Computer Science - Data Structures

Topic: Concrete Data Structures

Approach: Algebraic Calculus, Linear Algebra

CONTEXT

You have been hired by AlphaReflex, a MP's subsidiary. The development of a logical model of a basic system for algebraic calculus is needed.

The module will be used as an API for the physics system of a game in development; The logical module will provide some basic operations between mathematical objects such as: Sets, Vectors, Lines, Planes, Matrices, and additional elements such as Multivariable Linear Functions over the real scalar field and Linear vector transformations.

REQUIREMENTS

Sets Operations

- $\star$ Addition
- $\star$ Deletion
- $\star$ Subset operator
- $\star$ Subset-equal operator
- $\star$ Membership
- $\star$ Cardinality
- $\star\star$ Ordinality
- $\star\star\star$ Natural, left and right joins
- $\star$ Intersection
- $\star$ Union
- $\star$ Difference
- $\star$ Relative complement
- $\star$ Symmetric difference
- $\star$ Cartesian product

Vector Related Operations

- $\star$ Sum
- $\star$ Inner product
- $\star$ Scalar multiplication
- $\star$ Wedge product
- $\star$ Vector norm
- $\star$ Angle between two vectors
- $\star$ Scalar projection
- $\star$ Vector projection
- $\star$ Unitary vector

Planes and Lines Operations

- $\star^{\star}$ Intersection of two planes.
- $\star^{\star}$ Intersection of tree planes.
- $\star\star$ Intersection between a line and plane.
- $\star$ Angle between two planes.
- $\star$ Angle between two lines.
- $\star\star$ Build a line using an orthogonal plane.
- $\star\star$ Plane building using a point and an orthogonal line to it.
- $\star\star$ Build a line using an orthogonal line.

Matrices

- $\star$ Sum
- $\star\star$ Direct sum
- $\star$ Scalar multiplication
- $\star\star$ Multiplication
- $\star$ Natural trace
- $\star$ Transposed
- $\star\star$ Determinant
- $\star\star\star$ Inverse by Gauss-Jordan
- $\star\star\star$ LU Factor
- $\star\star$ Eigenvectors and Eigenvalues
- $\star^{\star\star}$ Characteristic polynomial

Linear Operations

- $\star^{\star}$ To define a multi-variable linear function
- $\star^{\star}$ To define a mono-variable linear transformation

  Simulating the values it takes by stepping.

For these operations[1], one can have the possibility of define the linear function and a stepping sequence for the variables in it.[2]

MATHEMATICAL OBJECT NOTATION

The current section shows the requirements itself[3] by describing the operations, notation and procedures on a mathematical form.

Is expected that student reach the capability of recognize the kind of algorithm to develop.

[1] Additionally, is necessary a complexity analysis for each operation related to a particular object.

[2] The mathematical objects mentioned above must be transformed in classes or structures according with the used programming language.

[3] Remembering that it is an academic project, some definitions needed are mentioned here, but in a real job it probably would not happen, in the next project is assumed that the student knows the context in which he is placed.

Sets

Let $\boldsymbol{A},\boldsymbol{B}$ as sets of n,m sizes which has unknown characteristics, but are both discrete and countable.

$$\boldsymbol{A}:\{a_1,a_2,...,a_n\},\boldsymbol{B}:\{b_1,b_2,...,b_m\}$$

Operation	Notation
Addition	$add(e,\boldsymbol{A})$
Deletion	$remove(e,\boldsymbol{A})$
Subset	$\boldsymbol{A}\subset\boldsymbol{B}$
Subset-equal	$\boldsymbol{A}\subseteq\boldsymbol{B}$
Membership	$e\in\boldsymbol{A}$
Cardinality	$\Vert\boldsymbol{A}\Vert=n\in\mathbb{Z}$
Ordinality	$order(\boldsymbol{A},fettle)$
Natural join	$\boldsymbol{A}\bowtie\boldsymbol{B}=\boldsymbol{C}$
Left join	$\boldsymbol{A}\ltimes\boldsymbol{B}=\boldsymbol{C}$
Right join	$\boldsymbol{A}\rtimes\boldsymbol{B}=\boldsymbol{C}$
Intersection	$\boldsymbol{A}\cap\boldsymbol{B}$
Union	$\boldsymbol{A}\cup\boldsymbol{B}$
Difference	$\boldsymbol{A}-\boldsymbol{B}$
Complement	$\boldsymbol{A}\complement\boldsymbol{B}$
Symmetric difference	$\boldsymbol{A}\Delta\boldsymbol{B}$
Cartesian product	$\boldsymbol{A}\times\boldsymbol{B}$

