Conditionality of Linear Systems of Equation and Matrices Using Projective Geometric Algebra

Vaclav Skala(✉)

University of West Bohemia, Pilsen, Czechia
skala@kiv.zcu.cz
http://www.VaclavSkala.eu

Abstract. Linear systems of equations and their reliable solution is a key part of nearly all computational systems and in a solution of many engineering problems. Mostly, the estimation of the matrix conditionality is used for an assessment of the solvability of linear systems, which are important for interpolation, approximation, and solution of partial differential equations especially for large data sets with large range of values.

In this contribution, a new approach to the matrix conditionality and the solvability of the linear systems of equations is presented. It is based on the application of the *geometric algebra* with the projective space representation using homogeneous coordinates representation. However, the physical floating-point representation, mostly the IEEE 754-219, has to be strictly respected as many algorithms fail due to this.

Keywords: Linear systems of equations · Matrix conditionality · Geometric algebra · Projective space · Duality · Radial basis function · RBF · Partial differential equations · PDE

1 Introduction

Solutions of linear systems of equations are probably one of the most used operations in many applications. Generally, there are two main groups of those:

- non-homogeneous systems of linear equations, i.e. $\mathbf{Ax} = \mathbf{b}$
- homogeneous system of equations, i.e. $\mathbf{Ax} = \mathbf{0}$

Those two groups of linear systems are solved differently, in spite of the fact, that they are dual problems in some sense. Therefore, there should be common methods for solving both groups of such linear systems of equations. The first case is deeply analyzed in the courses of numerical mathematics, while the second one is considered as marginal without advanced solution methods. (In the following, standard properties of the matrices are expected).

Using the *principle of duality* and *projective extension of the Euclidean space* the first type of the linear system, i.e. $\mathbf{Ax} = \mathbf{b}$, can be easily transformed to the second type,

This research was supported by the Czech Science Foundation (GACR) project GA 17-05534S.

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12250, pp. 3–17, 2020.
https://doi.org/10.1007/978-3-030-58802-1_1

i.e. $\mathbf{Ax} = \mathbf{0}$. The *geometric algebra* offers more general formalism, which can be used for a better understanding of the linear system of equations properties and behavior.

However, in practical applications, any implementation has to take into consideration the IEEE 754-219 or similar standards of the floating-point physical representation.

1.1 Geometrical Interpretation

Both types of the linear systems of equations, i.e. $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{Ax} = \mathbf{0}$, have a simple geometrical interpretation, which can be easily demonstrated on two simple problems in the E^2 case, see Fig. 1:

- an intersection computation of two lines, i.e. $\mathbf{Ax} = \mathbf{b}$; the matrix $\mathbf{A}$ is $n \times n$
- a line determination given by two points, i.e. $\mathbf{Ax} = \mathbf{0}$; the matrix $\mathbf{A}$ is $n \times (n+1)$

Similarly, in the E^3 case, where three planes intersect in a point or three points define a plane, if no singular cases are considered.

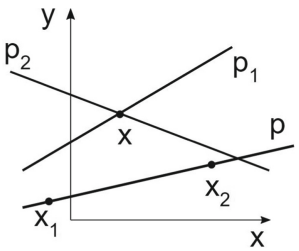

Fig. 1. Examples of "meet" and "union" operations in the E^2 case

Also, the singular or close to the singular case has to be properly solved in both cases. In the first case, if the lines are collinear, in the second one, if two points are the same.

Unfortunately, programmers do not solve such cases correctly in many cases. Usually, a condition like $det(\mathbf{A}) <= eps$ is used for a "*singularity*" detection and eps is taken as a "reasonable" small number.

Let us consider the intersection computation of two lines in the E^2 for simplicity. The Cramer's rule can be used and the solution of $\mathbf{Ax} = \mathbf{b}$ is given as $x = Det_x/Det$ and $y = Det_y/Det$. If $Det_x \sim det(\mathbf{A})$ then the x value might be close to 1 regardless of the eps value.

As those two problems in Fig. 1 are *dual*, i.e. the *meet* operator in the first case and *union* operator in the second one, the *principle of duality* must be kept [7]. Similarly, in the E^3 case in Fig. 2.

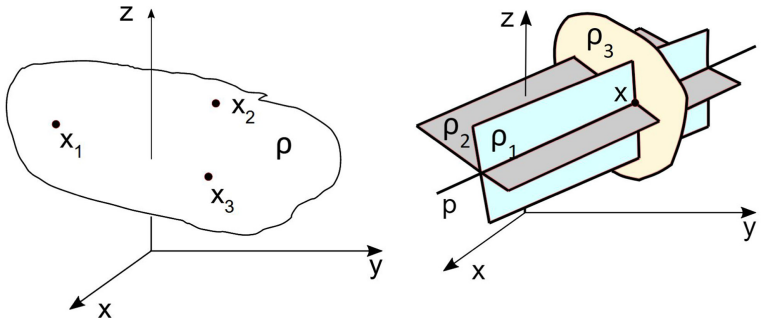

Fig. 2. Examples of "meet" and "union" operations in the E^3 case

1.2 Geometric Algebra

The vector algebra (Gibbs algebra), which is used in engineering practice, uses two basic operations on two vectors $\mathbf{a}, \mathbf{b}$ in E^n:

- the inner product (scalar product or dot product) $c = \mathbf{a} \cdot \mathbf{b}$, where c is a scalar value
- the outer product $\mathbf{c} = \mathbf{a} \wedge \mathbf{b}$ (the cross-product in E^3 $\mathbf{c} = \mathbf{a} \times \mathbf{b}$), where $\mathbf{c}$ is a bivector and has different properties as it represents an oriented area in the n-dimensional space, in general.

