Matrix multiplication

In mathematics, **matrix multiplication** or **matrix product** is a binary operation that produces a matrix from two matrices with entries in a field, or, more generally, in a ring or even a semiring. The matrix product is designed for representing 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 applied mathematics, statistics, physics, economics, and engineering.[1][2] In more detail, if $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{B}$ is an $m \times p$ matrix, their matrix product $\mathbf{AB}$ is an $n \times p$ matrix, in which the m entries across a row of $\mathbf{A}$ are multiplied with the m entries down a column of $\mathbf{B}$ and summed to produce an entry of $\mathbf{AB}$. When two linear maps are represented by matrices, then the matrix product represents the composition of the two maps.

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 commutative. In many applications, the matrix elements belong to a field, although the tropical semiring is also a common choice for graph shortest path problems.[3] Even in the case of matrices over fields, the product is not commutative in general, although it is associative and is distributive over matrix addition. The identity matrices (which are the square matrices whose entries are zero outside of the main diagonal and 1 on the main diagonal) are identity elements of the matrix product. It follows that the $n \times n$ matrices over a ring form a ring, which is noncommutative except if $n = 1$ and the ground ring is commutative.

A square matrix may have a multiplicative inverse, called an inverse matrix. In the common case where the entries belong to a commutative ring r, a matrix has an inverse if and only if its determinant 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 group under matrix multiplication, the subgroups of which are called matrix groups. Many classical groups (including all finite groups) are isomorphic to matrix groups; this is the starting point of the theory of group representations.

Computing matrix products is a central operation in all computational applications of linear algebra. Its computational complexity is $O(n^3)$ (for $n \times n$ matrices) for the basic algorithm (this complexity is $O(n^{2.373})$ for the asymptotically fastest known algorithm[4]). This nonlinear complexity means that matrix product is often the critical part of many algorithms. This is enforced by the fact that many operations on matrices, such as matrix inversion, determinant, solving systems of linear equations, have the same complexity. Therefore various algorithms have been devised for computing products of large matrices, taking into account the architecture of computers (see BLAS, for example).

Contents

Notation

This article will use the following notational conventions: matrices are represented by capital letters in bold, e.g. $\mathbf{A}$, vectors in lowercase bold, e.g. $\mathbf{a}$, and entries of vectors and matrices are italic (since 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 i, j entry of matrix $\mathbf{A}$ is indicated by $(\mathbf{A})_{ij}$, A_{ij} or a_{ij}, whereas a numerical label (not matrix entries) on a collection of matrices is subscripted only, e.g. $\mathbf{A}_1$, $\mathbf{A}_2$, etc.

Definition

If $\mathbf{A}$ is an $n \times m$ matrix and $\mathbf{B}$ is an $m \times p$ matrix,

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

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

$$\mathbf{C} = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{np} \end{pmatrix}$$

such that

$$c_{ij} = a_{i1}b_{1j} + \cdots + a_{im}b_{mj} = \sum_{k=1}^{m} a_{ik}b_{kj},$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, p$.

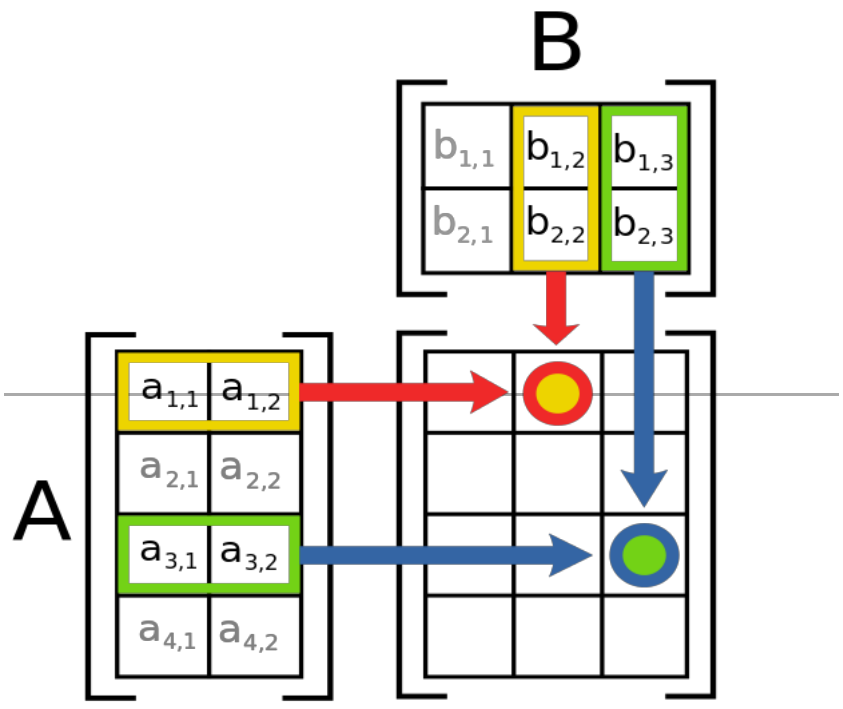
That is, the entry c_{ij} of the product is obtained by multiplying term-by-term the entries of the ith row of **A** and the jth column of **B**, and summing these m products. In other words, c_{ij} is the dot product of the ith row of **A** and the jth column of **B**.