† *Addition:* consist in the insertion of an element. Defined the set

$$\boldsymbol{A}:\{a_1,a_2,a_3\}$$

when the addition is effected over the set, the element will be inserted as the follows:

$$add(a_4,\boldsymbol{A})\Rightarrow\boldsymbol{A}:\{a_1,a_2,a_3,a_4\}$$

Note that adding a new element to $\boldsymbol{A}$ is an union with an unitary set, where the inserted element is added at the end of the set.

† *Deletion:* consists in the operation of verifying that any element e is in the set $\boldsymbol{A}$ and then if exists remove it from the set. as can see:

$$remove(a_1,\boldsymbol{A})\equiv\boldsymbol{A}:\{a_2,a_3,a_4\}$$

† *Subset:* operator consists in checking that a set $\boldsymbol{A}$ is contained by a set $\boldsymbol{B}$.

† *Subset-equal:* is the last mentioned operation plus checking the possibility that set $\boldsymbol{A}$ be equal to $\boldsymbol{B}$.

† *Membership:* responds if an element e is in a set $\boldsymbol{A}$.[4]

[4] These operations are boolean return type

† *Cardinality:* denotes the number of elements contained in a set.

† *Ordinality:* if the set has an order according to a specification or checking rule.

† *Natural join:* is a set operation which compares all the objects in between two sets and returns the objects which presents equality in a defined aspect (including properties of it or the object itself) it can be seen as an intersection.

† *Left and right joins:* operates the same that natural join but are no commutative, because them can add in the return, the elements in which the property is both null for the objects while natural join does not accomplish that task.

† *Intersection:* $\boldsymbol{A}\cap\boldsymbol{B}$ returns the common elements between the sets $\boldsymbol{A},\boldsymbol{B}$.

† *Union:* $\mathbf{A}\cup\boldsymbol{B}$ returns all the elements of both sets.

† *Difference:* $\boldsymbol{A}-\boldsymbol{B}$ the elements which are in $\boldsymbol{A}$ and not are in $\boldsymbol{B}$.

† *Complement:* $\boldsymbol{A}\complement\boldsymbol{B}$ returns the elements that are not in $\boldsymbol{A}$.

† *Symmetric difference:* returns the elements that are not in the intersection of two sets.

† *Cartesian product:* returns the combination of each element in $\boldsymbol{A}$ which each element in $\boldsymbol{B}$.

Vectors:

$$\boldsymbol{V}:\{\mathbf{v}|\mathbf{v}\in\mathbb{R}^3\}$$

let$\boldsymbol{V}$ be the set of all the vectors that belongs to $\mathbb{R}^3$.

Operation	Notation
Sum	$\mathbf{v}_1+\mathbf{v}_2=\mathbf{v}_3$
Inner product	$\mathbf{v}_1\cdot\mathbf{v}_2=\alpha\in\mathbb{R}$
Scalar multiplication	$\alpha\cdot\mathbf{v};\alpha\in\mathbb{R}$
Wedge product	$\mathbf{v}_1\times\mathbf{v}_2=\mathbf{v}_3$
Vector norm	$\Vert\mathbf{v}\Vert=v$
Angle between vectors	$\mathbf{v}_1\measuredangle\mathbf{v}_2$
Scalar projection	$proj_{\mathbf{v}_2}(\mathbf{v}_1)$
Vector projection	$comp_{\mathbf{v}_2}(\mathbf{v}_1)$
Unitary vector	$\hat{\mathbf{v}}=\Vert\mathbf{v}\Vert^{-1}\cdot\mathbf{v}$

† *Sum:*

$$\boldsymbol{v}=\langle v_1,v_2,v_3\rangle \tag{1}$$

$$\boldsymbol{w}=\langle w_1,w_2,w_3\rangle \tag{2}$$

$$\boldsymbol{v}+\boldsymbol{w}=\boldsymbol{w}+\boldsymbol{v}=\langle v_1+w_1,v_2+w_2,v_3+w_3\rangle \tag{3}$$