The *Geometric Algebra* (GA) uses a "new" product called *geometric product* defined as:

$$\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b} \tag{1}$$

where $\mathbf{ab}$ is the new entity. It should be noted, that it is a "bundle" of objects with different dimensionalities and properties, in general.

In the case of the n-dimensional space, the vectors are defined as $\mathbf{a} = (a_1\mathbf{e}_1 + ... + a_n\mathbf{e}_n)$, $\mathbf{b} = (b_1\mathbf{e}_1 + ... + b_n\mathbf{e}_n)$ and the $\mathbf{e}_i$ vectors form orthonormal vector basis in E^n. In the E^3 case, the following objects can be used if geometric algebra [26]:

1	0-vector (scalar)	$\mathbf{e}_{12}, \mathbf{e}_{23}, \mathbf{e}_{31}$	2-vectors (bivectors)
$\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$,	1-vector (vectors)	$\mathbf{e}_{123}$	3-vector (pseudoscalar)

The significant advantage of the geometric algebra is, that it is more general that than the Gibbs algebra and can handle all objects with dimensionality up to n.

The geometry algebra uses the following operations, including the inverse of a vector.

$$\mathbf{a} \cdot \mathbf{b} = \frac{1}{2}(\mathbf{ab} + \mathbf{ba}) \quad \mathbf{a} \wedge \mathbf{b} = -\mathbf{b} \wedge \mathbf{a} \quad \mathbf{a}^{-1} = \mathbf{a}/||\mathbf{a}||^2 \tag{2}$$

It should be noted, that geometric algebra is *anti-commutative* and the "pseudoscalar" I in the E^3case has the basis $\mathbf{e}_1\mathbf{e}_2\mathbf{e}_3$ (briefly as $\mathbf{e}_{123}$), i.e.

$$\mathbf{e}_i\mathbf{e}_j = -\mathbf{e}_j\mathbf{e}_i \quad \mathbf{e}_i\mathbf{e}_i = 1 \quad \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3 = I \quad \mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c} = q \tag{3}$$

where q is a scalar value (actually a pseudoscalar).

The geometric product of two vectors is represented as:

$$\mathbf{ab} = \sum_{i,j=1}^{n,n} a_i\mathbf{e}_i b_j\mathbf{e}_j \quad \mathbf{a}\cdot\mathbf{b} = \sum_{i=1}^{n,n} a_i\mathbf{e}_i b_i\mathbf{e}_i \tag{4}$$

$$\mathbf{a}\wedge\mathbf{b} = \sum_{i,j=1\&i\neq j}^{n,n} a_i\mathbf{e}_i b_j\mathbf{e}_j = \sum_{i,j=1,\&i>j}^{n} (a_i b_j - a_j b_i)\mathbf{e}_i\mathbf{e}_j \tag{5}$$

It is not a "friendly user" notation for a practical application and causes problems in practical implementations, especially due to anti-commutativity of the geometric product.

However, the geometric product can be easily represented by the *tensor product*, which can be easily represented by a matrix. The tensor product for the 4-dimensional case, as the homogeneous coordinates will be used in the following, is represented as:

$$\mathbf{ab} \underset{\text{repr}}{\Longleftrightarrow} \mathbf{ab}^T = \mathbf{a}\otimes\mathbf{b} = \mathbf{Q} = \begin{bmatrix} a_1b_1 & a_1b_2 & a_1b_3 & a_1b_4 \\ a_2b_1 & a_2b_2 & a_2b_3 & a_2b_4 \\ a_3b_1 & a_3b_2 & a_3b_3 & a_3b_4 \\ a_4b_1 & a_4b_2 & a_4b_3 & a_4b_4 \end{bmatrix} = \mathbf{B}+\mathbf{U}+\mathbf{D} \tag{6}$$

where $\mathbf{B}+\mathbf{U}+\mathbf{D}$ are Bottom triangular, Upper triangular, Diagonal matrices, a_4, b_4 will be used for the homogeneous coordinates, i.e. actually w_a, w_b (the projective notation will be explained later), and the operator $\otimes$ means the anti-commutative tensor product.

1.3 Projective Extension of the Euclidean Space

Let us consider the projective extension of the Euclidean space in the E^3 case and use of the homogeneous coordinates. The "hidden" advantage of this is that it enables us to represents points or values close to infinity.

Let us consider a vector $\mathbf{a} = [a_1, a_2, a_3 : a_4]^T, a_4 \neq 0$, which represents actually the vector $(a_1/a_4, a_2/a_4, a_3/a_4)$ in the E^3 space. The ":" symbol within the projective representation is used to distinguish different properties. In the case of a plane, the vector (a_1, a_2, a_3) represents the "*normal vector*" of a plane (actually it is a bivector), while a_4 actually represents a pseudo-distance of a plane from the origin.

