Šolski center Novo mesto

Srednja elektro šola in tehniška gimnazija

Šegova ulica 112

8000 Novo mesto

Z MATRIKAMI NAD RAVNINO

Raziskovalna naloga

Področje: Matematika

Avtorja: Nik Zupančič in Nejc Hočevar

Mentorica: mag. Simona Pustavrh, prof.

Novo mesto, 2021

Povzetek

Raziskovalna naloga govori o osnovah matrik ter njihovi uporabi pri linearnih transformacijah ravnine. Najprej se loteva vprašanja, kaj sploh so matrike, katere vrste matrik poznamo ter računskih operacij z njimi (seštevanje, odštevanje, množenje s skalarjem, množenje, transponiranje in inverz). Nato so predstavljene osnovne linearne transformacije ravnine, ki jih lahko opišemo z matrikami. K nalogi so kot izdelek priložene še animacije linearnih transformacij v Geogebri ter matrični kalkulator v programskih jezikih HTML in JavaScript.

Ključne besede: matrike, linearne transformacije, Java, JavaScript, HTML

Abstract

Our research paper deals with basic principles of matrices and how to use them when transforming a plane. In the first part we define what matrices are, which is followed by identifying types of matrices most commonly used, to finally focus on how to do different mathematical processes with them (addition, subtraction, multiplication with scalar, multiplication, transposing and inversion). The research continues with the description of how matrices act as basic linear plane transformations. To visualise things better, we also created linear transformation animations in GeoGebra and matrix calculator in programming languages HTML and JavaScript.

Key words: matrices, linear transformations, Java, JavaScript, HTML.

Kazalo vsebine

Kazalo slik

Kazalo primerov

1 UVOD

Ker naju veselita matematika in računalništvo, sva se odločila raziskati matrike in njihovo uporabo, saj je čar matematike prav v njeni vsestranskosti. Matrike so nama na začetku predstavljale čisto novo področje, zato sva svoje raziskovanje začela z definicijo matrike, nato pa sva se lotila še lastnosti in računskih operacij z njimi. Glede na to, da sva želela spoznati še njihov praktični pomen, sva raziskala tudi linearne transformacije ravnine ter načine, kako izvajamo računske operacije z matrikami v različnih programskih jezikih. Izdelala sva animacije v GeoGebri, s katerimi sva prikazala transformacije ravnine, za primer pa sva uporabila zanimivo sliko. Spoznala sva tudi, kako izvedemo računske operacije z matrikami v Javi, s pomočjo HTML-ja ter JavaScript-a pa sva izdelala še matrični kalkulator matrik dimenzij 2×2. Ker sta programska jezika v angleščini, sva tudi kalkulator naredila v angleškem jeziku.

2 MATRIKE

Matrika je pravokotna shema $m \cdot n$ elementov, ki so razporejeni v m vrstic in n stolpcev. Matrike označujemo z velikimi tiskanimi črkami (npr. A, B, C …), njihove *dimenzije* (rede) pa z $m \times n$.

Primer 2.1: Matrika 2 x 3

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

Dimenzije matrike A so 2×3, njene vrstice so $3 \quad 7 \quad 9$ in $5 \quad -2 \quad 3$, stolpci pa $\begin{matrix} 3 \\ 5 \end{matrix}$, $\begin{matrix} 7 \\ -2 \end{matrix}$ in $\begin{matrix} 9 \\ 3 \end{matrix}$.

Množico vseh matrik z m vrsticami in n stolpci označimo z $\mathbb{R}^{m \times n}$. Vsakemu *elementu matrike* lahko določimo mesto v matriki tako, da povemo, v kateri vrstici in stolpcu leži. Vrstice označujemo s črko i, stolpce pa z j. Element a_{ij} tako leži v i-ti vrstici in j-tem stolpcu. V splošnem matriko $A \in \mathbb{R}^{m \times n}$ zapišemo kot:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Matriki sta **enaki,** kadar se ujemata v vseh istoležnih elementih.

Elemente matrike A, katerih mesto ustreza predpisu a_{ii}, imenujemo **glavna diagonala** matrike A.

Matrike reda $\mathbb{R}^{n \times n}$ imenujemo **kvadratne matrike**. Le-te imajo enako število stolpcev in vrstic.

Primer 2.2: Diagonala kvadratne matrike

Na diagonali kvadratne matrike $A = \begin{bmatrix} 2 & 1 & 5 \\ 1 & 5 & 7 \\ 3 & 4 & 9 \end{bmatrix}$ so elementi $\begin{matrix} 2 & & \\ & 5 & \\ & & 9 \end{matrix}$.

Kvadratnim matrikam lahko izračunamo tudi **determinanto**.

Definicija determinante reda *n*, ki jo priredimo kvadratni matriki dimenzije $n \times n$, je precej zapletena. Ker pa bomo v nadaljevanju potrebovali le determinanto reda 2, ki jo določimo matrikam dimenzije 2×2, navajava le to definicijo, ki smo jo spoznali pri pouku že v prvem letniku:

Determinanta matrike $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ je enaka $\det(A) = ad - bc$.

3 POSEBNI PRIMERI MATRIK

I. **Ničelne matrike** imajo vse elemente enake 0.

Primer 3.1: Ničelna matrika

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

II. **Skalarji** so matrike razsežnosti 1×1.

Primer 3.2: Skalar

$$A = [4]$$

III. **Stolpčni vektorji** so matrike razsežnosti $m \times 1$.

Primer 3.3: Stolpični vektor

$$A = \begin{bmatrix} 7 \\ 3 \\ 2 \end{bmatrix}$$

IV. **Vrstični vektorji** so matrike razsežnosti $1 \times n$.

Primer 3.4: Vrstični vektor

$$A = \begin{bmatrix} 5 & 8 & 1 \end{bmatrix}$$

V. **Kvadratne matrike so** matrike razsežnosti $n \times n$.

Primer 3.5: Kvadratna matrika

$$A = \begin{bmatrix} 7 & 9 \\ 3 & 1 \end{bmatrix}$$

VI. **Simetralne (simetrične) matrike** so matrike, ki se ne spremenijo, če zamenjamo istoležne stolpce in vrstice.

Primer 3.6: Simetralna (simetrična) matrika

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

VII. **Diagonalne matrike** so matrike, ki imajo elemente izven diagonale enake nič.

Primer 3.7: Diagonalna matrika

$$A = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 7 \end{bmatrix}$$

VIII. **Skalarne matrike** so diagonalne matrike, ki imajo vse elemente na diagonali enake.

Primer 3.8: Skalarna matrika

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

IX. **Enotske (identične) matrike** so diagonalne matrike, ki imajo vrednost vseh elementov na diagonali enako 1. Označujemo jih z I ali E.

Primer 3.9: Enotska (identična) matrika

$$I = E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

X. **Spodnje/zgornje-trikotne matrike** so matrike, ki imajo vse elemente nad/pod diagonalo enake nič.

Primer 3.10: Spodnje-trikotna matrika

$$A = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 8 & 1 & 0 & 0 \\ 7 & 5 & 2 & 0 \\ 23 & 9 & 4 & 6 \end{bmatrix}$$

Primer 3.11: Zgornje-trikotna matrika

$$A = \begin{bmatrix} 5 & 8 & 4 & 7 \\ 0 & -6 & 3 & 2 \\ 0 & 0 & 15 & -9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4 ALGEBRAIČNE OPERACIJE NA MATRIKAH

4.1 SEŠTEVANJE IN ODŠTEVANJE MATRIK

Seštevamo (odštevamo) lahko le matriki *enakih dimenzij*, seštejemo (odštejemo) pa ju tako, da seštejemo (odštejemo) njune istoležne elemente:

$$A \pm B = (a_{ij}) \pm (b_{ij}) = (a_{ij} \pm b_{ij})$$

Primer 4.1: Seštevanje matrik

$$\begin{bmatrix} -3 & 2 \\ 5 & 7 \end{bmatrix} + \begin{bmatrix} 5 & 1 \\ 4 & -8 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 9 & -1 \end{bmatrix}$$

Lastnosti seštevanja matrik:

1. Komutativnost: $A + B = B + A$

   Dokaz 4.1: Komutativnost seštevanja

   Komutativnost seštevanja matrik sledi iz komutativnosti realnih števil:

$$A + B = (a_{ij}) + (b_{ij}) = (a_{ij} + b_{ij}) = (b_{ij} + a_{ij}) = B + A$$

2. Asociativnost: $(A + B) + C = A + (B + C)$

   Dokaz 4.2: Asociativnost seštevanja

   Tudi asociativnost seštevanja matrik sledi iz asociativnosti realnih števil:

$$(A + B) + C = \left((a_{ij}) + (b_{ij})\right) + (c_{ij}) = (a_{ij} + b_{ij}) + (c_{ij}) = (a_{ij} + b_{ij} + c_{ij}) =$$
$$= (a_{ij}) + (b_{ij} + c_{ij}) = (a_{ij}) + \left((b_{ij}) + (c_{ij})\right) = A + (B + C)$$

3. $A + 0 = A$
4. $-A + A = 0$

4.2 MNOŽENJE MATRIK S SKALARJEM

Pri množenju matrike s skalarjem α pomnožimo s skalarjem α vse elemente matrike:

$$\alpha A = \begin{bmatrix} \alpha a_{11} & \alpha a_{12} & \cdots & \alpha a_{1n} \\ \alpha a_{21} & \alpha 1_{22} & \cdots & \alpha a_{2n} \\ \vdots & \vdots & & \vdots \\ \alpha a_{m1} & \alpha a_{m2} & \cdots & \alpha a_{mn} \end{bmatrix}$$

