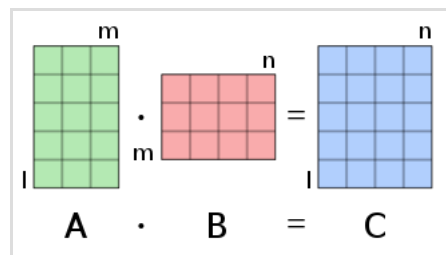# WIKIPEDIA
The Free Encyclopedia

Matrix multiplication

In [mathematics](#), particularly in [linear algebra](#), **matrix multiplication** is a [binary operation](#) that produces a [matrix](#) from two matrices. For matrix multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix. The resulting matrix, known as the **matrix product**, has the number of rows of the first and the number of columns of the second matrix. The product of matrices **A** and **B** is denoted as **AB**.[1]

Matrix multiplication was first described by the French mathematician [Jacques Philippe Marie Binet](#) in 1812,[2] to represent the [composition](#) of [linear maps](#) that are represented by matrices. Matrix multiplication is thus a basic tool of [linear algebra](#), and as such has numerous applications in many areas of mathematics, as well as in

For matrix multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix. The result matrix has the number of rows of the first and the number of columns of the second matrix.

[applied mathematics](#), [statistics](#), [physics](#), [economics](#), and [engineering](#).[3][4] Computing matrix products is a central operation in all computational applications of linear algebra.

Notation

This article will use the following notational conventions: matrices are represented by capital letters in bold, e.g. **A**; [vectors](#) in lowercase bold, e.g. **a**; and entries of vectors and matrices are italic (they are numbers from a field), e.g. A and a. [Index notation](#) is often the clearest way to express definitions, and is used as standard in the literature. The entry in row i, column j of matrix **A** is indicated by $(\mathbf{A})_{ij}$, A_{ij} or a_{ij}. In contrast, a single subscript, e.g. $\mathbf{A}_1$, $\mathbf{A}_2$, is used to select a matrix (not a matrix entry) from a collection of matrices.

Definitions

Matrix times matrix

If **A** is an $m \times n$ matrix and **B** is an $n \times p$ matrix,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix}$$

the *matrix product* **C** = **AB** (denoted without multiplication signs or dots) is defined to be the $m \times p$ matrix[5][6][7][8]

$$\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ & & & \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{pmatrix}$$

such that

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj} = \sum_{k=1}^{n} a_{ik}b_{kj},$$

for $i = 1, ..., m$ and $j = 1, ..., p$.

That is, the entry c_{ij} of the product is obtained by multiplying term-by-term the entries of the ith row of $\mathbf{A}$ and the jth column of $\mathbf{B}$, and summing these n products. In other words, c_{ij} is the dot product of the ith row of $\mathbf{A}$ and the jth column of $\mathbf{B}$.

Therefore, $\mathbf{AB}$ can also be written as

$$\mathbf{C} = \begin{pmatrix} a_{11}b_{11} + \cdots + a_{1n}b_{n1} & a_{11}b_{12} + \cdots + a_{1n}b_{n2} & \cdots & a_{11}b_{1p} + \cdots + a_{1n}b_{np} \\ a_{21}b_{11} + \cdots + a_{2n}b_{n1} & a_{21}b_{12} + \cdots + a_{2n}b_{n2} & \cdots & a_{21}b_{1p} + \cdots + a_{2n}b_{np} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{11} + \cdots + a_{mn}b_{n1} & a_{m1}b_{12} + \cdots + a_{mn}b_{n2} & \cdots & a_{m1}b_{1p} + \cdots + a_{mn}b_{np} \end{pmatrix}$$

Thus the product $\mathbf{AB}$ is defined if and only if the number of columns in $\mathbf{A}$ equals the number of rows in $\mathbf{B}$,[1] in this case n.

In most scenarios, the entries are numbers, but they may be any kind of mathematical objects for which an addition and a multiplication are defined, that are associative, and such that the addition is commutative, and the multiplication is distributive with respect to the addition. In particular, the entries may be matrices themselves (see block matrix).

Matrix times vector

A vector $\mathbf{x}$ of length n can be viewed as a column vector, corresponding to an $n \times 1$ matrix $\mathbf{X}$ whose entries are given by $\mathbf{X}_{i1} = \mathbf{x}_i$. If $\mathbf{A}$ is an $m \times n$ matrix, the matrix-times-vector product denoted by $\mathbf{Ax}$ is then the vector $\mathbf{y}$ that, viewed as a column vector, is equal to the $m \times 1$ matrix $\mathbf{AX}$. In index notation, this amounts to:

$$y_i = \sum_{j=1}^{n} a_{ij}x_j.$$

One way of looking at this is that the changes from "plain" vector to column vector and back are assumed and left implicit.