Thus the product **AB** is defined if and only if the number of columns in **A** equals the number of rows in **B**, in this case m.

Usually the entries are numbers, but they may be any kind 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).

Illustration

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



$$\underset{4\times 2 \text{ matrix}}{\begin{bmatrix} a_{11} & a_{12} \\ \cdot & \cdot \\ a_{31} & a_{32} \\ \cdot & \cdot \end{bmatrix}} \underset{2\times 3 \text{ matrix}}{\begin{bmatrix} \cdot & b_{12} & b_{13} \\ \cdot & b_{22} & b_{23} \end{bmatrix}} = \underset{4\times 3 \text{ matrix}}{\begin{bmatrix} \cdot & x_{12} & x_{13} \\ \cdot & \cdot & \cdot \\ \cdot & x_{32} & x_{33} \\ \cdot & \cdot & \cdot \end{bmatrix}}$$

The values at the intersections marked with circles are:

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

Fundamental applications

Historically, matrix multiplication has been introduced for making easier 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, engineering and computer science.

Linear maps

If a vector space has a finite basis, its elements (vectors) are uniquely represented by a finite sequence, called coordinate vector, or scalars, which are the coordinates of the vector on the basis. These coordinates are commonly organized as a column matrix (also called *column vector*), that is a matrix with only one column.

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 of the function composition (that is, $(B \circ A)(\mathbf{x}) = B(A(\mathbf{x}))$) is instanced here as a specific case of associativity of matrix product (see below):

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

System of linear equations

The general form of a system of linear equations is

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

Using same notation as above, such a system is equivalent with the single matrix equation

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

Dot product, bilinear form and inner product

The dot product of two column vectors is the matrix product

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

where $\mathbf{x}^{\mathsf{T}}$ is the row vector obtained by transposing $\mathbf{x}$ and the resulting 1×1 matrix is identified with its unique entry.

More generally, any bilinear form over a vector space of finite dimension may be expressed as a matrix product

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

and any inner product may be expressed as

$$\mathbf{x}^{\dagger}\mathbf{A}\mathbf{y},$$

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

General properties

Matrix multiplication shares some properties with usual multiplication. 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, even when the product remains definite after changing the order of the factors.[9][10]

Non-commutativity

An operation is commutative 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 is not defined in general. If $m = q \neq n = p$, the two products are defined, but have different sizes; thus they cannot be equal.

It follows that the equality of the two products makes sense only if $m = q = n = p$, that is if $\mathbf{A}$ and $\mathbf{B}$ are square matrices of the same size. Even in this case, one has in general

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

For example

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

and

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

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.

Distributivity

The matrix product is distributive with respect of 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)

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

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 of 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, 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 $i = 1, \ldots, n-1$, 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.

Complexity is not associative

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

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

Square matrices

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

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

If $n > 1$, many matrices do not have a multiplicative inverse. 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

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

A matrix that has an inverse is an invertible matrix. Otherwise, it is a singular matrix.

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 commutative, and, in particular, when it is a field, the determinant 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 invariants do not behave as well with products. Nevertheless, if R is commutative, $\mathbf{AB}$ and $\mathbf{BA}$ have the same trace, the same characteristic polynomial, and the same eigenvalues with the same multiplicities. However, the eigenvectors are generally different if $\mathbf{AB} \neq \mathbf{BA}$.

Powers of a matrix

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

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

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 exponentiation by squaring, 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 diagonal matrix. 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:

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

Complexity

The matrix multiplication algorithm that results of the definition requires, in the worst case, n^3 multiplications of scalars and $(n-1)n^2$ additions for computing the product of two square $n \times n$ matrices. Its computational complexity is therefore $O(n^3)$, in a model of computation for which the scalar operations require a constant time (in practice, this is the case for floating point numbers, but not for integers).