Primer 4.2: Množenje matrike s skalarjem

$$-\frac{1}{2} \cdot \begin{bmatrix} 2 & 4 \\ 6 & -8 \end{bmatrix} = \begin{bmatrix} -1 & -2 \\ -3 & 4 \end{bmatrix}$$

Lastnosti množenja matrik s skalarjem:

1. Komutativnost: $\beta A = A\beta$

   Dokaz 4.3: Komutativnost množenja s skalarjem

$$\beta A = \beta(a_{ij}) = (\beta a_{ij}) = (a_{ij}\beta) = A\beta$$

2. Distributivnost: $(\alpha \pm \beta)A = \alpha A + \beta A$ in $\alpha(A \pm B) = \alpha A + \alpha B$

   Dokaz 4.4: Distributivnost množenja s skalarjem

$$(\alpha \pm \beta)A = (\alpha \pm \beta)(a_{ij}) = (\alpha a_{ij} \pm \beta a_{ij}) = \alpha A + \beta A$$

$$\alpha(A \pm B) = \alpha\left((a_{ij}) \pm (b_{ij})\right) = \alpha(a_{ij} + b_{ij}) = (\alpha a_{ij}) + (\alpha b_{ij}) = \alpha A + \alpha B$$

3. Homogenost: $\alpha(\beta A) = \beta(\alpha A) = (\alpha\beta)A$

   Dokaz 4.5: Homogenost množenja s skalarjem

$$\alpha(\beta A) = \alpha\left(\beta(a_{ij})\right) = (\alpha\beta(a_{ij})) = \left(\beta\alpha(a_{ij})\right) = (\alpha\beta)A = \beta(\alpha A)$$

4. Množenje s skalarjem 1: $1 \cdot A = A$

4.3 MNOŽENJE MATRIK

Če želimo izračunati produkt dveh matrik AB, mora imeti druga matrika B enako število vrstic, kot ima prva A stolpcev. Produkt matrik A in B izračunamo tako, da skalarno pomnožimo vsako vrstico matrike A z vsakim stolpcem matrike B. To pomeni, da izračunamo vsoto produktov istoležnih elementov vrstice matrike A in stolpca matrike B. Tako produkt matrike A dimenzije $m \times n$ in matrike B dimenzije $n \times p$ definiramo kot:

$$A \cdot B = AB = \begin{bmatrix} \sum_{k=1}^{n} a_{1k} b_{k1} & \sum_{k=1}^{n} a_{1k} b_{k2} & \cdots & \sum_{k=1}^{n} a_{1k} b_{kp} \\ \sum_{k=1}^{n} a_{2k} b_{k1} & \sum_{k=1}^{n} a_{2k} b_{k2} & \cdots & \sum_{k=1}^{n} a_{2k} b_{kp} \\ \vdots & \vdots & & \vdots \\ \sum_{k=1}^{n} a_{mk} b_{k1} & \sum_{k=1}^{n} a_{mk} b_{k2} & \cdots & \sum_{k=1}^{n} a_{mk} b_{kp} \end{bmatrix}$$

Produkt $A \cdot B$ je dimenzije $m \times p$.

Primer 4.3: Množenje matrik

a) $\begin{bmatrix} 2 & 1 \\ 7 & -3 \end{bmatrix} \cdot \begin{bmatrix} 4 & 5 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 2 \cdot 4 + 1 \cdot (-1) & 2 \cdot 5 + 1 \cdot 2 \\ 7 \cdot 4 + (-3) \cdot (-1) & 7 \cdot 5 + (-3) \cdot 2 \end{bmatrix} = \begin{bmatrix} 7 & 12 \\ 31 & 29 \end{bmatrix}$

b) $\begin{bmatrix} 2 & 3 \\ -1 & 4 \\ 3 & -5 \end{bmatrix} \cdot \begin{bmatrix} 7 & 2 & 4 \\ 1 & 0 & -3 \end{bmatrix} =$

$$\begin{bmatrix} 2 \cdot 7 + 3 \cdot 1 & 2 \cdot 2 + 3 \cdot 0 & 2 \cdot 4 + 3 \cdot (-3) \\ (-1) \cdot 7 + 4 \cdot 1 & (-1) \cdot 2 + 4 \cdot 0 & (-1) \cdot 4 + 4 \cdot (-3) \\ 3 \cdot 7 + (-5) \cdot 1 & 3 \cdot 2 + (-5) \cdot 0 & 3 \cdot 4 + (-5) \cdot (-3) \end{bmatrix} = \begin{bmatrix} 17 & 4 & -1 \\ -3 & -2 & -16 \\ 16 & 6 & 27 \end{bmatrix}$$

Lastnosti množenja matrik:

1. Produkt **ni komutativen**: $AB \neq BA$

Dokaz 4.6: Nekomutativnost množenja matrik - dokaz s protiprimerom

$$AB = \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 7 & 1 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 11 & 2 \\ 33 & 3 \end{bmatrix}$$

$$BA = \begin{bmatrix} 7 & 1 \\ 3 & 0 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 17 & -3 \\ 6 & -3 \end{bmatrix}$$

$$AB = \begin{bmatrix} 11 & 2 \\ 33 & 3 \end{bmatrix} \neq \begin{bmatrix} 17 & -3 \\ 6 & -3 \end{bmatrix} = BA$$

2. Asociativnost: $(AB)C = A(BC)$

Dokaz 4.7: Asociativnost množenja matrik

$$(AB)C = \sum_{l=1}^{p} \left(\sum_{k=1}^{n} a_{ik} b_{kl} \right) c_{lj} = \sum_{l=1}^{p} \sum_{k=1}^{n} a_{ik} b_{kl} c_{lj}$$

$$A(BC) = \sum_{k=1}^{n} a_{ik} \left(\sum_{l=1}^{p} b_{kl} c_{lj} \right) = \sum_{k=1}^{n} a_{ik} \sum_{l=1}^{p} b_{kl} c_{lj}$$

$$(AB)C = \sum_{l=1}^{p} \sum_{k=1}^{n} a_{ik} b_{kl} c_{lj} = \sum_{k=1}^{n} a_{ik} \sum_{l=1}^{p} b_{kl} c_{lj} = A(BC)$$

3. Množenje matrike A z enotsko matriko I: $AI = IA = A$
4. Distributivnost: $(A + B)C = AC + BC$ in $A(B + C) = AB + AC$
5. $\alpha(AB) = (\alpha A)B = A(\alpha B)$

4.4 TRANSPONIRANJE MATRIK

Zamenjavo istoležnih stolpcev in vrstic, ki smo jo omenili pri simetralni matriki, imenujemo transponiranje. Transponirano matriko matrike A dimenzije $m \times n$ označimo z A^T, ki je dimenzije $n \times m$.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

Primer 4.4: Transponirana matrika

Naj bo $A = \begin{bmatrix} 2 & 1 \\ -9 & 7 \\ 3 & 4 \end{bmatrix}$, potem je $A^T = \begin{bmatrix} 2 & -9 & 3 \\ 1 & 7 & 4 \end{bmatrix}$.

Lastnosti transponiranja:

1. $(A^T)^T = A$

Dokaz 4.8: Zaporedno transponiranje matrike

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix},\ Q = A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

$$(A^T)^T = (Q)^T = Q^T = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

2. $(\alpha A)^T = \alpha A^T$

Dokaz 4.9: Transponiranje in množenje s skalarjem

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

$$(\alpha A)^T = \left(\begin{bmatrix} \alpha a_{11} & \alpha a_{12} & \cdots & \alpha a_{1n} \\ \alpha a_{21} & \alpha a_{22} & \cdots & \alpha a_{2n} \\ \vdots & \vdots & & \vdots \\ \alpha a_{m1} & \alpha a_{m2} & \cdots & \alpha a_{mn} \end{bmatrix}\right)^T = \begin{bmatrix} \alpha a_{11} & \alpha a_{21} & \cdots & \alpha a_{m1} \\ \alpha a_{12} & \alpha a_{22} & \cdots & \alpha a_{m2} \\ \vdots & \vdots & & \vdots \\ \alpha a_{1n} & \alpha a_{2n} & \cdots & \alpha a_{mn} \end{bmatrix}$$

$$\alpha A^T = \alpha \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} \alpha a_{11} & \alpha a_{21} & \cdots & \alpha a_{m1} \\ \alpha a_{12} & \alpha a_{22} & \cdots & \alpha a_{m2} \\ \vdots & \vdots & & \vdots \\ \alpha a_{1n} & \alpha a_{2n} & \cdots & \alpha a_{mn} \end{bmatrix}$$

3. $(A+B)^T = A^T + B^T$

Dokaz 4.10: Transponiranje vsote matrik

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

$$(A+B)^T = \left(\begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \cdots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22} & \cdots & a_{2n}+b_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & \cdots & a_{mn}+b_{mn} \end{bmatrix}\right)^T =$$

$$\begin{bmatrix} a_{11}+b_{11} & a_{21}+b_{21} & \cdots & a_{m1}+b_{m1} \\ a_{12}+b_{12} & a_{22}+b_{22} & \cdots & a_{m2}+b_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n}+b_{1n} & a_{2n}+b_{2n} & \cdots & a_{mn}+b_{mn} \end{bmatrix}$$

$$A^T + B^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{21} & \cdots & b_{m1} \\ b_{12} & b_{22} & \cdots & b_{m2} \\ \vdots & \vdots & & \vdots \\ b_{1n} & b_{2n} & \cdots & b_{mn} \end{bmatrix} =$$

$$\begin{bmatrix} a_{11}+b_{11} & a_{21}+b_{21} & \cdots & a_{m1}+b_{m1} \\ a_{12}+b_{12} & a_{22}+b_{22} & \cdots & a_{m2}+b_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n}+b_{1n} & a_{2n}+b_{2n} & \cdots & a_{mn}+b_{mn} \end{bmatrix}$$