Let us consider a second vector $\mathbf{b} = [b_1, b_2, b_3 : b_4]^T, b_4 \neq 0$. It can be seen, that the diagonal of the matrix $\mathbf{Q}$ actually represents the inner product in the projective representation as:

$$[(a_1b_1 + a_2b_2 + a_3b_3) : a_4b_4]^T \equiv \frac{a_1b_1 + a_2b_2 + a_3b_3}{a_4b_4} \tag{7}$$

where " $\equiv$ " means projectively equivalent. In the E^3 case, the inner product actually projectively represents the trace $tr(\mathbf{Q})$ of the matrix $\mathbf{Q}$. The outer product (the cross-product in the E^3 case) is then represented respecting the anti-commutativity as:

$$\mathbf{a}\wedge\mathbf{b} \underset{\text{repr}}{\Longleftrightarrow} \sum_{i,j=1\&i>j}^{3,3} (a_i b_j\mathbf{e}_i\mathbf{e}_j - b_i a_j\mathbf{e}_i\mathbf{e}_j) = \sum_{i,j\&i>j}^{3,3} (a_i b_j - b_i a_j)\mathbf{e}_i\mathbf{e}_j \tag{8}$$

It should be noted, that the outer product can be used for a solution of a linear system of equations $\mathbf{Ax} = \mathbf{b}$ or $\mathbf{Ax} = \mathbf{0}$, too.

1.4 Principle of Duality

The principle of duality is an important principle, in general. The duality principle for basic geometric entities and operators is presented by Table 1 and Table 2 [18].

Table 1. Duality of geometric entities

Duality of geometric entities					
Point in E^2	$\underset{\text{DUAL}}{\Longleftrightarrow}$	Line in E^3	Point in E^2	$\underset{\text{DUAL}}{\Longleftrightarrow}$	Plane in E^3

Table 2. Duality of operators

Duality of operators		
Union $\cup$	$\underset{\text{DUAL}}{\Longleftrightarrow}$	Intersection $\cap$

It means, that in the E^2 case a point is dual to a line and vice versa, the intersection of two lines is dual to a union of two points, i.e. line given by two points; similarly for the E^3 case, where a point is dual to a plane [7,14,15]. The direct consequence of the principle of duality is that, the intersection point $\mathbf{x}$ of two lines $\mathbf{p}_1, \mathbf{p}_2$, resp. a line $\mathbf{p}$ passing two given points $\mathbf{x}_1, \mathbf{x}_2$, is given as:

$$\mathbf{x} = \mathbf{p}_1 \wedge \mathbf{p}_2 \underset{\text{DUAL}}{\Longleftrightarrow} \mathbf{p} = \mathbf{x}_1 \wedge \mathbf{x}_2 \tag{9}$$

where $\mathbf{p}_i = [a_i, b_i : c_i]^T$, $\mathbf{x} = [x, y : w]^T$ (w is the homogeneous coordinate), $i = 1, 2$; similarly in the dual case.

In the case of the E^3 space, a point is dual to a plane and vice versa. It means that the intersection point $\mathbf{x}$ of three planes ρ_1, ρ_2, ρ_3, resp. a plane ρ passing three given points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ is given as:

$$\mathbf{x} = \rho_1 \wedge \rho_2 \wedge \rho_3 \underset{\text{DUAL}}{\Longleftrightarrow} \rho = \mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \mathbf{x}_3 \tag{10}$$

where $\mathbf{x} = [x, y, z : w]^T$, $\rho_i = [a_i, b_i, c_i : d_i]^T$, $i = 1, 2, 3$.

It can be seen that the above formulae is equivalent to the "extended" cross-product, which is natively supported by the GPU architecture. For an intersection computation, we get:

$$\mathbf{x} = \mathbf{p}_1 \wedge \mathbf{p}_2 = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_w \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{bmatrix} \qquad \mathbf{x} = \rho_1 \wedge \rho_2 \wedge \rho_3 = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_w \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{bmatrix} \tag{11}$$

Due to the principle of duality, the dual problem solution is given as:

$$\mathbf{p} = \mathbf{x}_1 \wedge \mathbf{x}_2 = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_w \\ x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \end{bmatrix} \qquad \rho = \mathbf{x}_1 \wedge \mathbf{x}_2 \wedge \mathbf{x}_3 = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_w \\ x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3 \end{bmatrix} \tag{12}$$

The above presented formulae proved the strength of the formal notation of the geometric algebra approach. Therefore, there is a natural question, what is the more convenient computation of the geometric product, as computation with the outer product, i.e. extended cross product, using standard vector algebra approach is not simple.

Fortunately, the geometric product of ρ_1, ρ_2, resp. of $\mathbf{x}_1$ and $\mathbf{x}_2$ vectors using homogeneous coordinates given as the anti-commutative tensor product is given as:

$\rho_1\rho_2$	a_2	b_2	c_2	d_2
a_1	a_1a_2	a_1b_2	a_1c_2	a_1d_2
b_1	b_1a_2	b_1b_2	b_1c_2	b_1d_2
c_1	c_1a_2	c_1b_2	c_1c_2	a_1d_2
d_1	d_1a_2	d_1b_2	d_1c_2	d_1d_2

$\mathbf{x}_1\mathbf{x}_2$	x_2	y_2	z_2	w_2
x_1	x_1x_2	x_1y_2	x_1z_2	x_1w_2
y_1	y_1x_2	y_1y_2	y_1z_2	y_1w_2
z_1	z_1x_2	z_1y_2	z_1z_2	x_1w_2
w_1	w_1x_2	w_1y_2	w_1z_2	w_1w_2

