONEC.

(4) Izračunaj jedro matrike $[A]$.

```
Ker = null(A);
```

Če je jedro prazna matrika, potem obstaja natanko ena rešitev $\boldsymbol{x} = \boldsymbol{x}_0$. KONEC.
Če jedro ni prazna matrika in je oblike $\texttt{Ker} = [\boldsymbol{x}_1\,\boldsymbol{x}_2 \ldots \boldsymbol{x}_k]$, potem obstaja neskončno mnogo rešitev. Splošna rešitev se glasi

$$\boldsymbol{x} = \boldsymbol{x}_0 + \alpha_1\,\boldsymbol{x}_1 + \alpha_2\,\boldsymbol{x}_2 + \cdots + \alpha_k\,\boldsymbol{x}_k, \qquad \alpha_1, \alpha_2\ \ldots, \alpha_k \in \mathbb{R}.$$

KONEC.

Anlgoritem zasluži kratek komentar.

Anlgoritem zasluži kratek komentar.

Naj bo $[A]$ matrika z m vrsticami in n stolpci.

Anlgoritem zasluži kratek komentar.

Naj bo $[A]$ matrika z m vrsticami in n stolpci. Rešitev $\boldsymbol{x}_0$, dobljena po metodi najmanjših kvadratov minimalizira funkcijo

$$f : \boldsymbol{x} \mapsto \|[A]\,\boldsymbol{x} - \boldsymbol{b}\| = \sqrt{\sum_{i=1}^{m} (([A]\,\boldsymbol{x})_i - b_i)^2}$$

po vseh $\boldsymbol{x}$ iz $\mathbb{R}^n$.

Anlgoritem zasluži kratek komentar.

Naj bo $[A]$ matrika z m vrsticami in n stolpci. Rešitev $\boldsymbol{x}_0$, dobljena po metodi najmanjših kvadratov minimalizira funkcijo

$$f : \boldsymbol{x} \mapsto \|[A]\,\boldsymbol{x} - \boldsymbol{b}\| = \sqrt{\sum_{i=1}^{m} (([A]\,\boldsymbol{x})_i - b_i)^2}$$

po vseh $\boldsymbol{x}$ iz $\mathbb{R}^n$.

Oznaki $([A]\,\boldsymbol{x})_i$ in b_i predstavljata i–ta elementa vektorjev $[A]\,\boldsymbol{x}$ in $\boldsymbol{b}$.

Anlgoritem zasluži kratek komentar.

Naj bo $[A]$ matrika z m vrsticami in n stolpci. Rešitev $\boldsymbol{x}_0$, dobljena po metodi najmanjših kvadratov minimalizira funkcijo

$$f : \boldsymbol{x} \mapsto \|[A]\,\boldsymbol{x} - \boldsymbol{b}\| = \sqrt{\sum_{i=1}^{m} (([A]\,\boldsymbol{x})_i - b_i)^2}$$

po vseh $\boldsymbol{x}$ iz $\mathbb{R}^n$.

Oznaki $([A]\,\boldsymbol{x})_i$ in b_i predstavljata i–ta elementa vektorjev $[A]\,\boldsymbol{x}$ in $\boldsymbol{b}$.

Razlago ukaza `pinv` v MATLABU dobimo z ukazom `help pinv`, razlago ukaza `null` s `help null`.

Anlgoritem zasluži kratek komentar.

Naj bo $[A]$ matrika z m vrsticami in n stolpci. Rešitev $\boldsymbol{x}_0$, dobljena po metodi najmanjših kvadratov minimalizira funkcijo

$$f : \boldsymbol{x} \mapsto \|[A]\,\boldsymbol{x} - \boldsymbol{b}\| = \sqrt{\sum_{i=1}^{m} (([A]\,\boldsymbol{x})_i - b_i)^2}$$

po vseh $\boldsymbol{x}$ iz $\mathbb{R}^n$.

Oznaki $([A]\,\boldsymbol{x})_i$ in b_i predstavljata i–ta elementa vektorjev $[A]\,\boldsymbol{x}$ in $\boldsymbol{b}$.

Razlago ukaza `pinv` v MATLABU dobimo z ukazom `help pinv`, razlago ukaza `null` s `help null`.

Algoritem je mogoče optimizirati glede same hitrosti delovanja, saj tule dvakrat zapored opravljamo singularni razcep matrike $[A]$, vendar ga zaradi večje preglednosti raje ohranimo v takšni obliki.

Anlgoritem zasluži kratek komentar.