4. $(AB)^T = B^T A^T$

Dokaz 4.11: Transponiranje produkta matrik

Če zmnožimo matriko A in B, je (i, j)-ti element produkta enak $\sum_{k=1}^{n} a_{ik} b_{kj}$, ko pa produkt transponiramo, je na istem mestu element $\sum_{k=1}^{n} a_{jk} b_{ki}$. Na drugi strani je pri $B^T A^T$ na (i, j)-tem mestu element enak skalarnemu produktu i-te vrstice B^T z j-tim stolpcem A^T oziroma je enak skalarnemu produktu i-tega stolpca matrike B z j-to vrstico matrike A. (i, j)-ti element $B^T A^T$ je tako:

$$\sum_{k=1}^{n} b_{ki} a_{jk} = \sum_{k=1}^{n} a_{jk} b_{ki}$$

4.5 INVERZNA MATRIKA

Matrik **ne moremo deliti**, dani matriki pa lahko izračunamo inverz oziroma inverzno matriko, s katero lahko rešimo probleme, kjer bi bilo potrebno deljenje (npr. matrične enačbe). Pogoj za inverz matrike A je, da matrika A pripada množici $\mathbb{R}^{n \times n}$ oziroma je kvadratna in ima determinanto različno od 0 [10]. Matrika A^{-1} je inverz matrike A, če velja:

$$A^{-1}A = AA^{-1} = I$$

Če za matriko obstaja inverz, jo imenujemo *obrnljiva matrika*.

Primer 4.5: Neobrnljiva matrika

Matrika $A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$ nima inverza, ker je $AB = 0$ za vse $B \in \mathbb{R}^{2\times 2}$.

Primer 4.6: Inverz enotske matrike

Enotska matrika *I* = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ *je inverzna sama sebi*, ker je $I \cdot I = I$, zato je $I^{-1} = I$.

Lastnosti inverzne matrike: (dokazi v [10])

1. $(A^{-1})^{-1} = A$
2. $(AB)^{-1} = B^{-1}A^{-1}$
3. $(A^{-1})^T = (A^T)^{-1}$
4. $(A + B)^{-1} \neq A^{-1} + B^{-1}$

4.5.1 ISKANJE INVERZA MATRIKE 2×2

Ker bomo v nadaljevanju raziskovalne naloge za transformacijo ravnine potrebovali le kvadratne matrike reda 2×2, se bomo omejili samo na iskanje inverza takih matrik in ga ne bomo razširjali. Inverz matrike dimenzije 2×2 izračunamo po formuli:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}, \det(A) \neq 0$$

Pravilnost formule lahko hitro dokažemo:

$$A^{-1}A = \frac{1}{\det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} ad - bc & 0 \\ 0 & -bc + ad \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Enak rezultat dobimo tudi če množimo AA^{-1}.

Inverz matrike A dimenzije 2×2 lahko izračunamo, kadar velja $\det(A) = ad - bc \neq 0$.

Primer 4.7: Inverz matrike 2 x 2

Izračunamo inverz matrike A:

$$A = \begin{bmatrix} 2 & 5 \\ -3 & 1 \end{bmatrix}$$

$$A^{-1} = \frac{1}{2 \cdot 1 - 5 \cdot (-3)} \begin{bmatrix} 1 & -5 \\ 3 & 2 \end{bmatrix} = \frac{1}{17} \begin{bmatrix} 1 & -5 \\ 3 & 2 \end{bmatrix}$$

Preverimo, če drži $A^{-1}A = I$:

$$A^{-1}A = \frac{1}{17} \begin{bmatrix} 1 & -5 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 2 & 5 \\ -3 & 1 \end{bmatrix} = \frac{1}{17} \begin{bmatrix} 17 & 0 \\ 0 & 17 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

5 MATRIKE IN LINEARNE TRANSFORMACIJE

5.1 LINEARNE TRANSFORMACIJE RAVNINE

Transformacija je preslikava ravnine nase. Med transformacijami ločimo linearne in nelinearne transformacije. V nadaljevanju bomo spoznali, da lahko transformacije ravnine zelo preprosto predstavimo z matrikami.

Transformacija A ravnine nase je linearna, če za vsako realno število α in za vsak par vektorjev $\vec{a}$ in $\vec{b}$ v ravnini velja:

$$A(\alpha\vec{a}) = \alpha(A\vec{a})$$

$$A\left(\vec{a} + \vec{b}\right) = A\vec{a} + A\vec{b}$$

Transformacijo lahko opišemo s spremembo baznih vektorjev $\vec{\imath}$ in $\vec{\jmath}$. Če poznamo njuni končni legi, lahko določimo, za katero transformacijo gre, oziroma s katero matriko smo ju pomnožili. Ker so vsi ostali vektorji v ravnini linearne kombinacije baznih vektorjev, se transformirajo enako.

Transformacijo v ravnini lahko zapišemo z matriko dimenzije 2×2, v kateri prvi stolpec predstavlja komponenti baznega vektorja $\vec{\imath}$ po transformaciji, drugi stolpec pa komponenti baznega vektorja $\vec{\jmath}$ po transformaciji. Obratno lahko razberemo, kako matrika transformira bazna vektorja in s tem tudi koordinatni sistem ter vse vektorje v njem.

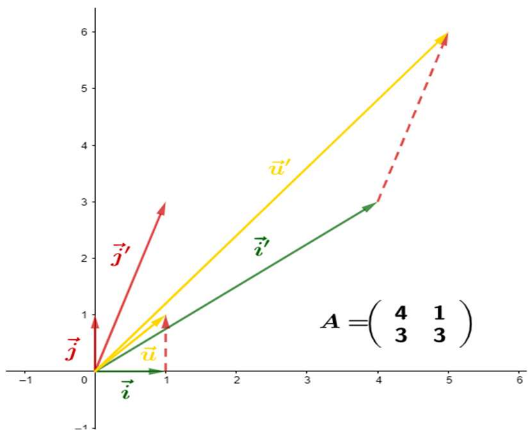
Primer 5.1: Linearna transformacija

Bazna vektorja pred transformacijo:

$$\vec{\imath} = (1,0), \qquad \vec{\jmath} = (0,1)$$

Bazna vektorja po transformaciji:

$$\vec{\imath}' = (4,3), \qquad \vec{\jmath}' = (1,3)$$



Slika 1: Linearna transformacija

Matriko, s katero smo transformirali, lahko tako zapišemo kot:

$$A = \begin{bmatrix} 4 & 1 \\ 3 & 3 \end{bmatrix}$$

Vektor $\vec{u}$ pred transformacijo:

$$\vec{u} = (1,1)$$

Vektor $\vec{u}$ po transformaciji:

$$\vec{u}' = A\vec{u} = \begin{bmatrix} 4 & 1 \\ 3 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

5.2 OSNOVNE LINEARNE TRANSFORMACIJE RAVNINE

V nadaljevanju je opisanih nekaj osnovnih linearnih transformacij ravnine, ki jih lahko v splošnem zapišemo z matrikami dimenzij 2×2. Pomembno je, da vektorje definiramo v stolpcu $\mathbb{R}^{n\times 1}$ in ustrezno preslikavo predstavimo kot produkt matrike in stolpca.

5.2.1 Razteg v smeri osi x

Pri raztegu vektorja (a, b) v smeri osi x s skalarjem α se prva komponenta vektorja pomnoži z α, druga pa ohrani.

Matrika raztega v smeri osi x:

$$A = \begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix}, \qquad \alpha \neq 0, 1$$

Razteg vektorja (a, b):

$$\begin{bmatrix} \alpha & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \alpha a \\ b \end{bmatrix}$$

Grafični prikaz:

Slika 2: Razteg v smeri osi x

5.2.2 Razteg v smeri osi y

Pri raztegu vektorja (a, b) v smeri osi y s skalarjem α se druga komponenta vektorja pomnoži z α, prva pa ohrani.

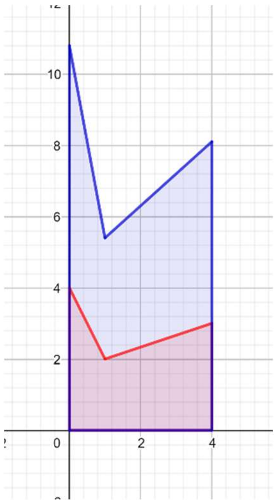
Matrika raztega v smeri osi y:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix}, \qquad \alpha \neq 0, 1$$

Razteg vektorja (a, b):

$$\begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a \\ \alpha b \end{bmatrix}$$

Grafični prikaz:



Slika 3: Ratzeg v smeri osi y

5.2.3 Strig v smeri osi x

Strižna transformacija oziroma strig je transformacija, pri kateri vse točke na dani premici ostanejo na mestu, ostale točke pa se pomaknjeno vzporedno s premico za razdaljo, ki je sorazmerna z oddaljenostjo točke od premice.

Pri strigu vektorja (a, b) s faktorjem α v smeri osi x se druga komponenta vektorja ohrani, prva pa je linearna kombinacija $a + \alpha b$.