† *Inner product:*

$$\langle \boldsymbol{v}|\boldsymbol{w}\rangle = \boldsymbol{v}\cdot\boldsymbol{w} = v_1\cdot w_1 + v_2\cdot w_2 + v_3\cdot w_3 \tag{4}$$

† *Scalar multiplication:*

$$k\boldsymbol{v} = k\cdot\boldsymbol{v} = \langle kv_1, kv_2, kv_3\rangle \tag{5}$$

† *Wedge product:*

$$\mathbf{v}_1\times\mathbf{v}_2 = \begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{bmatrix} \tag{6}$$

$$= (v_1w_2 - v_2w_1)\hat{\mathbf{x}} - (v_1w_3 - v_3w_1)\hat{\mathbf{y}} + (v_2w_3 - v_3w_2)\hat{\mathbf{z}} \tag{7}$$

† *Vector norm:*

$$\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + v_3^2} \tag{8}$$

† *Angle between vectors:*

$$\mathbf{v}_1\measuredangle\mathbf{v}_2 = \arccos\left(\frac{\mathbf{v}_1\cdot\mathbf{v}_2}{\|\mathbf{v}_1\|\cdot\|\mathbf{v}_2\|}\right) = \beta \tag{9}$$

† *Scalar projection:*

$$proj_{\mathbf{v}_2}(\mathbf{v}_1) = \frac{\mathbf{v}_1\cdot\mathbf{v}_2}{\|\mathbf{v}_2\|^2}\mathbf{v}_2 \tag{10}$$

† *Vector projection:*

$$comp_{\mathbf{v}_2}(\mathbf{v}_1) = \frac{\mathbf{v}_1\cdot\mathbf{v}_2}{\|\mathbf{v}_2\|} \tag{11}$$

† *Unitary vector:*

$$\hat{\mathbf{v}} = \|\mathbf{v}\|^{-1}\cdot\mathbf{v} \tag{12}$$

Planes and Lines

$$\Pi : \{\pi \mid \pi\subset\mathbb{R}^3\}; L : \{l \mid l\subset\mathbb{R}^3\}$$

Let Π, L be respectively sets that storage all the planes and lines that are subsets of $\mathbb{R}^3$

Operation	Notation
Intersection of two planes	$\pi_1\cap\pi_2 = l$
Intersection of tree planes	$\pi_1\cap\pi_2\cap\pi_3 = P$
Intersection between a line and plane	$\pi\cap l = P$
Angle between two planes	$\pi_1\measuredangle\pi_2 = \beta$
Angle between two lines	$l_1\measuredangle l_2 = \beta$
Line building using an orthogonal plane	$l_b(\pi)$
Plane building using a point and an orthogonal line to it	$\pi_b(l, P)$
Build a line using an orthogonal line	$l_b(l)$

This section is demonstrative, you have to define methods for doing an operation, proof and then code them.

Matrices

Operation	Notation
Sum	$A + B = C$
Direct sum	$A\oplus B = C$
Scalar multiplication	kA
Multiplication	$AB = C$
Determinant	$\|A\|\equiv\det(A)\equiv\lvert A\rvert$
Natural trace	$Tr(A)$
Transposed	A^T
Inverse by Gauss-Jordan	$[I\lvert A^{-1}]$
LU Factor	$LU = A$
Eigenvectors and Eigenvalues	$A\mathbf{v} = \lambda\mathbf{v}$
Characteristic polynomial	$(\lambda I - A) = 0$

† *Matrix definition:*

$$A_{n\times m} := \begin{Bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{Bmatrix} := (a_{ij})_{m\times n} \tag{13}$$

† *Sum:*

$$A_{n\times m} + B_{n\times m} := (a_{ij} + c_{ij})_{n\times m} := (c_{ij}) \tag{14}$$

† *Direct sum:*

$$\begin{Bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{Bmatrix} \bigoplus \begin{Bmatrix} b_{11} & b_{12} & \dots & b_{1s} \\ b_{21} & b_{22} & \dots & b_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ b_{r1} & b_{r2} & \dots & b_{rs} \end{Bmatrix} \tag{15}$$

$$= \begin{Bmatrix} a_{11} & a_{12} & \dots & a_{1n} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1s} \\ 0 & 0 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2s} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & b_{r1} & b_{r2} & \dots & b_{rs} \end{Bmatrix} \tag{16}$$

$$A_{m\times n}\oplus B_{r\times s} = C_{(m+s)\times(n+r)} \tag{17}$$

† *Scalar multiplication:*

$$kA := (ka_{ij})_{m\times n} = \begin{bmatrix} ka_{11} & ka_{12} & \dots & ka_{1n} \\ ka_{21} & ka_{22} & \dots & ka_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ ka_{m1} & ka_{m2} & \dots & ka_{mn} \end{bmatrix} \tag{18}$$