Naj bo $[A]$ matrika z m vrsticami in n stolpci. Rešitev $\boldsymbol{x}_0$, dobljena po metodi najmanjših kvadratov minimalizira funkcijo

$$f : \boldsymbol{x} \mapsto \|[A]\,\boldsymbol{x} - \boldsymbol{b}\| = \sqrt{\sum_{i=1}^{m} (([A]\,\boldsymbol{x})_i - b_i)^2}$$

po vseh $\boldsymbol{x}$ iz $\mathbb{R}^n$.

Oznaki $([A]\,\boldsymbol{x})_i$ in b_i predstavljata i–ta elementa vektorjev $[A]\,\boldsymbol{x}$ in $\boldsymbol{b}$.

Razlago ukaza `pinv` v MATLABU dobimo z ukazom `help pinv`, razlago ukaza `null` s `help null`.

Algoritem je mogoče optimizirati glede same hitrosti delovanja, saj tule dvakrat zapored opravljamo singularni razcep matrike $[A]$, vendar ga zaradi večje preglednosti raje ohranimo v takšni obliki.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.
Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.
Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res?

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.
Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.
Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$. Ker za vsak vektor $\boldsymbol{x}_i$ iz jedra matrike $[A]$ velja

$[A]\,\boldsymbol{x}_i = \mathbf{0}$,

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.
Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$. Ker za vsak vektor $\boldsymbol{x}_i$ iz jedra matrike $[A]$ velja $[A]\,\boldsymbol{x}_i = \boldsymbol{0}$, velja tudi $[A]\,\boldsymbol{x} = [A]\,\boldsymbol{x}_0 + \sum_{i=1}^{k}\alpha_i\,[A]\,\boldsymbol{x}_i = [A]\,(\boldsymbol{x}_0 + \sum_{i=1}^{k}\alpha_i\,\boldsymbol{x}_i) = \boldsymbol{b}$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.
Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$. Ker za vsak vektor $\boldsymbol{x}_i$ iz jedra matrike $[A]$ velja $[A]\,\boldsymbol{x}_i = \mathbf{0}$, velja tudi $[A]\,\boldsymbol{x} = [A]\,\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,[A]\,\boldsymbol{x}_i = [A]\,(\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i) = \boldsymbol{b}$. Torej je vektor $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i, \quad \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathrm{Ker}([A])$ tudi rešitev.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$. Ker za vsak vektor $\boldsymbol{x}_i$ iz jedra matrike $[A]$ velja $[A]\,\boldsymbol{x}_i = \boldsymbol{0}$, velja tudi $[A]\,\boldsymbol{x} = [A]\,\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,[A]\,\boldsymbol{x}_i = [A]\,(\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i) = \boldsymbol{b}$.

Torej je vektor $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i, \quad \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathrm{Ker}([A])$ tudi rešitev.

Ali smo zajeli vse rešitve?

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$. Ker za vsak vektor $\boldsymbol{x}_i$ iz jedra matrike $[A]$ velja $[A]\,\boldsymbol{x}_i = \boldsymbol{0}$, velja tudi $[A]\,\boldsymbol{x} = [A]\,\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,[A]\,\boldsymbol{x}_i = [A]\,(\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i) = \boldsymbol{b}$.
Torej je vektor $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i, \quad \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathrm{Ker}([A])$ tudi rešitev.

Ali smo zajeli vse rešitve? Označimo poljubno rešitev z $\boldsymbol{y}$. Potem velja $[A]\,\boldsymbol{y} = \boldsymbol{b}$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.
Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$. Ker za vsak vektor $\boldsymbol{x}_i$ iz jedra matrike $[A]$ velja $[A]\,\boldsymbol{x}_i = \boldsymbol{0}$, velja tudi $[A]\,\boldsymbol{x} = [A]\,\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,[A]\,\boldsymbol{x}_i = [A]\,(\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i) = \boldsymbol{b}$.

Torej je vektor $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i, \quad \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathrm{Ker}([A])$ tudi rešitev.

Ali smo zajeli vse rešitve? Označimo poljubno rešitev z $\boldsymbol{y}$. Potem velja $[A]\,\boldsymbol{y} = \boldsymbol{b}$.

Ker obstaja $\boldsymbol{x}_0$ (to je lahko kar $\boldsymbol{y}$), da velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ velja $[A]\,(\boldsymbol{y} - \boldsymbol{x}_0) = \boldsymbol{0}$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$. Ker za vsak vektor $\boldsymbol{x}_i$ iz jedra matrike $[A]$ velja $[A]\,\boldsymbol{x}_i = \mathbf{0}$, velja tudi $[A]\,\boldsymbol{x} = [A]\,\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,[A]\,\boldsymbol{x}_i = [A]\,(\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i) = \boldsymbol{b}$. Torej je vektor $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i, \quad \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathrm{Ker}([A])$ tudi rešitev.