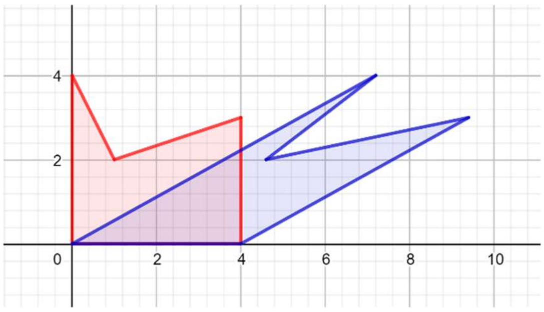
Matrika striga v smeri osi x:

$$A = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix}, \alpha \neq 0$$

Strig vektorja (a, b):

$$\begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a + \alpha b \\ b \end{bmatrix}$$

Grafični prikaz:



Slika 4: Strig v smeri osi x

5.2.4 Strig v smeri osi y

Pri strigu vektorja (a, b) s faktorjem α v smeri osi y se prva komponenta vektorja ohrani, druga pa je linearna kombinacija $\alpha a + b$.

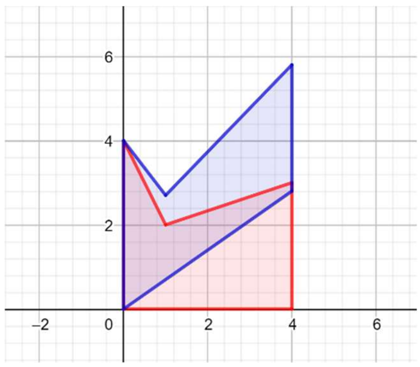
Matrika striga v smeri osi y:

$$A = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix}, \alpha \neq 0$$

Strig vektorja (a, b):

$$\begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a \\ \alpha a + b \end{bmatrix}$$

Grafični prikaz:



Slika 5: Strig v smeri osi y

5.2.5 Zrcaljenje čez os x

Pri zrcaljenju vektorja (a, b) čez os x se spremeni predznak drugi komponenti vektorja.

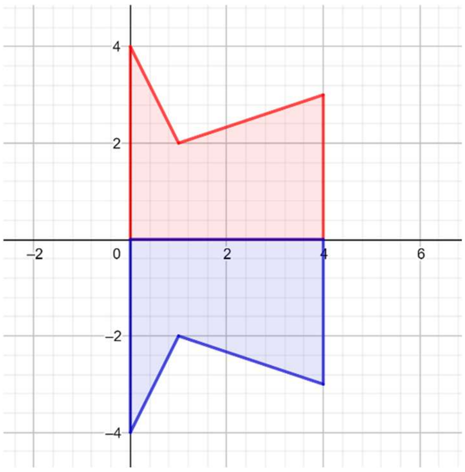
Matrika zrcaljenja čez os x:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Zrcaljenje vektorja (a, b):

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a \\ -b \end{bmatrix}$$

Grafični prikaz:



Slika 6: Zrcaljenje čez x os

5.2.6 Zrcaljenje čez os y

Pri zrcaljenju vektorja (a, b) čez os y se spremeni predznak prvi komponenti vektorja.

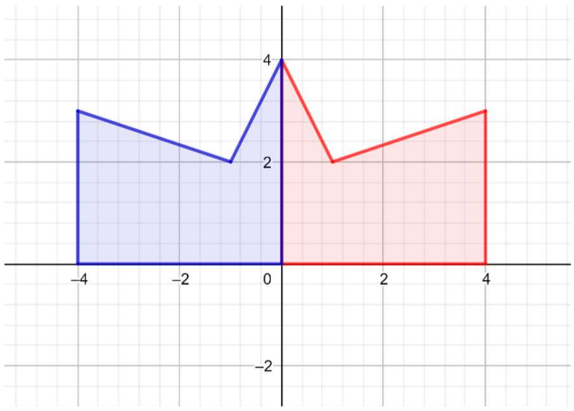
Matrika zrcaljenja čez os y:

$$A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Zrcaljenje vektorja (a, b):

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -a \\ b \end{bmatrix}$$

Grafični prikaz:



Slika 7: Zrcaljenje čez y os

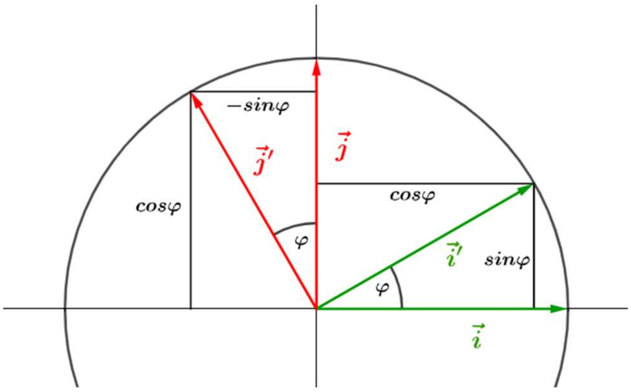
5.2.7 Vrtež okrog izhodišča

Matriko, ki povzroči vrtež, lahko prav tako določimo s pomočjo opazovanja spremembe položaja baznih vektorjev. Če ju zavrtimo za izbrani kot φ, lahko s pomočjo kotnih funkcij zapišemo njuni končni legi.

Ker sta $\vec{\imath}$ in $\vec{\jmath}$ med seboj pravokotna, sta kota, ki ju po transformaciji oklepata z osema, enaka. Ker imata vektorja na začetku dolžino 1, lahko obe komponenti njune končne lege zapišemo s kotnima funkcijama $sin\varphi$ in $cos\varphi$. Na sliki 8 sta z $\vec{\imath}$ in $\vec{\jmath}$ označena bazna vektorja v začetnem položaju, z $\vec{\imath}'$ in $\vec{\jmath}'$ pa bazna vektorja po transformaciji. Slednja sta nato razstavljena na x in y komponento, ki ju lahko zapišemo s kotnima funkcijama.

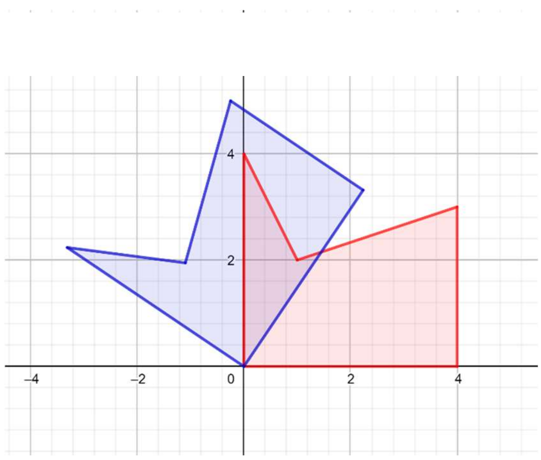
Matrika vrteža okrog koordinatnega izhodišča:

$$A = \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix}$$



Slika 8: Vrtež okrog izhodišča - izpeljava

Grafični prikaz:

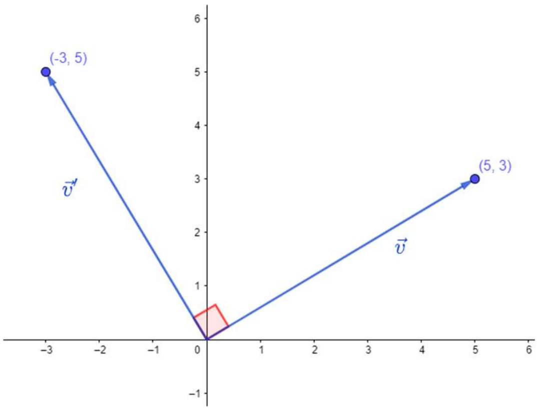


Slika 9:Vrtež okrog izhodišča

Primer 5.2: Vrtež za $90°$

$$\vec{v} = (a, b), A = \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix}, \varphi = 90°$$

$$\vec{v}' = A\vec{v} = \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} -b \\ a \end{bmatrix}$$



Slika 10: Vrtež za 90°

5.3 TRANSFORMACIJA Z INVERZNO MATRIKO

Pri transformiranju ravnine z matrikami je zanimivo vprašanje tudi, kako bi nanjo vplival inverz izbrane matrike, katere determinanta je različna od 0. Če primerjamo transformacijo, ki jo povzroči matrika A, je transformacija z matriko A^{-1} inverzna. Če torej izbrano ravnino najprej pomnožimo oziroma transformiramo z matriko A, nato pa še z matriko A^{-1}, bomo dobili izhodiščno sliko. Enako velja tudi v obratni smeri.

Opisana lastnost izhaja iz osnovne definicije inverza matrike, za katerega vedno velja:

$$A^{-1}A = AA^{-1} = I$$

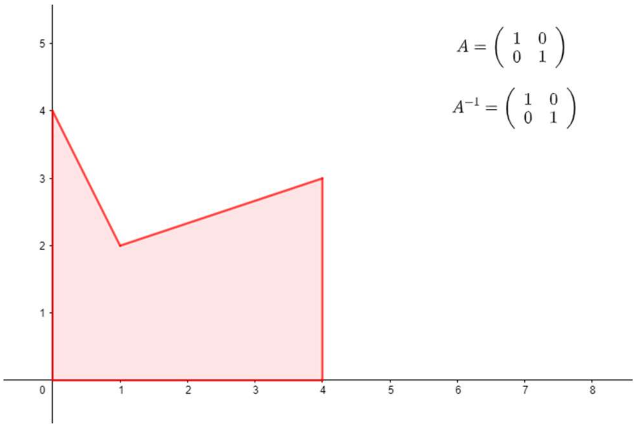
Ker pri množenju z enotsko matriko ni spremembe, je trditev, da je transformacija z matriko A^{-1} inverzna, pravilna.