Vector times matrix

Similarly, a vector $\mathbf{x}$ of length n can be viewed as a row vector, corresponding to a $1 \times n$ matrix. To make it clear that a row vector is meant, it is customary in this context to represent it as the

transpose of a column vector; thus, one will see notations such as $\mathbf{x}^\mathbf{T}\mathbf{A}$. The identity $\mathbf{x}^\mathbf{T}\mathbf{A} = (\mathbf{A}^\mathbf{T}\mathbf{x})^\mathbf{T}$ holds. In index notation, if $\mathbf{A}$ is an $n \times p$ matrix, $\mathbf{x}^\mathbf{T}\mathbf{A} = \mathbf{y}^\mathbf{T}$ amounts to:

$$y_k = \sum_{j=1}^{n} x_j a_{jk}.$$

Vector times vector

The dot product $\mathbf{a} \cdot \mathbf{b}$ of two vectors $\mathbf{a}$ and $\mathbf{b}$ of equal length is equal to the single entry of the 1×1 matrix resulting from multiplying these vectors as a row and a column vector, thus: $\mathbf{a}^\mathbf{T}\mathbf{b}$ (or $\mathbf{b}^\mathbf{T}\mathbf{a},$ which results in the same 1×1 matrix).

Illustration

The figure to the right illustrates diagrammatically the product of two matrices $\mathbf{A}$ and $\mathbf{B}$, showing how each intersection in the product matrix corresponds to a row of $\mathbf{A}$ and a column of $\mathbf{B}$.

$$
\overset{4\times 2 \text{ matrix}}{\begin{bmatrix} a_{11} & a_{12} \\ \cdot & \cdot \\ a_{31} & a_{32} \\ \cdot & \cdot \end{bmatrix}}
\overset{2\times 3 \text{ matrix}}{\begin{bmatrix} \cdot & b_{12} & b_{13} \\ \cdot & b_{22} & b_{23} \end{bmatrix}}
=
\overset{4\times 3 \text{ matrix}}{\begin{bmatrix} \cdot & c_{12} & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & c_{33} \\ \cdot & \cdot & \cdot \end{bmatrix}}
$$
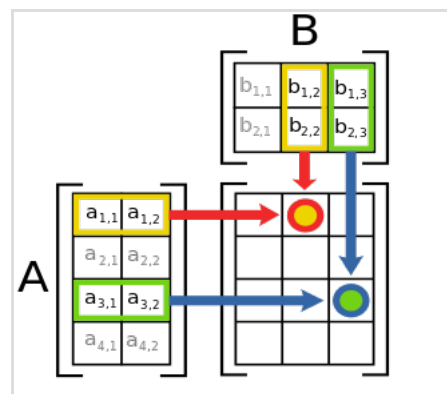
The values at the intersections, marked with circles in figure to the right, are:

$$c_{12} = a_{11}b_{12} + a_{12}b_{22}$$
$$c_{33} = a_{31}b_{13} + a_{32}b_{23}.$$

Fundamental applications

Historically, matrix multiplication has been introduced for facilitating and clarifying computations in linear algebra. This strong relationship between matrix multiplication and linear algebra remains fundamental in all mathematics, as well as in physics, chemistry, engineering and computer science.

Linear maps

If a vector space has a finite basis, its vectors are each uniquely represented by a finite sequence of scalars, called a coordinate vector, whose elements are the coordinates of the vector on the basis. These coordinate vectors form another vector space, which is isomorphic to the original vector space. A coordinate vector is commonly organized as a column matrix (also called a *column vector*), which is a matrix with only one column. So, a column vector represents both a coordinate vector, and a vector of the original vector space.

A linear map A from a vector space of dimension n into a vector space of dimension m maps a

column vector

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

onto the column vector

$$\mathbf{y} = A(\mathbf{x}) = \begin{pmatrix} a_{11}x_1 + \cdots + a_{1n}x_n \\ a_{21}x_1 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n \end{pmatrix}.$$

The linear map A is thus defined by the matrix

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

and maps the column vector $\mathbf{x}$ to the matrix product

$$\mathbf{y} = \mathbf{A}\mathbf{x}.$$

If B is another linear map from the preceding vector space of dimension m, into a vector space of dimension p, it is represented by a $p \times m$ matrix $\mathbf{B}$. A straightforward computation shows that the matrix of the composite map $B \circ A$ is the matrix product $\mathbf{BA}$. The general formula $(B \circ A)(\mathbf{x}) = B(A(\mathbf{x}))$ that defines the function composition is instanced here as a specific case of associativity of matrix product (see § Associativity below):

$$(\mathbf{BA})\mathbf{x} = \mathbf{B}(\mathbf{A}\mathbf{x}) = \mathbf{BA}\mathbf{x}.$$

Geometric rotations

Using a Cartesian coordinate system in a Euclidean plane, the rotation by an angle α around the origin is a linear map. More precisely,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

where the source point (x, y) and its image (x', y') are written as column vectors.