Ali smo zajeli vse rešitve? Označimo poljubno rešitev z $\boldsymbol{y}$. Potem velja $[A]\,\boldsymbol{y} = \boldsymbol{b}$. Ker obstaja $\boldsymbol{x}_0$ (to je lahko kar $\boldsymbol{y}$), da velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ velja $[A]\,(\boldsymbol{y} - \boldsymbol{x}_0) = \mathbf{0}$. Torej je vektor $\boldsymbol{y} - \boldsymbol{x}_0$ v jedru $\mathtt{Ker}([A])$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$. Ker za vsak vektor $\boldsymbol{x}_i$ iz jedra matrike $[A]$ velja $[A]\,\boldsymbol{x}_i = \boldsymbol{0}$, velja tudi $[A]\,\boldsymbol{x} = [A]\,\boldsymbol{x}_0 + \sum_{i=1}^k \alpha_i\,[A]\,\boldsymbol{x}_i = [A]\,(\boldsymbol{x}_0 + \sum_{i=1}^k \alpha_i\,\boldsymbol{x}_i) = \boldsymbol{b}$. Torej je vektor $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{i=1}^k \alpha_i\,\boldsymbol{x}_i, \quad \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathrm{Ker}([A])$ tudi rešitev.

Ali smo zajeli vse rešitve? Označimo poljubno rešitev z $\boldsymbol{y}$. Potem velja $[A]\,\boldsymbol{y} = \boldsymbol{b}$. Ker obstaja $\boldsymbol{x}_0$ (to je lahko kar $\boldsymbol{y}$), da velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ velja $[A]\,(\boldsymbol{y} - \boldsymbol{x}_0) = \boldsymbol{0}$. Torej je vektor $\boldsymbol{y} - \boldsymbol{x}_0$ v jedru $\mathtt{Ker}([A])$. Zato ga lahko zapišemo kot linearno kombinacijo baznih vektorjev iz jedra.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$. Ker za vsak vektor $\boldsymbol{x}_i$ iz jedra matrike $[A]$ velja $[A]\,\boldsymbol{x}_i = \boldsymbol{0}$, velja tudi $[A]\,\boldsymbol{x} = [A]\,\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,[A]\,\boldsymbol{x}_i = [A]\,(\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i) = \boldsymbol{b}$. Torej je vektor $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i, \quad \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathrm{Ker}([A])$ tudi rešitev.

Ali smo zajeli vse rešitve? Označimo poljubno rešitev z $\boldsymbol{y}$. Potem velja $[A]\,\boldsymbol{y} = \boldsymbol{b}$. Ker obstaja $\boldsymbol{x}_0$ (to je lahko kar $\boldsymbol{y}$), da velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ velja $[A]\,(\boldsymbol{y} - \boldsymbol{x}_0) = \boldsymbol{0}$. Torej je vektor $\boldsymbol{y} - \boldsymbol{x}_0$ v jedru $\mathtt{Ker}([A])$. Zato ga lahko zapišemo kot linearno kombinacijo baznih vektorjev iz jedra. Velja torej $\boldsymbol{y} - \boldsymbol{x}_0 = \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i$.

Pravilnosti delovanja algoritma ni težko preveriti. Označimo psevdoinverz matrike $[A]$ z $[A]^+$. V MATLABU izračunamo psevdoinverz matrike $[A]$ z ukazom `pinv(A)`. Iz učbenikov za numerično algebro preberemo, da vektor $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f na prejšnji strani. Če je takšnih rešitev več, program vrne takšnega z minimalno normo.

Preverimo korak 3. Če velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ potem očitno imamo vsaj eno rešitev. Če velja $[A]\,\boldsymbol{x}_0 \neq \boldsymbol{b}$, potem je tudi $f(\boldsymbol{x}_0) > 0$. Trdimo, da potem rešitve nimamo. Res? Pa recimo, da jo imamo. Označimo jo z $\boldsymbol{x}_1$. Potem očitno velja $[A]\,\boldsymbol{x}_1 = \boldsymbol{b}$. Vrednost funkcije $f(\boldsymbol{x}_1) = 0 < f(\boldsymbol{x}_0)$. To pa je v protislovju s tem, da rešitev $\boldsymbol{x}_0 = [A]^+\boldsymbol{b}$ minimalizira funkcijo f.