Transformacijo z inverzno matriko si lahko ogledate tudi na animaciji v GeoGebri:

https://www.geogebra.org/m/z2gtrvb3

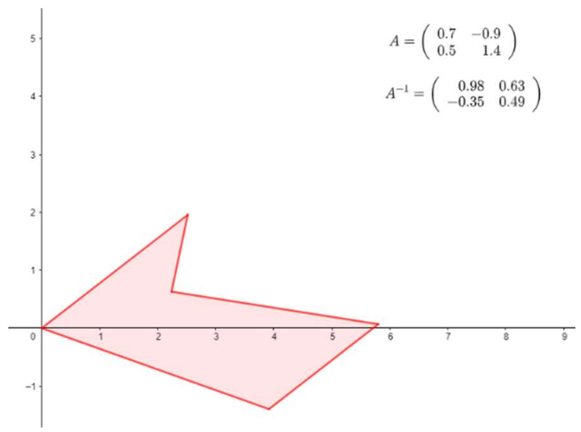
Primer 5.3: Transformacija z inverzno matriko

Na začetku imamo spodnji rdeči lik.



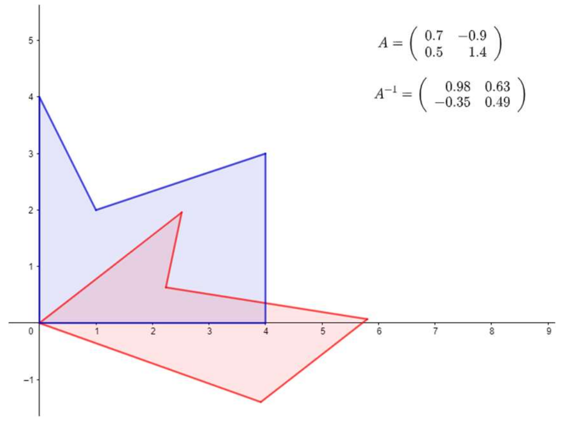
Slika 11: Vpliv inverzne matrike 1

Lik transformiramo z matriko $A = \begin{bmatrix} 0.7 & -0.9 \\ 0.5 & 1.4 \end{bmatrix}$



Slika 13: Vpliv inverzne matrike 2

Novo dobljeni rdeči lik nato pomnožimo z inverzom matrike A in ponovno dobimo začetno sliko (moder lik).



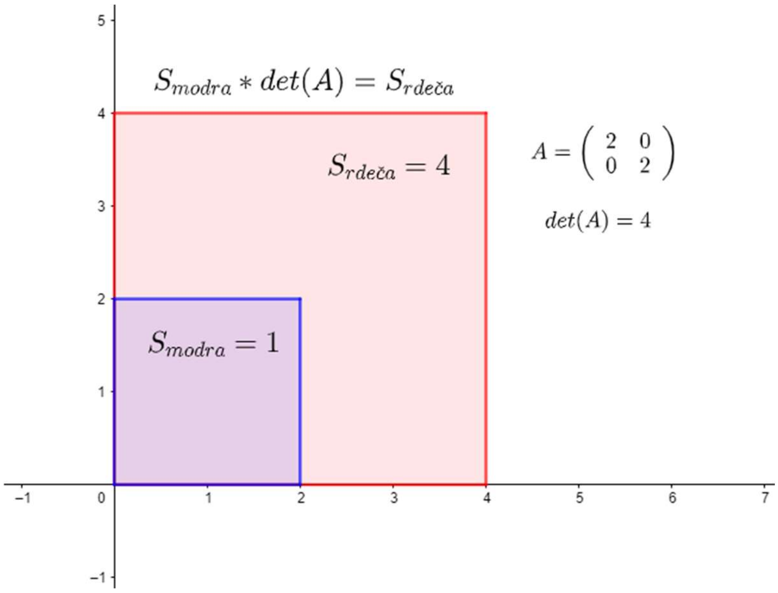
Slika 12: Vpliv inverzne matrike 2

5.4 VPLIV DETERMINANTE MATRIKE NA TRANSFORMACIJO

Pri linearnih transformacijah z matrikami je zanimiv tudi geometrični vpliv determinante matrike. Slednjo izračunamo po definiciji determinante:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \qquad \det(A) = ad - bc$$

Absolutna vrednost determinante matrike A nam pove faktor, za katerega se poveča oziroma zmanjša ploščina kateregakoli lika v ravnini po transformaciji z matriko. Če je vrednost determinante negativna, se ravnina pri transformaciji obrne.



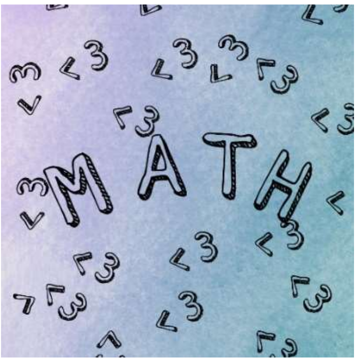
Slika 14: Vpliv determinante

Determinanta matrike ima lahko tudi vrednost 0. V tem primeru pride do dimenzionalne skrčitve ravnine v premico ali točko.

6 UPORABA MATRIK

6.1 GEOGEBRA

Za grafično predstavo transformacij ravnine sva v GeoGebri izdelala interaktivno animacijo *Transformacija slike z matrikami*. Osnova je *slika 15*, njena oglišča pa so definirana s tremi krajevnimi vektorji. Le-ti so produkt matrike in drugih treh krajevnih vektorjev, ki so enaki prvim v izhodiščnem položaju. Ker so vsi štirje elementi matrike spremenljivke določene z drsniki, lahko matriko spreminjamo, kot želimo in vidimo, kako



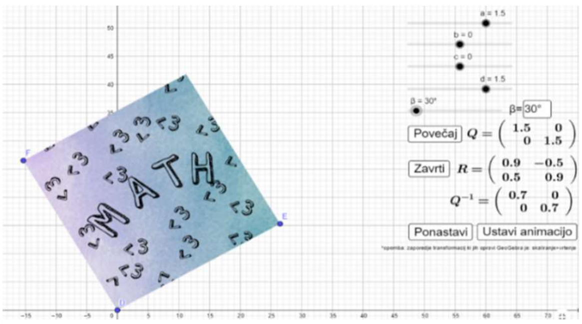
Slika 15: Math

le-to vpliva na sliko.

Posebej sta omogočena povečava ter vrtež, ročno pa lahko poustvarimo tudi ostale osnovne transformacije omenjene v poglavju o linearnih transformacijah. Ločeno sva naredila še animacijo z naslovom *Vpliv inverzne matrike na sliko*, ki prikazuje vpliv inverzne matrike.

Povezava do animacije *Transformacija slike z matrikami*:

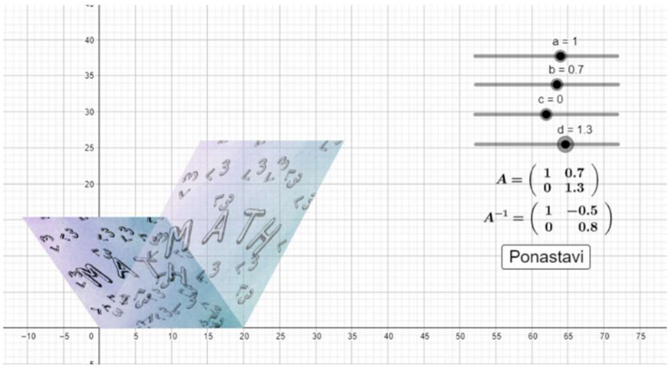
https://www.geogebra.org/m/hap7kqhx



Slika 16: Transformacija slike z matrikami

Povezava do animacije *Vpliv inverzne matrike na sliko*:

https://www.geogebra.org/m/z2gtrvb3



Slika 17: Vpliv inverzne matrike na sliko

6.1.1 Uporaba matrik v GeoGebri

Pri izdelavi animacij sva se naučila uporabljati matrike v GeoGebri in program nasploh. Ko spoznamo osnovno delovanje programa oziroma njegov uporabo, je le-ta precej intuitiven. Lahko ga uporabljamo za učenje in raziskovanje novih matematičnih pojmov in lastnosti ter za risanje in predstavitev ugotovitev. Delo z matrikami je v osnovi enostavno. Potrebno je znanje nekaterih ukazov ter natančnost, saj kmalu nastopi veliko različnih oznak, ki jih lahko med seboj pomešamo in tako ne dobimo željenega rezultata.