The composition of the rotation by α and that by β then corresponds to the matrix product

$$\begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} = \begin{bmatrix} \cos\beta\cos\alpha - \sin\beta\sin\alpha & -\cos\beta\sin\alpha - \sin\beta \\ \sin\beta\cos\alpha + \cos\beta\sin\alpha & -\sin\beta\sin\alpha + \cos\beta \end{bmatrix}$$

where appropriate underlined trigonometric identities are employed for the second equality. That is, the composition corresponds to the rotation by angle $\alpha + \beta$, as expected.

Resource allocation in economics

As an example, a fictitious factory uses 4 kinds of basic commodities, b_1, b_2, b_3, b_4 to produce 3 kinds of intermediate goods, m_1, m_2, m_3, which in turn are used to produce 3 kinds of final products, f_1, f_2, f_3. The matrices

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} \quad \text{and}$$

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 3 & 1 \\ 4 & 2 & 2 \end{pmatrix}$$

provide the amount of basic commodities needed for a given

$$\begin{pmatrix} 1 & 0 & 1 \\ 2 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 1 \\ 2 & 3 & 1 \\ 4 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 5 & 4 & 3 \\ 8 & 9 & 5 \\ 6 & 5 & 3 \\ 11 & 9 & 6 \end{pmatrix}$$

The computation of the bottom left entry of $\mathbf{AB}$ corresponds to the consideration of all paths (highlighted) from basic commodity b_4 to final product f_1 in the production flow graph.

amount of intermediate goods, and the amount of intermediate goods needed for a given amount of final products, respectively. For example, to produce one unit of intermediate good m_1, one unit of basic commodity b_1, two units of b_2, no units of b_3, and one unit of b_4 are needed, corresponding to the first column of $\mathbf{A}$.
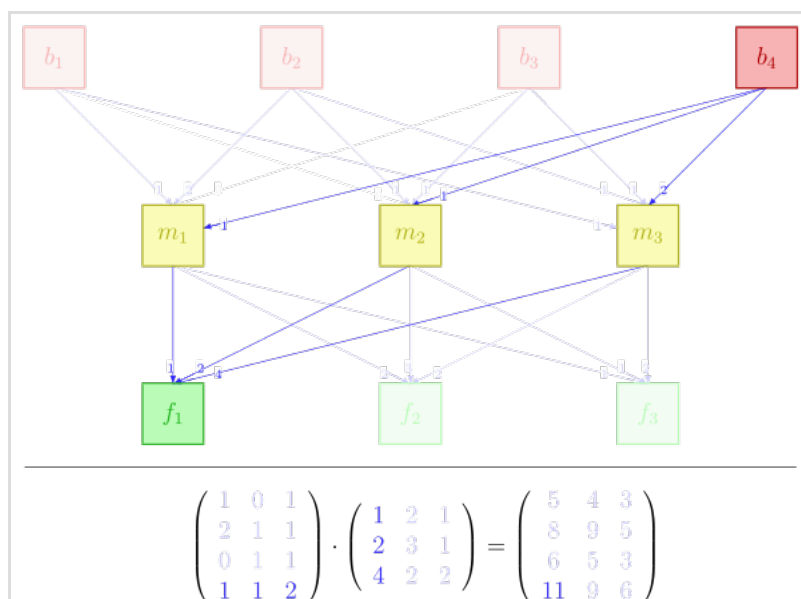
Using matrix multiplication, compute

$$\mathbf{AB} = \begin{pmatrix} 5 & 4 & 3 \\ 8 & 9 & 5 \\ 6 & 5 & 3 \\ 11 & 9 & 6 \end{pmatrix};$$

this matrix directly provides the amounts of basic commodities needed for given amounts of final goods. For example, the bottom left entry of $\mathbf{AB}$ is computed as $1 \cdot 1 + 1 \cdot 2 + 2 \cdot 4 = 11$, reflecting that 11 units of b_4 are needed to produce one unit of f_1. Indeed, one b_4 unit is needed for m_1, one for each of two m_2, and 2 for each of the four m_3 units that go into the f_1 unit, see picture.

In order to produce e.g. 100 units of the final product f_1, 80 units of f_2, and 60 units of f_3, the necessary amounts of basic goods can be computed as

$$(\mathbf{AB}) \begin{pmatrix} 100 \\ 80 \\ 60 \end{pmatrix} = \begin{pmatrix} 1000 \\ 1820 \\ 1180 \\ 2180 \end{pmatrix},$$

that is, 1000 units of b_1, 1820 units of b_2, 1180 units of b_3, 2180 units of b_4 are needed. Similarly, the product matrix $\mathbf{AB}$ can be used to compute the needed amounts of basic goods for other final-good amount data.[9]

System of linear equations

The general form of a [system of linear equations](link) is

$$a_{11}x_1 + \cdots + a_{1n}x_n = b_1,$$
$$a_{21}x_1 + \cdots + a_{2n}x_n = b_2,$$
$$\vdots$$
$$a_{m1}x_1 + \cdots + a_{mn}x_n = b_m.$$

Using same notation as above, such a system is equivalent with the single matrix [equation](link)

$$\mathbf{Ax} = \mathbf{b}.$$

Dot product, bilinear form and sesquilinear form

The [dot product](link) of two column vectors is the unique entry of the matrix product

$$\mathbf{x}^{\mathsf{T}}\mathbf{y},$$

where $\mathbf{x}^{\mathsf{T}}$ is the [row vector](link) obtained by [transposing](link) $\mathbf{x}$. (As usual, a $1{\times}1$ matrix is identified with its unique entry.)