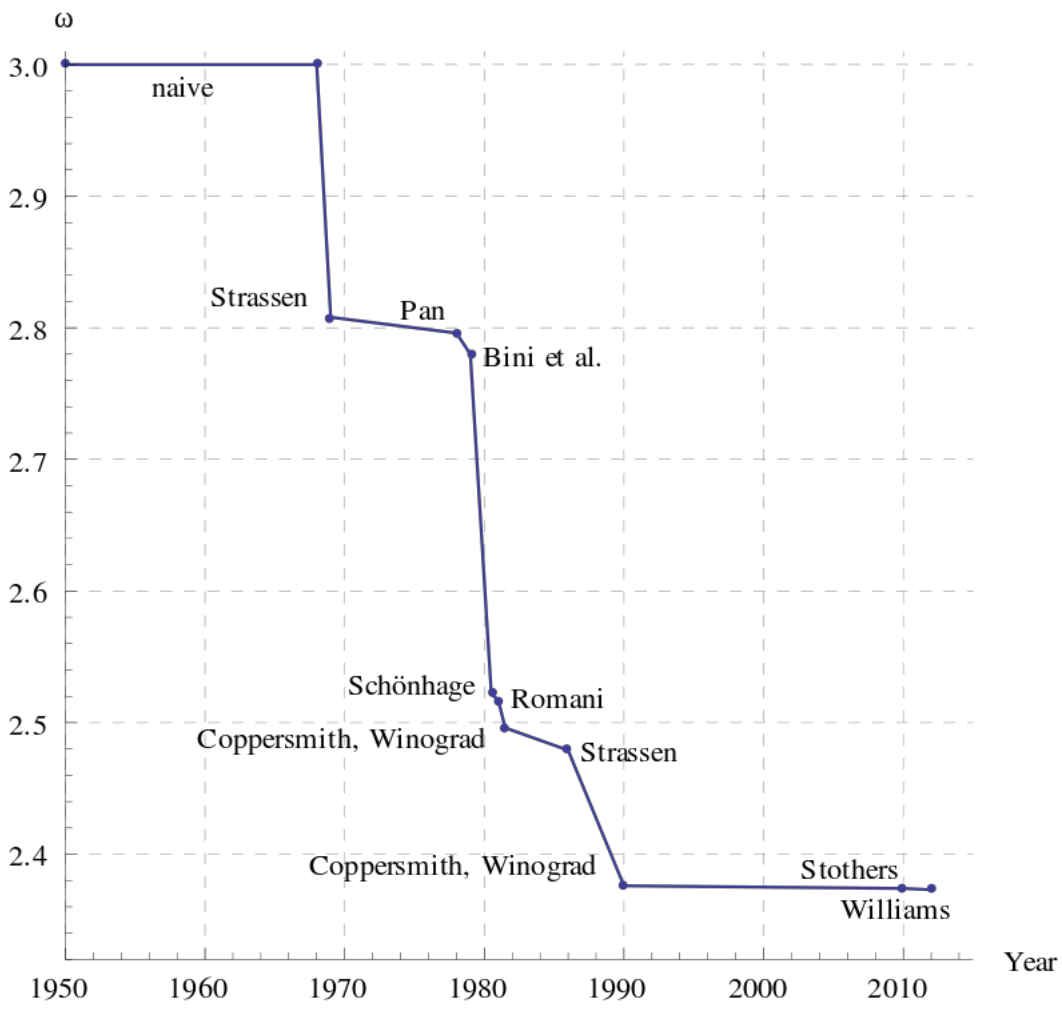
Rather surprisingly, this complexity is not optimal, as shown in 1969 by Volker Strassen, who provided an algorithm, now called Strassen's algorithm, with a complexity of $O(n^{\log_2 7}) \approx O(n^{2.807})$. The exponent appearing in the complexity of matrix multiplication has been improved several times, leading to Coppersmith–Winograd algorithm

with a complexity of $O(n^{2.376})$ (1990).[11] This algorithm has been slightly improved in 2013 by Virginia Vassilevska Williams (exponent 2.3729) and in 2014 by François Le Gall, for a final (up to date) complexity of $O(n^{2.3728639})$.[4]

The greatest lower bound for the exponent of matrix multiplication algorithm is generally called ω. One has $2 \leq \omega$, because one has to read the n^2 elements of a matrix for multiplying it by another matrix. Thus $2 \leq \omega < 2.373$. It is unknown whether $2 < \omega$. The largest known lower bound for matrix-multiplication complexity is $\Omega(n^2 \log(n))$, for a restricted kind of arithmetic circuits, and is due Ran Raz.[12]

Related complexities

The importance of the computational complexity of matrix multiplication relies on the facts that many algorithmic problems may be solved by means of matrix computation, and most problems on matrices have a complexity which is either the same as that of matrix multiplication (up to a multiplicative constant), or may be expressed in term of the complexity of matrix multiplication or its exponent ω.