The question is how to compute a line $\mathbf{p} \in E^3$ given as an intersection of two planes ρ_1, ρ_2, which is the dual problem to a line given by two points $\mathbf{x}_1$, $\mathbf{x}_2$ as those problems are dual. It leads to the Plücker coordinates used especially in robotics [17,21]. Using the geometric algebra, the principle of duality and projective representation, we can directly write:

$$\mathbf{p} = \rho_\mathbf{1} \wedge \rho_\mathbf{2} \quad \underset{\text{DUAL}}{\Longleftrightarrow} \quad \mathbf{p} = \mathbf{x}_1 \wedge \mathbf{x}_2 \tag{13}$$

It can be seen that the formula given above keeps the duality in the final formulae, too. A more detailed description can be found in [17].

It should be noted that the geometric algebra and the principle of duality offers:

- to solve dual problems by one programming sequence
- it naturally supports parallel programming as it uses vector-vector operations as the SSE instructions or GPU can be used
- the solution can avoid a division operation if the result can be left in the projective representation [24]
- results of operations are in symbolic form, which enables further symbolic processing using vector algebra or geometric algebra rules

2 Solution of Linear Systems of Equations

A solution of a linear system of equations is a part of the linear algebra and used in many computational systems. There are many publications related to methods for a linear system of equations following different criteria:

- memory requirements - linear systems with sparse or tri-diagonal matrices [8], etc.
- algorithm computational complexity [25] using a block decomposition

– parallel processing of large linear systems, especially in connection with GPUs applications [28]
– higher precision of computation and numerical stability [3,24].

It should be noted, that the linear system of equations $\mathbf{Ax} = \mathbf{b}$ can be transformed to the homogeneous system of linear equations, i.e. to the form $\mathbf{D}\xi = \mathbf{0}$, where $\mathbf{D} = [\mathbf{A}|-\mathbf{b}]$, $\xi = [\xi_1, ..., \xi_n : \xi_w]^T$, $x_i = \xi_i \,/\, \xi_w$, $i = 1, ..., n$. If $\xi_w \mapsto 0$ then the solution is in infinity and the vector $(\xi_1, ..., \xi_n)$ gives the "direction", only.

As the solution of a linear system of equations is equivalent to the outer product (generalized cross-vector) of vectors formed by rows of the matrix $\mathbf{D}$, the solution of the system $\mathbf{D}\xi = \mathbf{0}$ is defined as:

$$\xi = \mathbf{d}_1 \wedge \mathbf{d}_2 \wedge ... \wedge \mathbf{d}_n \qquad \mathbf{D}\xi = 0, \; i.e. \quad [\mathbf{A}| - \mathbf{b}]\xi = 0 \tag{14}$$

where: $\mathbf{d}_i$ is the i-th row of the matrix $\mathbf{D}$, i.e. $\mathbf{d}_i = (a_{i1}, ..., a_{in}, -b_i)$, $i = 1, ..., n$

The application of the projective extension of the Euclidean space enables us to transform the non-homogeneous system of linear equations $\mathbf{Ax} = \mathbf{b}$ to the homogeneous linear system $\mathbf{D}\xi = \mathbf{0}$, i.e.:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad \Longleftrightarrow_{\text{conversion}} \quad \begin{bmatrix} a_{11} & \cdots & a_{1n} & -b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \cdots & a_{nn} & -b_n \end{bmatrix} \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_n \\ \xi_w \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{15}$$

It is a very important result as a solution of a linear system of equations is formally the same for both types, i.e. homogeneous linear systems $\mathbf{Ax} = \mathbf{0}$ and non-homogeneous systems $\mathbf{Ax} = \mathbf{b}$. It means, that many problems might be transformed to the outer product application, e.g.. computation of barycentric coordinates [13], length, area, volume og geometric entities [12], etc.

As the solution is formally determined by Eq. 14, formal linear operators can be used for further symbolic processing using standard formula manipulation, as the geometry algebra is multi-linear. Even more, it is capable to handle more complex objects generally in the n-dimensional space, i.e. oriented surfaces, volumes, etc.

Also, it is possible to use the *'functional analysis* approach: "Let L is a linear operator, then the following operation is valid....". As there are many linear operators like derivation, integration, Laplace transform, etc., there is a huge potential of applications of that to the formal solution of the linear system of equations.

However, it is necessary to respect, that in the case of projective representation specific care is to be taken for deriving the rules for derivation, etc., as actually, a fraction is to be processed.

It should be noted, that a solution is in symbolic representation, too, which can be used for further processing and symbolic manipulation.

3 Matrix and Linear Systems Conditionality

Both types of the linear systems of equations, i.e. $\mathbf{Ax} = \mathbf{b}$ ($\mathbf{A}$ is $n \times n$) and $\mathbf{Ax} = \mathbf{0}$ ($\mathbf{A}$ is $(n+1) \times n$), actually have the same form $\mathbf{Ax} = \mathbf{0}$ ($\mathbf{A}$ is $(n+1) \times n$), now, if the projective

representation is used. Therefore, it is possible to show the differences between the matrix conditionality and conditionality (solvability) of a linear system of equations, see Fig. 3.