† *Multiplication:*

$$A_{m\times n}B_{n\times r} := (c_{ij})_{m\times r} = \sum_{k=1}^{n} a_{ik}b_{kj} \tag{19}$$

† *Natural trace:*

$$Tr(A_{n\times n}) := \prod_{i=1}^{n} a_{ii} \tag{20}$$

† *Transposed:*

$$A^T := a_{ji} \tag{21}$$

† *Determinant:*

$$A := (a_{ij})_{n\times n} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \tag{22}$$

$$\det(A) = \det(L)\det(U) = \det(U) \tag{23}$$

† *Inverse by Gauss-Jordan:*

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & 0 & 0 & \dots & 1 \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} 1 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1n} \\ 0 & 1 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} \tag{25}$$

$$A^{-1} := \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} \tag{26}$$

† *LU Factor:*

$$A := \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{27}$$

$$\equiv \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \tag{28}$$

$$= LU \tag{29}$$

† *Eigenvectors and Eigenvalues:*

$$A\mathbf{v} = \lambda\mathbf{v} \tag{30}$$

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \lambda[v_1, v_2, \dots, v_n] \tag{31}$$

† *Characteristic polynomial:*

$$\because A\mathbf{v} = \lambda\mathbf{v} \tag{32}$$

$$\therefore A\mathbf{v} = \lambda\mathbf{v} = \lambda I\mathbf{v} \tag{33}$$

$$\because A\mathbf{v} = \lambda I\mathbf{v} \tag{34}$$

$$\therefore \lambda I\mathbf{v} - A\mathbf{v} = 0 \tag{35}$$

$$\therefore (\lambda I - A)\mathbf{v} = 0 \tag{36}$$

$$P_A(\lambda) = \det(\lambda I - A) \tag{37}$$

$$\text{if } P_A(\lambda) = 0; \text{ then } \lambda \text{ is an eigenvalue}$$

Linear Operations

† *Linear functions:*

$$g : \mathbb{R} \to \mathbb{R} \tag{38}$$

$$g(x) = ax + b \tag{39}$$

$$f : \mathbb{R}^n \to \mathbb{R} \tag{40}$$

$$x \mapsto f(x_1, x_2, \dots, x_n) \tag{41}$$

$$f(x) = \sum_{i=1}^{n} g_i(x_i) \tag{42}$$

† *Linear vector transformations:*

$$f : \mathbb{R} \to \mathbb{R} \tag{43}$$

$$f(x) = ax + b \tag{44}$$

$$\mathbf{v}_n = [v_a, v_b, v_c, \dots, v_n]; \mathbf{v}_n \in \mathbb{R}^n \tag{45}$$

$$T : \mathbb{R}^n \to \mathbb{R}^n \tag{46}$$

$$\mathbf{v}_n \mapsto T(\mathbf{v}_n) \tag{47}$$

$$T(\mathbf{v}_n) = \begin{bmatrix} f_1(v_a) \\ f_2(v_b) \\ f_3(v_c) \\ \vdots \\ f_n(v_n) \end{bmatrix} \tag{48}$$

† *Linear matrix transformations:*

$$f : \mathbb{R} \to \mathbb{R} \tag{49}$$
$$f(x_i) = a_i x_i + b_i \tag{50}$$

$$A_{mn} := \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \tag{51}$$

$$T : \mathbb{K}_{n\times m} \to \mathbb{K}_{n\times m} \tag{52}$$
$$A_{mn} \mapsto T(A_{mn}) \tag{53}$$

$$T(A_{mn}) = \begin{bmatrix} f_{11}(A_{11}) & f_{12}(A_{12}) & \dots & f_{1n}(A_{1n}) \\ f_{21}(A_{21}) & f_{22}(A_{22}) & \dots & f_{2n}(A_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{m1}(A_{m1}) & f_{n2}(A_{m2}) & \dots & f_{mn}(A_{mn}) \end{bmatrix} \tag{54}$$

Clarification

Commonly you may ask, what do I refer when I say that the intersection of two planes is needed. I refer to a quasi-interpolation reconstruction, that because what is asked to you is to give a set of spatial points where two planes intersects.