More generally, any [bilinear form](link) over a vector space of finite dimension may be expressed as a matrix product

$$\mathbf{x}^{\mathsf{T}}\mathbf{Ay},$$

and any [sesquilinear form](link) may be expressed as

$$\mathbf{x}^{\dagger}\mathbf{Ay},$$

where $\mathbf{x}^{\dagger}$ denotes the [conjugate transpose](link) of $\mathbf{x}$ (conjugate of the transpose, or equivalently transpose of the conjugate).

General properties

Matrix multiplication shares some properties with usual [multiplication](link). However, matrix multiplication is not defined if the number of columns of the first factor differs from the number of rows of the second factor, and it is [non-commutative](link),[10] even when the product remains defined after changing the order of the factors.[11][12]

Non-commutativity

An operation is [commutative](link) if, given two elements $\mathbf{A}$ and $\mathbf{B}$ such that the product $\mathbf{AB}$ is defined, then $\mathbf{BA}$ is also defined, and $\mathbf{AB} = \mathbf{BA}$.

If $\mathbf{A}$ and $\mathbf{B}$ are matrices of respective sizes $m \times n$ and $p \times q$, then $\mathbf{AB}$ is defined if $n = p$, and $\mathbf{BA}$ is defined if $m = q$. Therefore, if one of the products is defined, the other one need not be defined. If $m = q \neq n = p$, the two products are defined, but have different sizes; thus they cannot be equal. Only if $m = q = n = p$, that is, if $\mathbf{A}$ and $\mathbf{B}$ are square matrices of the same size, are both products defined and of the same size. Even in this case, one has in general

$$\mathbf{AB} \neq \mathbf{BA}.$$

For example

$$\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},$$

but

$$\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

This example may be expanded for showing that, if $\mathbf{A}$ is a $n \times n$ matrix with entries in a field F, then $\mathbf{AB} = \mathbf{BA}$ for every $n \times n$ matrix $\mathbf{B}$ with entries in F, if and only if $\mathbf{A} = c\mathbf{I}$ where $c \in F$, and $\mathbf{I}$ is the $n \times n$ identity matrix. If, instead of a field, the entries are supposed to belong to a ring, then one must add the condition that c belongs to the center of the ring.

One special case where commutativity does occur is when $\mathbf{D}$ and $\mathbf{E}$ are two (square) diagonal matrices (of the same size); then $\mathbf{DE} = \mathbf{ED}$.[10] Again, if the matrices are over a general ring rather than a field, the corresponding entries in each must also commute with each other for this to hold.

Distributivity

The matrix product is distributive with respect to matrix addition. That is, if $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ are matrices of respective sizes $m \times n$, $n \times p$, $n \times p$, and $p \times q$, one has (left distributivity)

$$\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC},$$

and (right distributivity)

$$(\mathbf{B} + \mathbf{C})\mathbf{D} = \mathbf{BD} + \mathbf{CD}.[10]$$

This results from the distributivity for coefficients by

$$\sum_k a_{ik}(b_{kj} + c_{kj}) = \sum_k a_{ik}b_{kj} + \sum_k a_{ik}c_{kj}$$
$$\sum_k (b_{ik} + c_{ik})d_{kj} = \sum_k b_{ik}d_{kj} + \sum_k c_{ik}d_{kj}.$$

Product with a scalar

If $\mathbf{A}$ is a matrix and c a scalar, then the matrices $c\mathbf{A}$ and $\mathbf{A}c$ are obtained by left or right multiplying all entries of $\mathbf{A}$ by c. If the scalars have the commutative property, then $c\mathbf{A} = \mathbf{A}c$.

If the product $\mathbf{AB}$ is defined (that is, the number of columns of $\mathbf{A}$ equals the number of rows of $\mathbf{B}$), then

$$c(\mathbf{AB}) = (c\mathbf{A})\mathbf{B} \text{ and } (\mathbf{AB})c = \mathbf{A}(\mathbf{B}c).$$

If the scalars have the commutative property, then all four matrices are equal. More generally, all four are equal if c belongs to the center of a ring containing the entries of the matrices, because in this case, $c\mathbf{X} = \mathbf{X}c$ for all matrices $\mathbf{X}$.