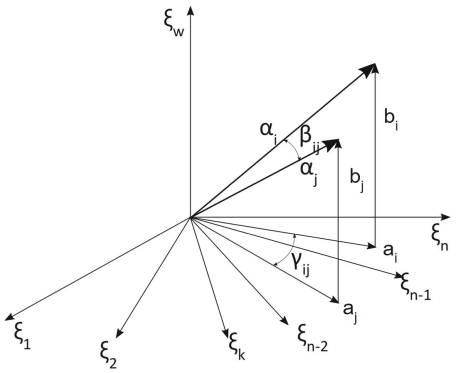

Fig. 3. Difference between matrix and linear system conditionality

For the matrix conditionality, the eigenvalues are usually used [10,11] and the ratio $rat_\lambda = |\lambda_{max}|/|\lambda_{min}|$ is mostly used as a criterion. It should be noted, that $\lambda \in C$, i.e the eigenvalues are complex, in general. If the ration rat_λ is high, the matrix is said to be ill-conditioned [4,6], especially in the case of large data with a large span of data [22,23]. There are two cases, which are needed to be taken into consideration:

– non-homogeneous systems of linear equations, i.e. $\mathbf{Ax} = \mathbf{b}$. In this case, the matrix conditionality is considered as a criterion for the solvability of the linear system of equations. It depends on the matrix **A** properties, i.e. on eigenvalues.

$$\begin{bmatrix} 10^2 & 0 & 0 \\ 0 & 10^0 & 0 \\ 0 & 0 & 10^{-2} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_3 \end{bmatrix} \tag{16}$$

A conditionality number $\kappa(\mathbf{A}) = |\lambda_{max}|/|\lambda_{min}|$ is usually used as the solvability criterion. In the case of the Eq. 16, the matrix conditionality is $\kappa(\mathbf{A}) = 10^2/10^{-2} = 10^4$. However, if the 1st row is multiplied by 10^{-2} and the 3rd row is multiplied by 10^2, then the conditionality is $\kappa(\mathbf{A}) = 1$.

Therefore, the estimation of the matrix conditionality using ratio of eigenvalues is not applicable. It should be noted, that the right side vector **b** is to be appropriately multiplied.

Classification of possible conditionality evaluation:

- *volumetric* based - as a matrix conditionality the value of $det(\mathbf{A})$ is taken. However, its value is given by a volume of a hyper-polytope (n-dimensional parallelepiped) given by the rows of the matrix $\mathbf{A}$. The "standard" matrix conditionality using eigenvectors is not quite appropriate, as shown above, Eq. 16.
- *area based* - the area of bivectors defined as $||\mathbf{a}_i \wedge \mathbf{a}_j||$. This evaluation was recently used for higher size of the Hilbert matrix is [19,20]. However, the results were not satisfactory in some cases.
- *angular* - the proposed matrix conditionality, which might be estimated using angles of bivectors formed by the rows i and j of the matrix $\mathbf{A}$. This approach reflects angular relations of all bivectors of the matrix $\mathbf{A}$ or ratios of angles $\gamma_{i,j}$, where $i = 1,...,n-1, j = k+1,...,n, i < j$, see Fig. 3. The number of all bivectors is given as $n(n-1)/2$.

It should be noted, that the matrix conditionality does not consider any influence on the values of the vector $\mathbf{b}$ of the system of linear equations $\mathbf{Ax} = \mathbf{b}$. The values of the vector $\mathbf{b}$ might have a significant influence to the solvability of the linear system of equations, see Fig. 3. Also, the values of the vector $\mathbf{b}$ might have a significant influence to the numerical precision.

– a homogeneous system of equations $\mathbf{Ax} = \mathbf{0}$, when the system of linear equations $\mathbf{Ax} = \mathbf{b}$ is expressed in the projective space. In this case, the vector $\mathbf{b}$ is taken into account and bivector area and bivector angles properties can be used for solvability evaluation. The standard *volumetric* evaluation cannot be used, as the matrix $\mathbf{A}$ is not a square matrix.

4 Basic Operations with Matrices

For a solution of a linear system of equations some specific operations are used, e.g.. row multiplication by a non-zero constant, which does not have any influence to the result correctness. Let us have a look at such operations more deeply. The basic operations can be split into two types:

– *row multiplication* by a constant $p \neq 0$. It changes the row values, but does not change the result of the linear system $\mathbf{Ax} = \mathbf{0}$, resp. $\mathbf{Ax} = \mathbf{b}$, if the vector $\mathbf{b}$ is multiplied appropriately.
– *column multiplication* - by a constant $s \neq 0$. This operation is not usually mentioned in numerical methods. It means, that there is a scaling of the relevant element of the vector $\mathbf{x}$ of the system $\mathbf{Ax} = \mathbf{0}$, resp. $\mathbf{Ax} = \mathbf{b}$.