Še korak 4. Naj velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$. Ker za vsak vektor $\boldsymbol{x}_i$ iz jedra matrike $[A]$ velja $[A]\,\boldsymbol{x}_i = \boldsymbol{0}$, velja tudi $[A]\,\boldsymbol{x} = [A]\,\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,[A]\,\boldsymbol{x}_i = [A]\,(\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i) = \boldsymbol{b}$. Torej je vektor $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i, \quad \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathrm{Ker}([A])$ tudi rešitev.

Ali smo zajeli vse rešitve? Označimo poljubno rešitev z $\boldsymbol{y}$. Potem velja $[A]\,\boldsymbol{y} = \boldsymbol{b}$. Ker obstaja $\boldsymbol{x}_0$ (to je lahko kar $\boldsymbol{y}$), da velja $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ velja $[A]\,(\boldsymbol{y} - \boldsymbol{x}_0) = \boldsymbol{0}$. Torej je vektor $\boldsymbol{y} - \boldsymbol{x}_0$ v jedru $\texttt{Ker}([A])$. Zato ga lahko zapišemo kot linearno kombinacijo baznih vektorjev iz jedra. Velja torej $\boldsymbol{y} - \boldsymbol{x}_0 = \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i$. Torej tudi $\boldsymbol{y}$ pripada množici rešitev $\left\{\boldsymbol{x}_0 + \sum_{i=1}^{k} \alpha_i\,\boldsymbol{x}_i, \qquad \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \texttt{Ker}([A])\right\}$.

Rešitev naloge:

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.
  Dobimo $\boldsymbol{x}0 = \begin{bmatrix} 0.2720 & -0.1814 & 0.1058 & 0.0730 \end{bmatrix}^T$.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.
  Dobimo $\boldsymbol{x}0 = \begin{bmatrix} 0.2720 & -0.1814 & 0.1058 & 0.0730 \end{bmatrix}^T$.
- Preverimo, da je enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ zadoščeno.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.
  Dobimo $\boldsymbol{x}0 = \begin{bmatrix} 0.2720 & -0.1814 & 0.1058 & 0.0730 \end{bmatrix}^T$.
- Preverimo, da je enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ zadoščeno. Rešitev je torej ena sama ali pa jih je neskončno mnogo.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.
  Dobimo $\boldsymbol{x}0 = \begin{bmatrix} 0.2720 & -0.1814 & 0.1058 & 0.0730 \end{bmatrix}^T$.
- Preverimo, da je enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ zadoščeno. Rešitev je torej ena sama ali pa jih je neskončno mnogo. Da bi to ugotovili, gremo na korak (4).

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.
  Dobimo $\boldsymbol{x}0 = \begin{bmatrix} 0.2720 & -0.1814 & 0.1058 & 0.0730 \end{bmatrix}^T$.
- Preverimo, da je enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ zadoščeno. Rešitev je torej ena sama ali pa jih je neskončno mnogo. Da bi to ugotovili, gremo na korak (4).
- Izračunamo jedro matrike `Ker = null(A)`.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.
  Dobimo $\boldsymbol{x}0 = \begin{bmatrix} 0.2720 & -0.1814 & 0.1058 & 0.0730 \end{bmatrix}^T$.
- Preverimo, da je enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ zadoščeno. Rešitev je torej ena sama ali pa jih je neskončno mnogo. Da bi to ugotovili, gremo na korak (4).
- Izračunamo jedro matrike `Ker = null(A)`. Dobimo

$$\texttt{Ker} = \begin{bmatrix} 0 & 0.5555 \\ -0.0639 & 0.8298 \\ -0.6389 & -0.0340 \\ 0.7667 & 0.0408 \end{bmatrix}.$$

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.
  Dobimo $\boldsymbol{x}0 = \begin{bmatrix} 0.2720 & -0.1814 & 0.1058 & 0.0730 \end{bmatrix}^T$.
- Preverimo, da je enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ zadoščeno. Rešitev je torej ena sama ali pa jih je neskončno mnogo. Da bi to ugotovili, gremo na korak (4).
- Izračunamo jedro matrike `Ker = null(A)`. Dobimo

$$\texttt{Ker} = \begin{bmatrix} 0 & 0.5555 \\ -0.0639 & 0.8298 \\ -0.6389 & -0.0340 \\ 0.7667 & 0.0408 \end{bmatrix}.$$

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.
  Dobimo $\boldsymbol{x}0 = \begin{bmatrix} 0.2720 & -0.1814 & 0.1058 & 0.0730 \end{bmatrix}^T$.
- Preverimo, da je enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ zadoščeno. Rešitev je torej ena sama ali pa jih je neskončno mnogo. Da bi to ugotovili, gremo na korak (4).
- Izračunamo jedro matrike `Ker = null(A)`. Dobimo

$$\texttt{Ker} = \begin{bmatrix} 0 & 0.5555 \\ -0.0639 & 0.8298 \\ -0.6389 & -0.0340 \\ 0.7667 & 0.0408 \end{bmatrix}.$$