These properties result from the bilinearity of the product of scalars:

$$c\left(\sum_k a_{ik}b_{kj}\right) = \sum_k (ca_{ik})b_{kj}$$

$$\left(\sum_k a_{ik}b_{kj}\right) c = \sum_k a_{ik}(b_{kj}c).$$

Transpose

If the scalars have the commutative property, the transpose of a product of matrices is the product, in the reverse order, of the transposes of the factors. That is

$$(\mathbf{AB})^{\mathsf{T}} = \mathbf{B}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}$$

where $^{\mathsf{T}}$ denotes the transpose, that is the interchange of rows and columns.

This identity does not hold for noncommutative entries, since the order between the entries of $\mathbf{A}$ and $\mathbf{B}$ is reversed, when one expands the definition of the matrix product.

Complex conjugate

If $\mathbf{A}$ and $\mathbf{B}$ have complex entries, then

$$(\mathbf{AB})^* = \mathbf{A}^*\mathbf{B}^*$$

where * denotes the entry-wise complex conjugate of a matrix.

This results from applying to the definition of matrix product the fact that the conjugate of a sum is the sum of the conjugates of the summands and the conjugate of a product is the product of the conjugates of the factors.

Transposition acts on the indices of the entries, while conjugation acts independently on the entries themselves. It results that, if $\mathbf{A}$ and $\mathbf{B}$ have complex entries, one has

$$(\mathbf{AB})^{\dagger} = \mathbf{B}^{\dagger}\mathbf{A}^{\dagger},$$

where † denotes the conjugate transpose (conjugate of the transpose, or equivalently transpose of the conjugate).

Associativity

Given three matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, the products $(\mathbf{AB})\mathbf{C}$ and $\mathbf{A}(\mathbf{BC})$ are defined if and only if the number of columns of $\mathbf{A}$ equals the number of rows of $\mathbf{B}$, and the number of columns of $\mathbf{B}$ equals the number of rows of $\mathbf{C}$ (in particular, if one of the products is defined, then the other is also defined). In this case, one has the associative property

$$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC}).$$

As for any associative operation, this allows omitting parentheses, and writing the above products as $\mathbf{ABC}$.

This extends naturally to the product of any number of matrices provided that the dimensions match. That is, if $\mathbf{A}_1$, $\mathbf{A}_2$, ..., $\mathbf{A}_n$ are matrices such that the number of columns of $\mathbf{A}_i$ equals the number of rows of $\mathbf{A}_{i+1}$ for $i = 1, ..., n - 1$, then the product

$$\prod_{i=1}^{n} \mathbf{A}_i = \mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_n$$

is defined and does not depend on the order of the multiplications, if the order of the matrices is kept fixed.

These properties may be proved by straightforward but complicated summation manipulations. This result also follows from the fact that matrices represent linear maps. Therefore, the associative property of matrices is simply a specific case of the associative property of function composition.

Computational complexity depends on parenthezation

Although the result of a sequence of matrix products does not depend on the order of operation (provided that the order of the matrices is not changed), the computational complexity may depend dramatically on this order.

For example, if $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are matrices of respective sizes 10×30, 30×5, 5×60, computing $(\mathbf{AB})\mathbf{C}$ needs 10×30×5 + 10×5×60 = 4,500 multiplications, while computing $\mathbf{A}(\mathbf{BC})$ needs 30×5×60 + 10×30×60 = 27,000 multiplications.

Algorithms have been designed for choosing the best order of products, see Matrix chain multiplication. When the number n of matrices increases, it has been shown that the choice of the best order has a complexity of $O(n \log n)$.

Application to similarity

Any invertible matrix $\mathbf{P}$ defines a similarity transformation (on square matrices of the same size as $\mathbf{P}$)

$$S_{\mathbf{P}}(\mathbf{A}) = \mathbf{P}^{-1} \mathbf{A} \mathbf{P}.$$

Similarity transformations map product to products, that is

$$S_{\mathbf{P}}(\mathbf{AB}) = S_{\mathbf{P}}(\mathbf{A}) S_{\mathbf{P}}(\mathbf{B}).$$

In fact, one has

$$\mathbf{P}^{-1}(\mathbf{AB})\mathbf{P} = \mathbf{P}^{-1}\mathbf{A}(\mathbf{PP}^{-1})\mathbf{BP} = (\mathbf{P}^{-1}\mathbf{AP})(\mathbf{P}^{-1}\mathbf{BP}).$$

Square matrices

Let us denote $\mathcal{M}_n(R)$ the set of $n{\times}n$ <u>square matrices</u> with entries in a <u>ring</u> R, which, in practice, is often a <u>field</u>.

In $\mathcal{M}_n(R)$, the product is defined for every pair of matrices. This makes $\mathcal{M}_n(R)$ a <u>ring</u>, which has the <u>identity matrix</u> $\mathbf{I}$ as <u>identity element</u> (the matrix whose diagonal entries are equal to 1 and all other entries are 0). This ring is also an <u>associative R-algebra</u>.