It means, that those two main operations with both linear system of equations $\mathbf{Ax} = \mathbf{0}$, resp. $\mathbf{Ax} = \mathbf{b}$ can be described as

$$\mathbf{PAS}\,\mathbf{S}^{-1}\mathbf{x} = \mathbf{0}, \quad \mathbf{Ax} = \mathbf{0}, \quad \mathbf{A}\ \ n \times (n+1) \tag{17}$$

resp.

$$\mathbf{PAS}\,\mathbf{S}^{-1}\mathbf{x} = \mathbf{Pb}, \quad \mathbf{Ax} = \mathbf{b}, \quad \mathbf{A}\ \ n \times n \tag{18}$$

where: $\mathbf{P}$, resp. $\mathbf{S}$ are diagonal matrices with $p_i \neq 0$, $s_j \neq 0$ multiplicative constants. The matrix $\mathbf{S}$ represents an-isotropic scaling of the size $(n+1) \times (n+1)$ in the first

case, resp. $n \times n$ in the second one. It should be noted, that the diagonal matrix $\mathbf{P}$ does not have any influence on the result, due to the projective representation, and because $\mathbf{S}\,\mathbf{S}^{-1}$ is the identity matrix the solution of the linear system of equations is correct in both cases.

It can be easily proved that both operations *do have* an influence on eigenvalues and therefore $\kappa(\mathbf{A}) = |\lambda_{max}|/|\lambda_{min}|$ cannot be used for assessment of the solvability of the linear system of equations. For the same reasons, also the assessment based of the area on bivectors cannot be used.

4.1 Angular Criterion

The only *angular* criterion is invariant to the row multiplications, while only the column multiplication changes angles of the bivectors. There are several significant consequences for the numerical solution of linear systems of equations:

- the solvability of a linear system of equations can be improved by the column multiplications, only, if unlimited precision is considered. Therefore, the matrix-based pre-conditioners might not solve the solvability problems and might introduce additional numerical problems.
- the precision of computation is significantly influenced by addition and subtraction operations, as the exponents must be the same for those operations with mantissa. Also, the multiplication and division operations using exponent change by $2^{\pm k}$ should be preferred.

However, the question is, whether the above presented operations cannot be efficiently used for better numerical stability of numerical solutions. In the matrix operations, it is necessary to respect the floating-point representation [27] as it is a crucial factor in the solution of large or ill-conditioned linear systems of equations.

5 Preconditioning

There are several methods used to improve the ratio $\kappa(\mathbf{A}) = |\lambda_{max}|/|\lambda_{min}|$ of the matrix $\mathbf{A}$ of the linear system, e.g.. matrix eigenvalues shifting or preconditioning [1,2]. The preconditioning is usually based on solving a linear system $\mathbf{Ax} = \mathbf{0}$:

$$\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b} \tag{19}$$

where $\mathbf{M}^{-1}$ is a matrix, which can cover complicated computation, including Fourier transform. The inverse operation, i.e. $\mathbf{M}^{-1}$, is computationally very expensive as it is of $O(n^3)$ complexity. Therefore, they are not easily applicable for large systems of linear equations used nowadays. There are some methods based on incomplete factorization, etc., which might be used [11].

5.1 Simplified Preconditioning - A Proposed Approach

The proposed matrix conditionality improvement method requires only determination of the diagonal matrices values **P** and **S**, i.e. multiplicative coefficients $p_i \neq 0$, $s_j \neq 0$, which have to be optimized. This is a significant reduction of computational complexity, as it decreases the cost of finding sub-optimal p_i, s_j values.

The proposed approach was tested on the Hilbert's matrix, which is extremely ill-conditioned. The Hilbert matrix conditionality can be estimated as [5,9].

$$\kappa(\mathbf{H_n}) \simeq e^{3.5n} \tag{20}$$

where n is the size of the Hilbert matrix; for $n = 10$ the conditionality is estimated as $\kappa(\mathbf{H_n}) \simeq 1.56810^{15}$.

The Hilbert matrix **H** defined as:

$$h_{i,j} = \frac{1}{i+j-1} \tag{21}$$

where: $i, j = 1, \ldots n$.

The experimental results of the original conditionality $\kappa(\mathbf{H}_{orig})$ and conditionality using the proposed method $\kappa(\mathbf{H}_{new})$ are presented in Table 3.

5.2 Preliminary Experimental Results

To prove the proposed approach, the Hilbert matrix was taken, as it is ill-conditioned. The conditionality of the matrix $\mathbf{H}_{orig}$, modified by diagonal matrices **P** and **S**, was evaluated by the $cond(matrix)$ function in the Octave system using the simplified Monte Carlo approach.

Table 3. Conditionality of modified the Hilbert matrix: Experimental results (*with Octave warnings)

N	cond($\mathbf{H}_{orig}$)	cond($\mathbf{H}_{new}$)	N	cond($\mathbf{H}_{orig}$)	cond($\mathbf{H}_{new}$)
3	5.2406e+02	2.5523e+02	7	4.7537e+08	1.4341e+08
4	1.5514e+04	6.0076e+03	8	1.5258e+10	6.0076e+03
5	4.7661e+05	1.6099e+05	9	4.9315e+11	1.3736e+11
6	1.4951e+07	5.0947e+06	10	1.6024e+13	4.1485e+12
			20	1.6024e+13*	4.1485e+12

The experiments proved, that the conditionality cond($\mathbf{H}_{new}$) of the modified matrix using the proposed approach was decreased by more than half of the magnitude for higher values of n, see Table 3. This is consistent with the recently obtained results [16], where the inverse Hilbert matrix computation using the modified Gauss elimination without division operation was analyzed.