Osnovni ukazi za delo z matrikami:

1. Vnos matrike $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$:

   $A = \{\{a, b\}, \{c, d\}\}$

2. Transponiranje matrike $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$:

   $Transpose(A)$

3. Inverz matrike $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$:

   A^{-1}

4. Produkt matrike $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ in matrike $B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$:

   $A \cdot B$

5. Produkt matrike $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ in stolpčnega vektorja $\hat{v} = \begin{bmatrix} e \\ f \end{bmatrix}$:

   $A \cdot \hat{v}$

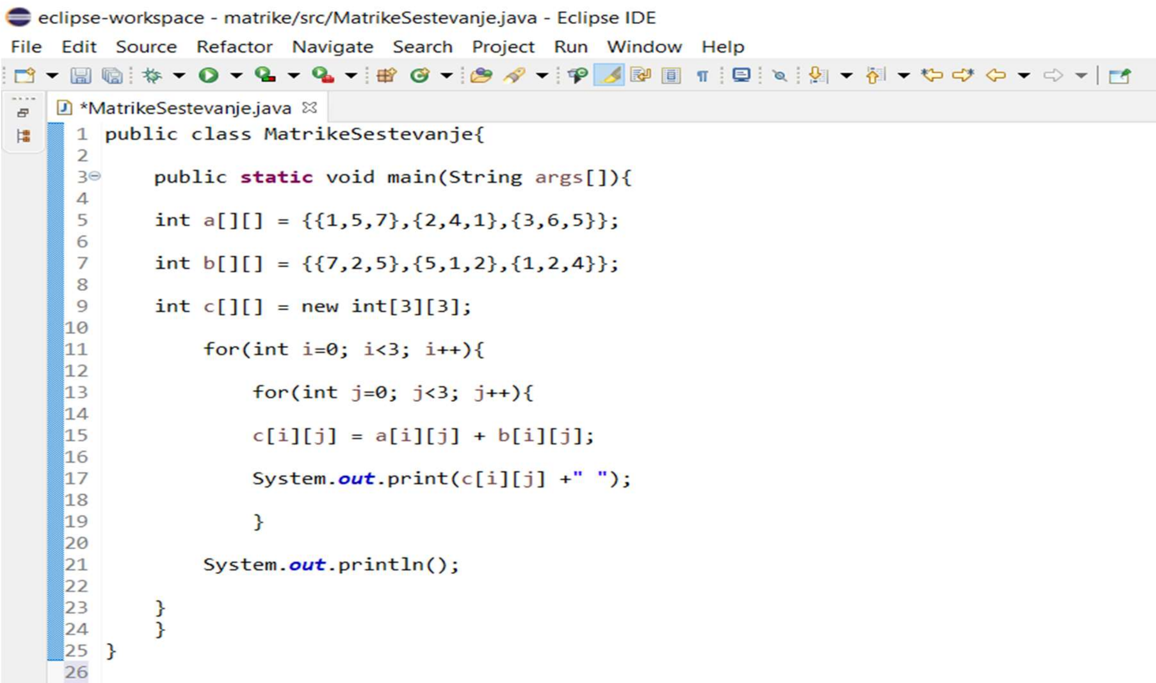
Zelo priročno v GeoGebri je tudi to, da točke avtomatsko zapiše kot stolpične krajevne vektorje, zato jih pred množenjem z matriko ni potrebno spreminjati.

6.2 JAVA

V programu tehniške gimnazije imamo strokovni predmet računalništvo, zato sva se odločila, da uporabo matrik predstaviva tudi v programskem jeziku Java. V nadaljevanju sledijo opisani programi osnovnih algebraičnih operacij na matrikah.

6.2.1 Seštevanje in odštevanje matrik

Sama operacija seštevanja oz. odštevanja matrik je enostavna, saj je potrebno le sešteti oz. odšteti istoležne elemente matrike, pri čemer pa morata biti matriki enakih dimenzij. Takole v programskem okolju Eclipse izgleda program za seštevanje dveh matrik dimenzij 3×3.



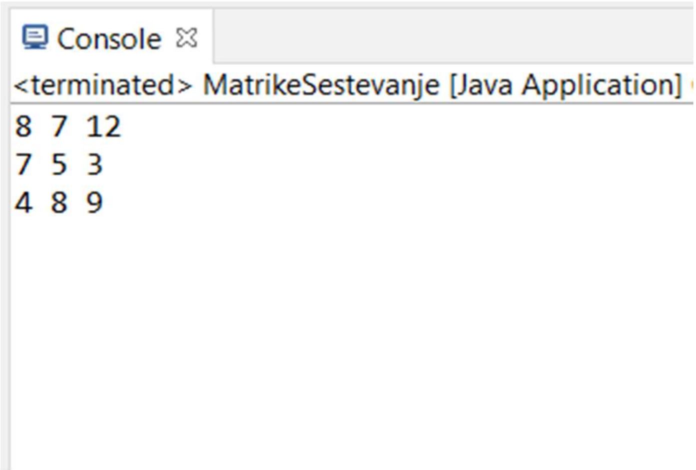
Slika 18: Seštevanje matrik v Javi

V programu najprej določimo dve matriki, ki sta označeni a in b. Matriko določimo tako, da v posamezen zaviti oklepaj zapišemo vrstico izbrane matrike in nato v naslednji zaviti oklepaj naslednjo vrstico matrike. Enako ponovimo še za drugo matriko, v našem primeru za števec b.

Nato določimo še tretjo matriko oz. števec c, ki nam bo služil za zapis nove matrike, ki bo seštevek matrik a in b. Ta matrika bo imela tri vrstice in tri stolpce, kar je logično, saj seštevamo dve matriki, s prav tako po tremi vrsticami in stolpci.

Sedaj je potrebno le še sešteti obe matriki. To storimo tako, da zapišemo dve ˝for˝ zanki. Ena zanka sešteva vrstice matrik in druga stolpce. Zanki potekata od prvega elementa vrstice oziroma stolpca do tretjega elementa s pomikom za ena, kar je določeno v zanki.

Na koncu samo še izpišemo dobljeno matriko, to storimo s funkcijo ˝System.out.print˝, ki nam izpiše matriko c. Dobimo sledeči izpis.



Slika 19: Izpis oz. rezultat seštevanja matrik

Dobili smo seštevek dveh matrik dimenzij 3×3, kar lahko tudi sami hitro in enostavno preverimo.

Če pa bi želeli, da program matriki odšteje, bi morali v 15. vrstici kode samo spremeniti znak $+$ v znak $-$.

6.2.2 Množenje matrik

Operacija množenja matrik je v praksi veliko bolj zapletena od seštevanja in odštevanja. To pa ne velja za množenje matrik s pomočjo programa. Od nas ta operacija zahteva le še eno dodatno zanko.

eclipse-workspace - matrike/src/MatrikeMnozenje.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

*MatrikeMnozenje.java Console

```java
public class MatrikeMnozenje{

    public static void main(String args[]){

        int a[][]={{3,2,4},{4,1,0},{1,3,0}};

        int b[][]={{3,1,2},{5,2,1},{5,3,1}};

        int c[][]=new int[3][3];

        for(int i=0;i<3;i++){

            for(int j=0;j<3;j++){

                c[i][j]=0;

                for(int k=0;k<3;k++)
                    {
                    c[i][j]+=a[i][k]*b[k][j];
                    }
                    System.out.print(c[i][j]+" ");
                }
                System.out.println();
        }
    }
}
```

Slika 20: Množenje matrik v Javi

Prav tako kot pri seštevanju in odštevanju matrik, najprej določimo dve matriki, ki bi ju radi množili, in matriko, v katero bomo zapisali rezultat.

Za razliko od predhodnih operacij pri množenju ne množimo istoležnih elementov matrike, zato bomo potrebovali tri ˝for˝ zanke, saj množimo vrstice prve matrike s stolpci druge matrike.

V programu po pravilih za množenje matrik izračunamo še matriko c in jo izpišemo.

```
Console
<terminated> MatrikeMnozenje [Java Application]
39 19 12
17 6 9
18 7 5
```

Slika 21: Izpis oz. rezultat množenja matrik

Dobili smo novo matriko c, ki je produkt matrik a in b.

6.2.3 Transponiranje matrike

Transponirati matriko, kot že vemo, pomeni, da ji zamenjamo vrstice in stolpce. Vrstice matrike postanejo stolpci nove matrike in stolpci postanejo vrstice nove matrike. Spodaj lahko vidimo program, ki transponira matriko dimenzije 3×3.