If $n > 1$, many matrices do not have a <u>multiplicative inverse</u>. For example, a matrix such that all entries of a row (or a column) are 0 does not have an inverse. If it exists, the inverse of a matrix $\mathbf{A}$ is denoted $\mathbf{A}^{-1}$, and, thus verifies

$$\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}.$$

A matrix that has an inverse is an <u>invertible matrix</u>. Otherwise, it is a <u>singular matrix</u>.

A product of matrices is invertible if and only if each factor is invertible. In this case, one has

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}.$$

When R is <u>commutative</u>, and, in particular, when it is a field, the <u>determinant</u> of a product is the product of the determinants. As determinants are scalars, and scalars commute, one has thus

$$\det(\mathbf{AB}) = \det(\mathbf{BA}) = \det(\mathbf{A})\det(\mathbf{B}).$$

The other matrix <u>invariants</u> do not behave as well with products. Nevertheless, if R is commutative, $\mathbf{AB}$ and $\mathbf{BA}$ have the same <u>trace</u>, the same <u>characteristic polynomial</u>, and the same <u>eigenvalues</u> with the same multiplicities. However, the <u>eigenvectors</u> are generally different if $\mathbf{AB} \neq \mathbf{BA}$.

Powers of a matrix

One may raise a square matrix to any <u>nonnegative integer power</u> multiplying it by itself repeatedly in the same way as for ordinary numbers. That is,

$$\begin{aligned}
\mathbf{A}^0 &= \mathbf{I}, \\
\mathbf{A}^1 &= \mathbf{A}, \\
\mathbf{A}^k &= \underbrace{\mathbf{AA}\cdots\mathbf{A}}_{k \text{ times}}.
\end{aligned}$$

Computing the kth power of a matrix needs $k - 1$ times the time of a single matrix multiplication, if it is done with the trivial algorithm (repeated multiplication). As this may be very time consuming, one generally prefers using <u>exponentiation by squaring</u>, which requires less than $2\log_2 k$ matrix multiplications, and is therefore much more efficient.

An easy case for exponentiation is that of a <u>diagonal matrix</u>. Since the product of diagonal matrices amounts to simply multiplying corresponding diagonal elements together, the kth power of a diagonal matrix is obtained by raising the entries to the power k:

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}^k = \begin{bmatrix} a_{11}^k & 0 & \cdots & 0 \\ 0 & a_{22}^k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn}^k \end{bmatrix}.$$
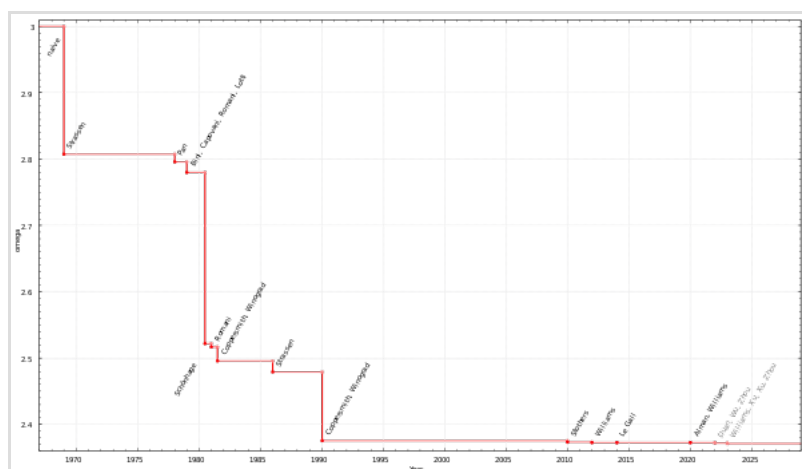
Abstract algebra

The definition of matrix product requires that the entries belong to a semiring, and does not require multiplication of elements of the semiring to be <u>commutative</u>. In many applications, the matrix elements belong to a field, although the <u>tropical semiring</u> is also a common choice for graph <u>shortest path</u> problems.[13] Even in the case of matrices over fields, the product is not commutative in general, although it is <u>associative</u> and is <u>distributive</u> over <u>matrix addition</u>. The <u>identity matrices</u> (which are the <u>square matrices</u> whose entries are zero outside of the main diagonal and 1 on the main diagonal) are <u>identity elements</u> of the matrix product. It follows that the $n \times n$ matrices over a <u>ring</u> form a ring, which is noncommutative except if $n = 1$ and the ground ring is commutative.

A square matrix may have a <u>multiplicative inverse</u>, called an <u>inverse matrix</u>. In the common case where the entries belong to a <u>commutative ring</u> R, a matrix has an inverse if and only if its <u>determinant</u> has a multiplicative inverse in R. The determinant of a product of square matrices is the product of the determinants of the factors. The $n \times n$ matrices that have an inverse form a <u>group</u> under matrix multiplication, the <u>subgroups</u> of which are called <u>matrix groups</u>. Many classical groups (including all <u>finite groups</u>) are <u>isomorphic</u> to matrix groups; this is the starting point of the theory of <u>group representations</u>.