Now, Let $\mathscr{P}_1, \mathscr{P}_2$ be two planes in $\mathbb{R}^3$:

$$\mathscr{P}_1 : a_1x + b_1y + c_1z = d_1$$
$$\mathscr{P}_2 : a_2x + b_2y + c_2z = d_2$$

then those planes has a normal form:

$$\mathscr{P}_1 : \langle a_1, b_1, c_1\rangle \cdot \langle x, y, z\rangle = d_1$$
$$\mathscr{P}_2 : \langle a_2, b_2, c_2\rangle \cdot \langle x, y, z\rangle = d_2$$

Now, remember that the cross product of two vectors generates an orthogonal one to those. Hence:

$$\langle a_1, b_1, c_1\rangle \times \langle a_2, b_2, c_2\rangle = \mathbf{v}$$

is the direction vector. And what remains is to find a point in the intersection line. This is solved by doing:

$$\boxed{z = 0}$$
$$\therefore a_1x + b_1y = d_1$$
$$\therefore a_2x + b_2y = d_2$$

obtaining a system of linear equations. If we take the first equation and solve by substitution:

$$\implies x = \frac{d_1 - b_1y}{a_1}$$
$$\implies a_2\frac{d_1 - b_1y}{a_1} + b_2y = d_2$$
$$\implies \frac{-a_2b_1}{a_1}y + b_2y = d_2 - \frac{a_2d_1}{a_1}$$
$$\implies \frac{a_1b_2 - a_2b_1}{a_1}y = d_2 - \frac{a_2d_1}{a_1}$$
$$\implies \boxed{y = \frac{a_1}{a_1b_2 - a_2b_1}\left(d_2 - \frac{a_2d_1}{a_1}\right)}$$

$$\implies \boxed{x = \frac{d_1 - b_1}{a_1b_2 - a_2b_1}\left(d_2 - \frac{a_2d_1}{a_1}\right)}$$

we get the values x, y, z to substitute in the position vector for constructing the line of intersection between the planes:

$$\mathscr{P}_1 \cap \mathscr{P}_2 = \left\langle \frac{d_1 - b_1}{a_1b_2 - a_2b_1}\left(d_2 - \frac{a_2d_1}{a_1}\right), \frac{a_1}{a_1b_2 - a_2b_1}\left(d_2 - \frac{a_2d_1}{a_1}\right), 0 \right\rangle + \gamma\langle a_1, b_1, c_1\rangle \times \langle a_2, b_2, c_2\rangle$$

Where γ is the generator value which independently of x, y, z constructs the intersection line of those planes. (γ is a variable).

Why quasi-interpolation?: Because we analytically can derive the formula, however this data is being manipulated in a computer, therefore the generated set of values has to be discrete and in addition we are giving more a numerical approximation rather than using an interpolation method.

Additional Requirements

- ⋆ All the algebraic sets must have the capacity of storage numbers, letters, in general, any kind of object.
- ⋆ The sets can be built from an unordered way and there must exists a form to obtain an ordered instance of them.
- ⋆ Attempts of copy from internet source code will meant the signature fail and other consequences.
- ⋆ Is elemental that the project distribution be given by the MVC architecture.
- ⋆ Lines should be managed as vector functions for simplifying some operations, the planes must be managed in vector form.

Notes

- ⋆ You are not allowed to use any native class from any std library present in the program language that you are going to use. Given the case, your real qualification will be recalculated deducting 20% from the original one.
- ⋆ Questions with the instructor (me). **Do not be shy**, if you do not know how to do something then ask. But first try to make a brief supposition about how to solve the problem.

Support Material

[1] D. Poole, Linear Algebra: A Modern Introduction. Cengage Learning, 2014.
[2] S. Lang, Introduction to Linear Algebra. Springer, 1997.
[3] J. Hefferon, Linear Algebra. Orthogonal Publishing L3c, 2017.
[4] K. Hoffman and R. Kunze Alden, Linear Algebra. Prentice Hall, 1971.
[5] S. Grossman I., Elementary Linear Algebra. Brooks/Cole Publishing Company, 1994.
[6] S. Boyd and L. Vandenberghe, Introduction to Applied Linear Algebra. Cambridge University Press, 2018.