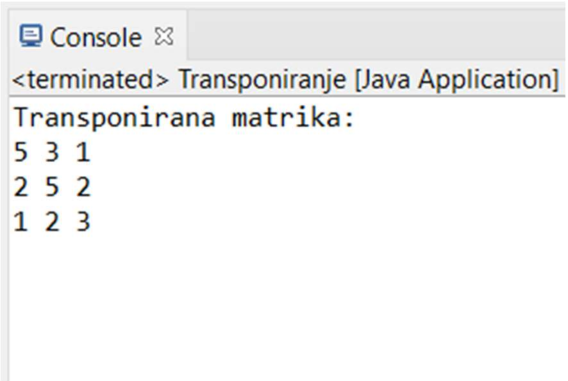
```
eclipse-workspace - matrike/src/Transponiranje.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Transponiranje.java
public class Transponiranje
{
    public static void main(String[] args) {
        int rows, cols;
          int a[][] = {
                          {5, 2, 1},
                          {3, 5, 2},
                          {1, 2, 3}    };
        rows = a.length;
        cols = a[0].length;

        int t[][] = new int[cols][rows];
        for(int i = 0; i < cols; i++){
            for(int j = 0; j < rows; j++){
                t[i][j] = a[j][i];
            }
        }
        System.out.println("Transponirana matrika:");

        for(int i = 0; i < cols; i++){
            for(int j = 0; j < rows; j++){
               System.out.print(t[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Slika 22: Transponiranje matrike v Javi

Na začetku programa najprej določimo matriko, ki jo bomo transponirali, le da tokrat uporabimo drugačen način zapisa, ki bere celotne vrstice in stolpce, kar nam bo prišlo še kako prav, saj jih bomo morali le zamenjati med seboj.

Ključna vrstica programske kode je 15. vrstica, v kateri v novo spremenljivko t zapišemo vrstice in stolpce matrike, a v obratnem vrstnem redu; na ta način smo jih med seboj zamenjali.

Na koncu samo še izpišemo novo matriko, ki jo program poimenuje *Transponirana matrika.*



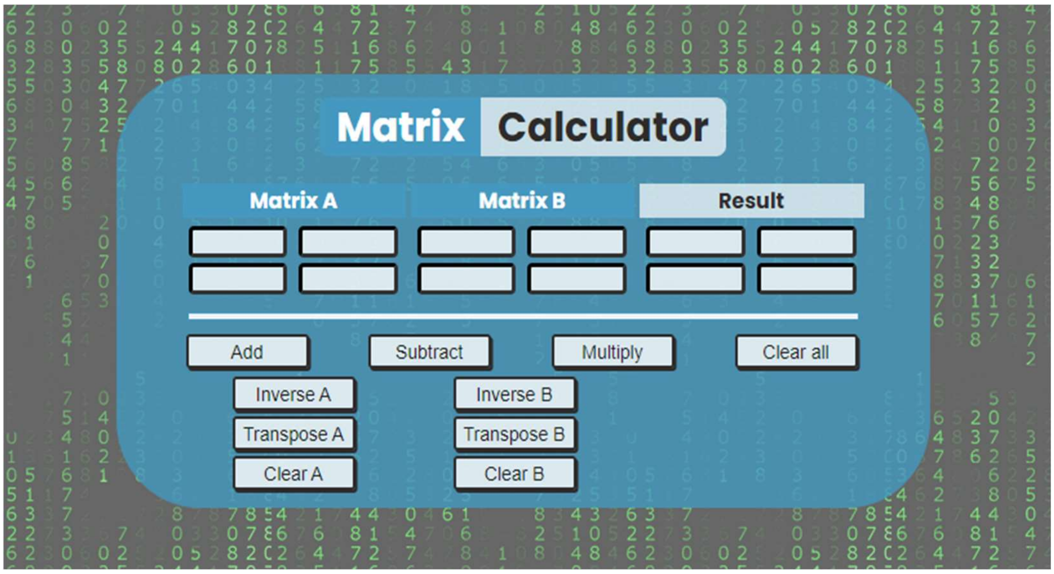
Slika 23: Izpis transponirane matrike

Tudi pri tej operaciji lahko sami hitro preverimo pravilnost delovanja programa, saj takoj razberemo, da so stolpci nove matrike enaki vrsticam predhodne matrike in obratno.

6.3 MATRIČNI KALKULATOR

Ker je lahko računanje z matrikami zamudno, sva se odločila izdelati matrični kalkulator[1]. Računski del je zapisan v programskem jeziku JavaScript, ostalo pa v HTML-ju ter css-u[2]. Kalkulator lahko računa matrike dimenzij 2×2 in zmore vse opisane računske operacije.

Povezava do kalkulatorja: https://nejchocevar8.wixsite.com/matrix-calculator



Slika 24: Matrični kalkulator

[1] Pri računanju inverza kalkulator zaradi zaokroževanja ne povrne vedno začetnih vrednosti, če matriko večkrat obrnemo.

[2] Celotno kodo kalkulatorja si lahko ogledate v *Prilogi 1*.

7 ZAKLJUČEK

Med raziskovanjem in izdelavo raziskovalne naloge naju je ves časa vodila želja po pridobivanju novih spoznanj. Spoznala sva novo področje matematike, ki nama je zelo všeč in ga bi v nadaljnje rada še podrobneje spoznala. Prav tako sva se naučila veliko novega o programskem svetu. Naloga nama je omogočila ter naju spodbudila, da sva pridobila nova znanja in jih praktično prikazala. Hkrati nama je omogočila, da izraziva svojo kreativnost in ustvariva izdelek po svojih željah, v katerega sva seveda vložila veliko truda.

Možnosti za nadaljnje raziskovanje so široko odprte. Matrike so zelo kompleksna tema in se uporabljajo še na mnogih drugih področjih (npr. robotika, grafična obdelava slik). Vesela sva, da sva izbrala to temo in pridobila nova znanja ter izkušnje.

8 VIRI IN LITERATURA

1) 3Blue1Brown. (16.. Marec 2019). *3Blue1Brown*. Pridobljeno iz Essence of linear algebra: https://www.youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab, dne 19. 1. 2021

2) *Doba*. (Januar 2018). Pridobljeno iz Doba: http://www.doba.si/ftp/egradiva/pms-df/index.html, dne 10. 1. 2021

3) *GeeksforGeeks*. (Marec 2017). Pridobljeno iz GeeksforGeeks: https://www.geeksforgeeks.org/matrix/, dne 3. 2. 2021

4) Hrvatin, E. (2017). Vizualizacija polarnega razcepa linearnih transformacij. Ljubljana, Ljubljana, Slovenija: Univerza v LJubljani, Pedagoška fakulteta. Pridobljeno iz http://pefprints.pef.uni-lj.si/4654/1/diploma.pdf, dne 19. 1. 2021

5) *javaTpoint*. (Februar 2021). Pridobljeno iz javaTpoint: https://www.javatpoint.com/array-in-java, dne 3. 2. 2021

6) Košir. (Januar 2014). *fmf.uni-lj.* Pridobljeno iz fmf.uni-lj: https://www.fmf.uni-lj.si/~kosir/poucevanje/skripta/matrike.pdf, dne 10. 1. 2021

7) *Mathbits*. (2020). Pridobljeno iz Mathbits: https://mathbits.com/JavaBitsNotebook/Graphics/GraphicIntro.html, dne 3. 2. 2021

8) *Oracle*. (2020). Pridobljeno iz Oracle: https://docs.oracle.com/javase/tutorial/2d/basic2d/index.html, dne 3. 2. 2021

9) Orel, B. (2017). Linearna algebra. Ljubljana, Ljubljana, Slovenija. Pridobljeno iz http://eprints.fri.uni-lj.si/3857/1/orel2.pdf, dne 21. 2. 2021

10) Pustavrh, S. (2007). *Matematika s statistiko : [učbenik za 1. letnik višje strokovne šole : promet]*. Novo mesto: Šolski center, Višja strokovna šola.

11) Škraba, A. (Januar 2020). *Astra.si*. Pridobljeno iz Astra.si: https://astra.si/algebra/matrike/, dne 10. 1. 2021

12) *w3schools*. (Januar 2021). Pridobljeno iz w3schools: https://www.w3schools.com/, dne 3. 2. 2021

13) *Wikiwand*. (2017). Pridobljeno iz Strižna matrika: https://www.wikiwand.com/sl/Stri%C5%BEna_matrika, dne 13. 2. 2021

14) Željko, M. (30.. maj 2011). *DMFA.* Pridobljeno iz DMFA: http://zeljko.dmfa.si/Lectures/2011/Zapiski-Bio-Teden12.pdf, dne 19. 1. 2021

9 PRILOGE

Priloga 1 (Matrični kaklulator):

```
<!DOCTYPE html>
<html>
<head>
        <title>Matrix calculator</title>
        <script>
                // Rounding function: Math.round(num*100)/100
                function add()
                {
                let a = Number(document.getElementById("a").value)
                let e = Number(document.getElementById("e").value)
                let b = Number(document.getElementById("b").value)
                let f = Number(document.getElementById("f").value)
                let c = Number(document.getElementById("c").value)
                let g = Number(document.getElementById("g").value)
                let d = Number(document.getElementById("d").value)
                let h = Number(document.getElementById("h").value)
                let i = Math.round(eval(a+e)*100)/100
                let j = Math.round(eval(b+f)*100)/100
                let k = Math.round(eval(c+g)*100)/100
                let l = Math.round(eval(d+h)*100)/100

                document.getElementById("i").value = i
                document.getElementById("j").value = j
                document.getElementById("k").value = k
                document.getElementById("l").value = l
                }

                function subtract()
                {
                let a = Number(document.getElementById("a").value)
                let e = Number(document.getElementById("e").value)
                let b = Number(document.getElementById("b").value)
                let f = Number(document.getElementById("f").value)
                let c = Number(document.getElementById("c").value)
```

```
let g = Number(document.getElementById("g").value)
let d = Number(document.getElementById("d").value)
let h = Number(document.getElementById("h").value)
let i = Math.round(eval(a-e)*100)/100
let j = Math.round(eval(b-f)*100)/100
let k = Math.round(eval(c-g)*100)/100
let l = Math.round(eval(d-h)*100)/100

document.getElementById("i").value = i
document.getElementById("j").value = j
document.getElementById("k").value = k
document.getElementById("l").value = l
}
function multiply()
{
let a = Number(document.getElementById("a").value)
let e = Number(document.getElementById("e").value)
let b = Number(document.getElementById("b").value)
let f = Number(document.getElementById("f").value)
let c = Number(document.getElementById("c").value)
let g = Number(document.getElementById("g").value)
let d = Number(document.getElementById("d").value)
let h = Number(document.getElementById("h").value)
let i = Math.round(eval((a*e)+(b*g))*100)/100
let j = Math.round(eval((a*f)+(b*h))*100)/100
let k = Math.round(eval((c*e)+(d*g))*100)/100
let l = Math.round(eval((c*f)+(d*h))*100)/100

document.getElementById("i").value = i
document.getElementById("j").value = j
document.getElementById("k").value = k
document.getElementById("l").value = l
}
function inverseA()
{
let a = Number(document.getElementById("a").value)
let e = Number(document.getElementById("e").value)
let b = Number(document.getElementById("b").value)
```