  Vidimo, da je jedro predstavlja matrika, sestavljena iz dveh stolpcev.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.
  Dobimo $\boldsymbol{x}0 = \begin{bmatrix} 0.2720 & -0.1814 & 0.1058 & 0.0730 \end{bmatrix}^T$.
- Preverimo, da je enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ zadoščeno. Rešitev je torej ena sama ali pa jih je neskončno mnogo. Da bi to ugotovili, gremo na korak (4).
- Izračunamo jedro matrike `Ker = null(A)`. Dobimo

$$\mathtt{Ker} = \begin{bmatrix} 0 & 0.5555 \\ -0.0639 & 0.8298 \\ -0.6389 & -0.0340 \\ 0.7667 & 0.0408 \end{bmatrix}.$$

  Vidimo, da je jedro predstavlja matrika, sestavljena iz dveh stolpcev. Obstaja torej dvo–parametrična družina rešitev.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.
  Dobimo $\boldsymbol{x}0 = \begin{bmatrix}0.2720 & -0.1814 & 0.1058 & 0.0730\end{bmatrix}^T$.
- Preverimo, da je enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ zadoščeno. Rešitev je torej ena sama ali pa jih je neskončno mnogo. Da bi to ugotovili, gremo na korak (4).
- Izračunamo jedro matrike `Ker = null(A)`. Dobimo

$$\texttt{Ker} = \begin{bmatrix} 0 & 0.5555 \\ -0.0639 & 0.8298 \\ -0.6389 & -0.0340 \\ 0.7667 & 0.0408 \end{bmatrix}.$$

Vidimo, da je jedro predstavlja matrika, sestavljena iz dveh stolpcev. Obstaja torej dvo–parametrična družina rešitev. Splošno rešitev zapišemo kot

$$\boldsymbol{x}_0 = \begin{bmatrix} 0.2720 \\ -0.1814 \\ 0.1058 \\ 0.0730 \end{bmatrix} + \alpha_1 \begin{bmatrix} 0 \\ -0.0639 \\ -0.6389 \\ 0.7667 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0.5555 \\ 0.8298 \\ -0.0340 \\ 0.0408 \end{bmatrix}, \qquad \alpha_1, \alpha_2 \in \mathbb{R}.$$

Poiščimo še rešitev sistema enačb:

$$\begin{aligned} 3\,x_1 - 2\,x_2 - x_3 - x_4 &= 1 \\ 6\,x_1 - 4\,x_2 + 4\,x_3 + 3\,x_4 &= 3 \\ 3\,x_1 - 2\,x_2 + 5\,x_3 + 4\,x_4 &= 2 \\ 9\,x_1 - 6\,x_2 + 3\,x_3 + 2\,x_4 &= 3 \end{aligned}$$

Poiščimo še rešitev sistema enačb:

$$\begin{aligned} 3\,x_1 - 2\,x_2 - x_3 - x_4 &= 1 \\ 6\,x_1 - 4\,x_2 + 4\,x_3 + 3\,x_4 &= 3 \\ 3\,x_1 - 2\,x_2 + 5\,x_3 + 4\,x_4 &= 2 \\ 9\,x_1 - 6\,x_2 + 3\,x_3 + 2\,x_4 &= 3 \end{aligned}$$

Omenjena naloga se od prvotne razlikuje samo po desni strani tj. vektorju $\boldsymbol{b} = [1\,3\,2\,3]^T$.

Rešitev naloge:

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`. Dobimo

$$\boldsymbol{x}_0 = \begin{bmatrix} 0.2091 \\ -0.1394 \\ 0.1369 \\ 0.1024 \end{bmatrix}.$$

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`. Dobimo

$$\boldsymbol{x}_0 = \begin{bmatrix} 0.2091 \\ -0.1394 \\ 0.1369 \\ 0.1024 \end{bmatrix}.$$

- Enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ **ni** zadoščeno.

Rešitev naloge:

- Vnesemo matriko $[A]$ in vektor desnih strani $\boldsymbol{b}$.
- Izračunamo vektor $\boldsymbol{x}_0$ po enačbi `x0 = pinv(A) * b`. Dobimo

$$\boldsymbol{x}_0 = \begin{bmatrix} 0.2091 \\ -0.1394 \\ 0.1369 \\ 0.1024 \end{bmatrix}.$$

- Enačbi $[A]\,\boldsymbol{x}_0 = \boldsymbol{b}$ **ni** zadoščeno. Rešitve torej ni.