The Hilbert matrix conditionality improvement also improved the angular criterion based on maximizing the ratio $\kappa_{rat}(\mathbf{H})$ defined as:

$$\kappa_{rat}(\mathbf{H}) = \frac{\cos\gamma_{min}}{cos\gamma_{max}} \tag{22}$$

It says, how the angles $\cos\gamma_{ij}$, formed by the vectors $\mathbf{a}_{ij}$ of the bivectors are similar, see Fig. 3. It means, that if the ratio $\kappa_{rat}(\mathbf{A}) \simeq 1$ the angles of all bivectors are nearly equal. In the case of conditionality assessment of the linear system of equations $\mathbf{Ax} = \mathbf{0}$, the angles β_{ij}, formed by the angels α_{ij} have to be taken into account, see Fig. 3. The ratio $\kappa_{rat}(\mathbf{H})$ is then defined as:

$$\kappa_{rat}(\mathbf{H}) = \frac{\cos\beta_{min}}{cos\beta_{max}} \tag{23}$$

The advantage of the angular criterion is that it is common for the conditionality evaluation in the cases, i.e.:

- the matrix conditionality
- conditionality of the linear system of equations

It should be noted, that this conditionality assessment method gives different values of conditionality of those two different cases, as in the first case only the matrix is evaluated, while in the second one the value of the $\mathbf{b}$ in the $\mathbf{Ax} = \mathbf{b}$ is taken into account.

Table 4. Conditionality of modified the Hilbert matrix: Experimental results (*with Octave warnings)

N	3	4	5	6	7
$\kappa_{rat}(\mathbf{H}_{orig})$	0.54464	0.39282	0.31451	0.26573	0.23195
$\kappa_{rat}(\mathbf{H}_{new})$	0.98348	0.97740	0.98173	0.96283	0.87961
N	8	9	10		20
$\kappa_{rat}(\mathbf{H}_{orig})$	0.20694	0.18755	0.17199	$\cdots$	0.09917*
$\kappa_{rat}(\mathbf{H}_{new})$	0.92500	0.96435	0.96322	$\cdots$	0.74701*

The results presented in Table 4 reflects the improvement of the Hilbert matrix by proposed approach using the diagonal matrices $\mathbf{P}$ and $\mathbf{S}$ used as the multipliers.

6 Conclusion

This contribution briefly presents a new approach to the matrix conditionality and the solvability of linear systems of equations assessment, based on the bivectors angular properties based on geometry algebra. The presented approach enables to make conditionality assessment for both types of linear equations, i.e. $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{Ax} = \mathbf{0}$.

Also, the equivalence of the solution of linear systems of equations with the application of the outer product (extended cross product) has been presented. It offers simple

and efficient solutions to many computational problems, if combined with the principle of duality and projective notation. Also, the presented approach supports direct GPU application with a potential of significant speed-up and parallelism. Even more, the approach is applicable to n-dimensional problem solutions, as the geometric algebra is multidimensional.

In the future, the proposed approach will be more deeply analyzed and experimentally verified using numerical operations with values restricted to 2^k as in this case the mantissa value is not changed. The expected application is in the radial basis function interpolation and approximation for large data, when matrices are ill-conditioned, and in the solution of partial differential equations.

Acknowledgment. The author would like to thank their colleagues and students at the University of West Bohemia, Plzen, for their discussions and suggestions and implementations, especially to Michal Smolik and Martin Cervenka for discussions and help with the Octave implementation, to Filip Hacha, Vitek Poor and Jan Kasak for additional verification of some parts in the Octave system. Thanks belong also to anonymous reviewers for their valuable comments and hints provided. This research was supported by the Czech Science Foundation (GACR) No. GA 17-05534S.

Appendix A

Computation of a bivector area in the n-dimensional space can be made as follows:

$$cos(\alpha_{ij}) = \frac{\mathbf{a}_i.\mathbf{a}_j}{||\mathbf{a}_i||\ ||\mathbf{a}_j||} \qquad sin(\alpha_{ij}) = \frac{||\mathbf{a}_i \wedge \mathbf{a}_j||}{||\mathbf{a}_i||\ ||\mathbf{a}_j||} \tag{1}$$

Therefore the square of the bivector area is given as

$$||\mathbf{a}_i \wedge \mathbf{a}_j||^2 = ||\mathbf{a}_i||^2||\mathbf{a}_j||^2 sin^2(\alpha_{ij}) \tag{2}$$

As the following identity is valid

$$sin^2(\alpha_{ij}) = 1 - cos^2(\alpha_{ij}) \tag{3}$$

then it can be expressed as

$$||\mathbf{a}_i \wedge \mathbf{a}_j||^2 = ||\mathbf{a}_i||^2||\mathbf{a}_j||^2(1 - cos^2(\alpha_{ij})) \tag{4}$$

by a substitution of Eq. 1

$$||\mathbf{a}_i \wedge \mathbf{a}_j||^2 = ||\mathbf{a}_i||^2||\mathbf{a}_j||^2(1 - \frac{(\mathbf{a}_i \cdot \mathbf{a}_j)^2}{||\mathbf{a}_i||^2\ ||\mathbf{a}_j||^2}) \tag{5}$$

and algebraic manipulation

$$||\mathbf{a}_i \wedge \mathbf{a}_j||^2 = ||\mathbf{a}_i||^2||\mathbf{a}_j||^2(\frac{||\mathbf{a}_i||^2 \cdot ||\mathbf{a}_j||^2 - (\mathbf{a}_i \cdot \mathbf{a}_j)^2}{||\mathbf{a}_i||^2||\mathbf{a}_j||^2}) \tag{6}$$

i.e.