```
let f = Number(document.getElementById("f").value)
let c = Number(document.getElementById("c").value)
let g = Number(document.getElementById("g").value)
let d = Number(document.getElementById("d").value)
let h = Number(document.getElementById("h").value)
let n = Math.round(eval((1/(a*d-b*c))*d)*100)/100
let x = Math.round(eval((1/(a*d-b*c))*(-b))*100)/100
let y = Math.round(eval((1/(a*d-b*c))*(-c))*100)/100
let z = Math.round(eval((1/(a*d-b*c))*a)*100)/100

document.getElementById("a").value = n
document.getElementById("b").value = x
document.getElementById("c").value = y
document.getElementById("d").value = z
}
function inverseB()
{
let a = Number(document.getElementById("a").value)
let e = Number(document.getElementById("e").value)
let b = Number(document.getElementById("b").value)
let f = Number(document.getElementById("f").value)
let c = Number(document.getElementById("c").value)
let g = Number(document.getElementById("g").value)
let d = Number(document.getElementById("d").value)
let h = Number(document.getElementById("h").value)
let n = Math.round(eval((1/(e*h-f*g))*h)*100)/100
let x = Math.round(eval((1/(e*h-f*g))*(-f))*100)/100
let y = Math.round(eval((1/(e*h-f*g))*(-g))*100)/100
let z = Math.round(eval((1/(e*h-f*g))*e)*100)/100

document.getElementById("e").value = n
document.getElementById("f").value = x
document.getElementById("g").value = y
document.getElementById("h").value = z
}
function transposeA()
{
let a = Number(document.getElementById("a").value)
```

```
let b = Number(document.getElementById("b").value)
let c = Number(document.getElementById("c").value)
let d = Number(document.getElementById("d").value)
let n = a
let x = c
let y = b
let z = d
document.getElementById("a").value = n
document.getElementById("b").value = x
document.getElementById("c").value = y
document.getElementById("d").value = z
}
function transposeB()
{
let e = Number(document.getElementById("e").value)
let f = Number(document.getElementById("f").value)
let g = Number(document.getElementById("g").value)
let h = Number(document.getElementById("h").value)
let n = e
let x = g
let y = f
let z = h

document.getElementById("e").value = n
document.getElementById("f").value = x
document.getElementById("g").value = y
document.getElementById("h").value = z
}
function clearA()
{
document.getElementById("a").value = ""
document.getElementById("b").value = ""
document.getElementById("c").value = ""
document.getElementById("d").value = ""
}
function clearB()
{
document.getElementById("e").value = ""
```

```
                document.getElementById("f").value = ""
                document.getElementById("g").value = ""
                document.getElementById("h").value = ""
                }
                function clearall()
                {
                document.getElementById("a").value = ""
                document.getElementById("b").value = ""
                document.getElementById("c").value = ""
                document.getElementById("d").value = ""
                document.getElementById("e").value = ""
                document.getElementById("f").value = ""
                document.getElementById("g").value = ""
                document.getElementById("h").value = ""
                document.getElementById("i").value = ""
                document.getElementById("j").value = ""
                document.getElementById("k").value = ""
                document.getElementById("l").value = ""
                }
        </script>
    <style>
    @import url("https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap");
:root {
  --Calc-color: rgb(46, 44, 44);
  --matrixBG-color: rgb(69, 153, 192);
}
body {
  width: 100%;
  min-height: 100%;
  margin: 0;
  background: #ffffff;
  position: relative;
  height: auto;
}
body::before {
  content: '';
  position: fixed;
  top: 0;
```

```
  bottom: 0;
  left: 0;
  right: 0;
  background-image: url("https://media.giphy.com/media/WoD6JZnwap6s8/giphy.gif");
  opacity: 0.6;
  /* background-size: cover; */
  z-index: -1;
}
table {
  width: 100%;
}
.boxes {
  text-align: center;
}
section.calculator {
  max-width: 630px;
  max-height: 317px;
  color: white;
  background: rgba(69, 153, 192, 0.8);
  margin: 100px auto 50px auto;
  padding: 1em 3em 2em 3em;
  border-radius: 4em;
  position: relative;
  -webkit-box-sizing: border-box;
          box-sizing: border-box;
}
section.calculator .heading {
  color: white;
  margin-bottom: 0.5em;
  font-weight: bold;
  font-size: 30px;
  font-family: 'Poppins';
}
section.calculator .heading .Matrix {
  background: var(--matrixBG-color);
  background-size: 10px 2000px;
  border-radius: 8px 0px 0px 8px;
}
```

```
section.calculator .heading .Calculator {
  color: var(--Calc-color);
  background: rgba(255, 255, 255, 0.7);
  border-radius: 0px 8px 8px 0px;
}
section.calculator .headingAB {
  font-family: 'Poppins';
  color: white;
  font-weight: bold;
  text-align: center;
  background: var(--matrixBG-color);
  background-position: center;
  margin-bottom: 3px;
}
section.calculator .headingR {
  font-family: 'Poppins';
  color: var(--Calc-color);
  font-weight: bold;
  text-align: center;
  background: rgba(255, 255, 255, 0.7);
  background-position: center;
  margin-bottom: 3px;
}
section.calculator .enternumber {
  width: 5em;
  background: rgba(255, 255, 255, 0.8);
  color: var(--Calc-color);
  border: 3px inset #323232;
  border-radius: 0.3em;
  font-weight: bold;
}
section.calculator .result {
  width: 5em;
  background: rgba(255, 255, 255, 0.8);
  color: var(--Calc-color);
  border: 3px inset #323232;
  border-radius: 0.3em;
  font-weight: bold;
```

```
}
.line{
    height: 3px;
    margin: 7px;
    background-color: rgba(255, 255, 255, 0.9);
    border: 1.5px solid var(--matrixBG-color);
}
.button {
  color: #1e1e1e;
  text-align: center;
  text-decoration: none;
  background-color: rgba(255, 255, 255, 0.8);
  border: 2px solid var(--Calc-color);
  border-radius: 3px;
  font-size: 14px;
  cursor: pointer;
  height: 25px;
  width: 95px;
  -webkit-box-shadow: 1.5px 2px var(--Calc-color);
          box-shadow: 1.5px 2px var(--Calc-color);
}
.button:hover {
  background-color: rgba(30, 30, 30, 0.8);
  color: rgba(255, 255, 255, 0.9);
}
.button:active {
  -webkit-transform: translateY(0.9px);
          transform: translateY(0.9px);
}
.buttons1 {
  width: 569px;
  margin-left: 3px;
}
.buttons2 {
  width: 356px;
  margin-left: 3px;
  text-align: center;
}
```

```
/*# sourceMappingURL=Calc.css.map */
   </style>
</head>
<body>
            <section class="calculator">
                        <table class="boxes">
                                    <caption class="heading"> <span class="Matrix">  Matrix
 </span></span><span class="Calculator">  Calculator  </span></caption>
                                    <div>
                                                <tr>
                                                            <td>
                                                                        <table>
                                                                                    <caption
class="headingAB">Matrix A</caption>
                                                                                    <tr>
                                                                                                <td><input
class="enternumber" type="text" id="a" maxlength="9"></td>
                                                                                                <td><input
class="enternumber" type="text" id="b" maxlength="9"></td>
                                                                                    </tr>
                                                                                    <tr>
                                                                                                <td><input
class="enternumber" type="text" id="c" maxlength="9"></td>
                                                                                                <td><input
class="enternumber" type="text" id="d" maxlength="9"></td>
                                                                                    </tr>
                                                                        </table>
                                                            </td>

                                                            <td>
                                                                        <table>
                                                                                    <caption
class="headingAB">Matrix B</caption>
                                                                                    <tr>
                                                                                                <td><input
class="enternumber" type="text" id="e" maxlength="9"></td>
                                                                                                <td><input
class="enternumber" type="text" id="f" maxlength="9"></td>
                                                                                    </tr>
                                                                                    <tr>
                                                                                                <td><input
class="enternumber" type="text" id="g" maxlength="9"></td>
```

```
<td><input
class="enternumber" type="text" id="h" maxlength="9"></td>
</tr>
</table>
</td>
<td>
<table>
<caption
class="headingR">Result</caption>
<tr>
<td><input
class="result" type="text" id="i" disabled></td>
<td><input
class="result" type="text" id="j" disabled></td>
</tr>
<tr>
<td><input
class="result" type="text" id="k" disabled></td>
<td><input
class="result" type="text" id="l" disabled></td>
</tr>
</table>
</td>
</tr>
</div>
</table>
<hr class="line">
<div class="buttons">
<table class="buttons1">
<tr class="row1">
<td><input class="button" type="button" value="Add"
onclick="add()"/></td>
<td><input class="button" type="button" value="Subtract"
onclick="subtract()"/></td>
<td><input class="button" type="button" value="Multiply"
onclick="multiply()"/></td>
<td><input class="button" type="button" value="Clear all"
onclick="clearall()"/></td>
</tr>
</table>
<table class="buttons2">
<tr class="row2">
```

```
                    <td><input class="button" type="button" value="Inverse A"
onclick="inverseA()"/></td>
                    <td><input class="button" type="button" value="Inverse B"
onclick="inverseB()"/></td>
                    <td></td>
                    <td></td>
                </td>

                <tr class="row2">
                    <td><input class="button" type="button" value="Transpose
A" onclick="transposeA()"/></td>
                    <td><input class="button" type="button" value="Transpose
B" onclick="transposeB()"/></td>
                    <td></td>
                    <td></td>
                </td>
                <tr class="row2">
                    <td><input class="button" type="button" value="Clear A"
onclick="clearA()"/></td>
                    <td><input class="button" type="button" value="Clear B"
onclick="clearB()"/></td>
                    <td></td>
                    <td></td>
                </td>
            </table>
        </div>
    </section>
</body>

</html>
```