Computational complexity

The matrix multiplication <u>algorithm</u> that results from the definition requires, in the <u>worst case</u>, n^3 multiplications and $(n-1)n^2$ additions of scalars to compute the product of two square $n \times n$ matrices. Its <u>computational complexity</u> is therefore $O(n^3)$, in a <u>model of computation</u> for which the scalar operations take constant time.

Improvement of estimates of exponent ω over time for the computational complexity of matrix multiplication $O(n^\omega)$

Rather surprisingly, this complexity is not optimal, as shown in 1969 by Volker Strassen, who provided an algorithm, now called <u>Strassen's algorithm</u>, with a complexity of

$O(n^{\log_2 7}) \approx O(n^{2.8074})$.[14] Strassen's algorithm can be parallelized to further improve the performance.[15] As of January 2024, the best peer-reviewed matrix multiplication algorithm is by Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou and has complexity $O(n^{2.371552})$.[16][17] It is not known whether matrix multiplication can be performed in $n^{2 + o(1)}$ time.[18] This would be optimal, since one must read the n^2 elements of a matrix in order to multiply it with another matrix.

Since matrix multiplication forms the basis for many algorithms, and many operations on matrices even have the same complexity as matrix multiplication (up to a multiplicative constant), the computational complexity of matrix multiplication appears throughout numerical linear algebra and theoretical computer science.

Generalizations

Other types of products of matrices include:

- Block matrix operations
- Cracovian product, defined as $\mathbf{A} \wedge \mathbf{B} = \mathbf{B}^{\mathrm{T}} \mathbf{A}$
- Frobenius inner product, the dot product of matrices considered as vectors, or, equivalently the sum of the entries of the Hadamard product
- Hadamard product of two matrices of the same size, resulting in a matrix of the same size, which is the product entry-by-entry
- Kronecker product or tensor product, the generalization to any size of the preceding
- Khatri-Rao product and Face-splitting product
- Outer product, also called dyadic product or tensor product of two column matrices, which is $\mathbf{ab}^{\mathsf{T}}$
- Scalar multiplication

See also

- Matrix calculus, for the interaction of matrix multiplication with operations from calculus

Notes

1. Nykamp, Duane. "Multiplying matrices and vectors" (https://mathinsight.org/matrix_vector_m ultiplication). *Math Insight*. Retrieved September 6, 2020.
2. O'Connor, John J.; Robertson, Edmund F., "Jacques Philippe Marie Binet" (https://mathshist ory.st-andrews.ac.uk/Biographies/Binet.html), *MacTutor History of Mathematics Archive*, University of St Andrews
3. Lerner, R. G.; Trigg, G. L. (1991). *Encyclopaedia of Physics* (2nd ed.). VHC publishers. ISBN 978-3-527-26954-9.
4. Parker, C. B. (1994). *McGraw Hill Encyclopaedia of Physics* (https://archive.org/details/mcgr awhillencycl1993park) (2nd ed.). McGraw-Hill. ISBN 978-0-07-051400-3.
5. Lipschutz, S.; Lipson, M. (2009). *Linear Algebra*. Schaum's Outlines (4th ed.). McGraw Hill (USA). pp. 30–31. ISBN 978-0-07-154352-1.
6. Riley, K. F.; Hobson, M. P.; Bence, S. J. (2010). *Mathematical methods for physics and engineering* (https://archive.org/details/mathematicalmeth00rile). Cambridge University Press. ISBN 978-0-521-86153-3.

7. Adams, R. A. (1995). *Calculus, A Complete Course* (3rd ed.). Addison Wesley. p. 627. ISBN 0-201-82823-5.

8. Horn, Johnson (2013). *Matrix Analysis* (2nd ed.). Cambridge University Press. p. 6. ISBN 978-0-521-54823-6.

9. Peter Stingl (1996). *Mathematik für Fachhochschulen – Technik und Informatik* (in German) (5th ed.). Munich: Carl Hanser Verlag. ISBN 3-446-18668-9. Here: Exm.5.4.10, p.205-206

10. Weisstein, Eric W. "Matrix Multiplication" (https://mathworld.wolfram.com/MatrixMultiplicatio n.html). *mathworld.wolfram.com*. Retrieved 2020-09-06.

11. Lipcshutz, S.; Lipson, M. (2009). "2". *Linear Algebra*. Schaum's Outlines (4th ed.). McGraw Hill (USA). ISBN 978-0-07-154352-1.

12. Horn, Johnson (2013). "Chapter 0". *Matrix Analysis* (2nd ed.). Cambridge University Press. ISBN 978-0-521-54823-6.

13. Motwani, Rajeev; Raghavan, Prabhakar (1995). *Randomized Algorithms* (https://books.goog le.com/books?id=QKVY4mDivBEC&pg=PA280). Cambridge University Press. p. 280. ISBN 9780521474658.