$$||\mathbf{a}_i \wedge \mathbf{a}_j||^2 = ||\mathbf{a}_i||^2||\mathbf{a}_j||^2 \frac{||\mathbf{a}_i||^2||\mathbf{a}_j||^2 - (\mathbf{a}_i \cdot \mathbf{a}_j)^2}{||\mathbf{a}_i||^2||\mathbf{a}_j||^2} \tag{7}$$

Now, reduction can be used

$$||\mathbf{a}_i \wedge \mathbf{a}_j||^2 = ||\mathbf{a}_i||^2||\mathbf{a}_j||^2 - (\mathbf{a}_i \cdot \mathbf{a}_j)^2 \tag{8}$$

and finally the square of the area of a bivector is given as:

$$||\mathbf{a}_i \wedge \mathbf{a}_j|| = \sqrt{(||\mathbf{a}_i||^2||\mathbf{a}_j||^2 - (\mathbf{a}_i \cdot \mathbf{a}_j)^2)} \tag{9}$$

References

1. Benzi, M.: Preconditioning techniques for large linear systems: a survey. J. Comput. Phys. **182**(2), 418–477 (2002)
2. Chen, K.: Matrix Preconditioning Techniques and Applications. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, New York (2005)
3. Ferronato, M.: Preconditioning for sparse linear systems at the dawn of the 21st century: history, current developments, and future perspectives. ISRN Appl. Math. **4**(ID 127647), 1–49 (2012). An optional note
4. George, A., Ikramov, K.: The conditionally and expected error of two methods of computing the pseudo-eigenvalues of a complex matrix. Comput. Math. Math. Phys. **35**(11), 1403–1408 (1995)
5. Higham, N.J.: Accuracy and Stability of Numerical Algorithms, 3rd edn. SIAM (1961)
6. Ikramov, K., Chugunov, V.: On the conditionality of eigenvalues close to the boundary of the numerical range of a matrix. Doklady Math. **57**(2), 201–202 (1998)
7. Johnson, M.: Proof by duality: or the discovery of "new" theorems. Math. Today **1**(6), 201–213 (1996). An optional note
8. Majdisova, Z., Skala, V.: Big geodata surface approximation using radial basis functions: a comparative study. Comput. Geosci. **109**, 51–58 (2017)
9. Quaerteroni, A., Saleri, F., Gervasio, P.: Scientific Computing with MATLAB and Octave, 3rd edn. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12430-3
10. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM (2011)
11. Saad, Y.: Numerical problems for Large Eigenvalue Problems, 2nd edn. SIAM (2011)
12. Skala, V.: Length, area and volume computation in homogeneous coordinates. Int. J. Image Graph. **6**(4), 625–639 (2006)
13. Skala, V.: Barycentric coordinates computation in homogeneous coordinates. Comput. Graph. (Pergamon) **32**(1), 120–127 (2008)
14. Skala, V.: Intersection computation in projective space using homogeneous coordinates. Int. J. Image Graph. **8**(4), 615–628 (2008)
15. Skala, V.: Duality, barycentric coordinates and intersection computation in projective space with GPU support. WSEAS Trans. Math. **9**(6), 407–416 (2010)
16. Skala, V.: Modified Gaussian elimination without division operations. In: ICNAAM 2013 Proceedings, AIP Conference Proceedings, vol. 1558, pp. 1936–1939. AIP (2013)
17. Skala, V.: Plücker Coordinates and Extended Cross Product for Robust and Fast Intersection Computation, pp. 57–60 (2016)
18. Skala, V.: "Extended cross-product" and solution of a linear system of equations. Lect. Not. Comput. Sci. **9786**, 18–35 (2016)
19. Skala, V.: High dimensional and large span data least square error: numerical stability and conditionality. Int. J. Appl. Phys. Math. **7**(3), 148–156 (2017)
20. Skala, V.: Least square method robustness of computations: what is not usually considered and taught. FedCSIS **2017**, 537–541 (2017)

21. Skala, V.: Projective Geometry, Duality and Plücker Coordinates for Geometric Computations with Determinants on GPUs, vol. 1863 (2017)
22. Skala, V.: RBF Interpolation with CSRBF of Large Data Sets, vol. 108, pp. 2433–2437 (2017)
23. Skala, V.: RBF Approximation of Big Data Sets with Large Span of Data, vol. 2018-January, pp. 212–218 (2018)
24. Skala, V., Ondracka, V.: A Precision of Computation in the Projective Space, pp. 35–40 (2011)
25. Strassen, V.: Gaussian elimination is not optimal. Numerische Mathematik **13**(4), 354–356 (1969)
26. Vince, J.: Geometric Algebra for Computer Graphics. Springer, London (2008). https://doi.org/10.1007/978-1-84628-997-2
27. |WEB: IEEE 754–2019 - IEEE standard for floating-point arithmetic (2019)
28. Yamazaki, I., Ida, A., Yokota, R., Dongarra, J.: Distributed-memory lattice h-matrix factorization. Int. J. High Perform. Comput. Appl. **33**(5), 1046–1063 (2019)