14. Volker Strassen (Aug 1969). "Gaussian elimination is not optimal" (http://www.digizeitschrifte n.de/dms/img/?PID=GDZPPN001168215). *Numerische Mathematik*. **13** (4): 354–356. doi:10.1007/BF02165411 (https://doi.org/10.1007%2FBF02165411). S2CID 121656251 (htt ps://api.semanticscholar.org/CorpusID:121656251).

15. C.-C. Chou and Y.-F. Deng and G. Li and Y. Wang (1995). "Parallelizing Strassen's Method for Matrix Multiplication on Distributed-Memory MIMD Architectures" (https://core.ac.uk/dow nload/pdf/82778592.pdf) (PDF). *Computers Math. Applic*. **30** (2): 49–69. doi:10.1016/0898-1221(95)00077-C (https://doi.org/10.1016%2F0898-1221%2895%290007 7-C).

16. Vassilevska Williams, Virginia; Xu, Yinzhan; Xu, Zixuan; Zhou, Renfei. *New Bounds for Matrix Multiplication: from Alpha to Omega*. Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). pp. 3792–3835. arXiv:2307.07970 (https://arxi v.org/abs/2307.07970). doi:10.1137/1.9781611977912.134 (https://doi.org/10.1137%2F1.97 81611977912.134).

17. Nadis, Steve (March 7, 2024). "New Breakthrough Brings Matrix Multiplication Closer to Ideal" (https://www.quantamagazine.org/new-breakthrough-brings-matrix-multiplication-clos er-to-ideal-20240307). Retrieved 2024-03-09.

18. that is, in time $n^{2+\mathrm{f}(n)}$, for some function f with $f(n){\rightarrow}0$ as $n{\rightarrow}\infty$

References

- Henry Cohn, Robert Kleinberg, Balázs Szegedy, and Chris Umans. Group-theoretic Algorithms for Matrix Multiplication. arXiv:math.GR/0511460 (https://arxiv.org/abs/math.G R/0511460). *Proceedings of the 46th Annual Symposium on Foundations of Computer Science*, 23–25 October 2005, Pittsburgh, PA, IEEE Computer Society, pp. 379–388.

- Henry Cohn, Chris Umans. A Group-theoretic Approach to Fast Matrix Multiplication. arXiv:math.GR/0307321 (https://arxiv.org/abs/math.GR/0307321). *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, 11–14 October 2003, Cambridge, MA, IEEE Computer Society, pp. 438–449.

- Coppersmith, D.; Winograd, S. (1990). "Matrix multiplication via arithmetic progressions" (htt ps://doi.org/10.1016%2Fs0747-7171%2808%2980013-2). *J. Symbolic Comput*. **9** (3): 251– 280. doi:10.1016/s0747-7171(08)80013-2 (https://doi.org/10.1016%2Fs0747-7171%280 8%2980013-2).

- Horn, Roger A.; Johnson, Charles R. (1991), *Topics in Matrix Analysis*, Cambridge University Press, ISBN 978-0-521-46713-1

- Knuth, D.E., *The Art of Computer Programming* *Volume 2: Seminumerical Algorithms*. Addison-Wesley Professional; 3 edition (November 14, 1997). ISBN 978-0-201-89684-8. pp. 501.
- Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T. (2007), *Numerical Recipes: The Art of Scientific Computing* (3rd ed.), Cambridge University Press, ISBN 978-0-521-88068-8.
- Ran Raz. On the complexity of matrix product. In Proceedings of the thirty-fourth annual ACM symposium on Theory of computing. ACM Press, 2002. doi:10.1145/509907.509932 (https://doi.org/10.1145%2F509907.509932).
- Robinson, Sara, *Toward an Optimal Algorithm for Matrix Multiplication,* SIAM News 38(9), November 2005. PDF (https://web.archive.org/web/20100331095603/http://www.siam.org/pdf/news/174.pdf)
- Strassen, Volker, *Gaussian Elimination is not Optimal*, Numer. Math. 13, p. 354–356, 1969.
- Styan, George P. H. (1973), "Hadamard Products and Multivariate Statistical Analysis" (http://dml.cz/bitstream/handle/10338.dmlcz/102190/CzechMathJ_37-1987-4_14.pdf) (PDF), *Linear Algebra and Its Applications*, **6**: 217–240, doi:10.1016/0024-3795(73)90023-2 (https://doi.org/10.1016%2F0024-3795%2873%2990023-2)
- Williams, Virginia Vassilevska (2012-05-19). "Multiplying matrices faster than coppersmith-winograd" (http://dl.acm.org/citation.cfm?id=2213977.2214056). *Proceedings of the 44th symposium on Theory of Computing - STOC '12*. ACM. pp. 887–898. CiteSeerX 10.1.1.297.2680 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.297.2680). doi:10.1145/2213977.2214056 (https://doi.org/10.1145%2F2213977.2214056). ISBN 9781450312455. S2CID 14350287 (https://api.semanticscholar.org/CorpusID:14350287).