The bound on ω over time.

There are several advantages of expressing complexities in terms of the exponent ω of matrix multiplication. Firstly, if ω is improved, this will automatically improve the known upper bound of complexity of many algorithms. Secondly, in practical implementations, one never uses the matrix multiplication algorithm that has the best asymptotical complexity, because the constant hidden behind the big O notation is too large for making the algorithm competitive for sizes of matrices that can be manipulated in a computer. Thus expressing complexities in terms of ω provide a more realistic complexity, since it remains valid whichever algorithm is chosen for matrix computation.

Problems that have the same asymptotic complexity as matrix multiplication include determinant, matrix inversion, Gaussian elimination (see next section). Problems with complexity that is expressible in terms of ω include characteristic polynomial, eigenvalues (but not eigenvectors), Hermite normal form, and Smith normal form.

Matrix inversion, determinant and Gaussian elimination

In his 1969 paper, where he proved the complexity $O(n^{2.807})$ for matrix computation, Strassen proved also the Matrix inversion, determinant and Gaussian elimination have, up to a multiplicative constant, the same computational complexity as matrix multiplication. The proof does not make any assumptions on matrix multiplication that is used, except that its complexity is $O(n^{\omega})$ for some $\omega \geq 2$

The starting point of Strassen's proof is using block matrix multiplication. Specifically, a matrix of even dimension $2n \times 2n$ may be partitioned in four $n \times n$ blocks

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

Under this form, its inverse is

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix},$$

provided that A and $D - CA^{-1}B$ are invertible.

Thus, the inverse of a $2n \times 2n$ matrix may be computed with two inversions, six multiplications and four additions or additive inverses of $n \times n$ matrices. It follows that, denoting respectively by $I(n)$, $M(n)$ and $A(n) = n^2$ the number of operations needed for multiplying, inverting and adding $n \times n$ matrices, one has

$$I(2n) \leq 2I(n) + 6M(n) + 4A(n).$$

If $n = 2^k$, one may apply this formula recursively:

$$\begin{aligned} I(2^k) &\leq 2I(2^{k-1}) + 6M(2^{k-1}) + 4A(2^{k-1}) \\ &\leq 2^2 I(2^{k-2}) + 6(M(2^{k-1}) + 2M(2^{k-2})) + 4(A(2^{k-1}) + 2A(2^{k-2})) \\ &\cdots \end{aligned}$$

If $M(n) \leq cn^\omega$, and $\alpha = 2^\omega \geq 4$, one gets eventually

$$\begin{aligned} I(2^k) &\leq 2^k I(1) + 6c(\alpha^{k-1} + 2\alpha^{k-2} + \cdots + 2^{k-1}\alpha^0) + k2^{k+1} \\ &\leq 2^k + 6c\frac{\alpha^k - 2^k}{\alpha - 2} + k2^{k+1} \\ &\leq d(2^k)^\omega. \end{aligned}$$

for some constant d.

For matrices whose dimension is not a power of two, the same complexity is reached by increasing the dimension of the matrix to a power of two, by padding the matrix with rows and columns whose entries are 1 on the diagonal and 0 elsewhere.

This proves the asserted complexity for matrices such that all submatrices that have to be inverted are indeed invertible. This complexity is thus proved for almost all matrices, as a matrix with randomly chosen entries is invertible with probability one.

The same argument applies to LU decomposition, as, if the matrix A is invertible, the equality

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ 0 & D - CA^{-1}B \end{bmatrix}$$

defines a block LU decomposition that may be applied recursively to A and $D - CA^{-1}B$, for getting eventually a true LU decomposition of the original matrix.

The argument applies also for the determinant, since it results from the block LU decomposition that

$$\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(A) \det(D - CA^{-1}B).$$

Other matrix multiplications

The term "matrix multiplication" is most commonly reserved for the definition given in this article. It could be more loosely applied to other operations on matrices.

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

Notes

1. Lerner, R. G.; Trigg, G. L. (1991). *Encyclopaedia of Physics* (2nd ed.). VHC publishers. ISBN 3-527-26954-1.
2. Parker, C. B. (1994). *McGraw Hill Encyclopaedia of Physics* (2nd ed.). ISBN 0-07-051400-3.
3. Motwani, Rajeev; Raghavan, Prabhakar (1995). *Randomized Algorithms* (https://books.google.com/books?id=QKVY4mDivBEC&pg=PA280). Cambridge University Press. p. 280. ISBN 9780521474658.
4. Le Gall, François (2014), "Powers of tensors and fast matrix multiplication", *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC 2014)*, arXiv:1401.7714 (https://arxiv.org/abs/1401.7714), Bibcode:2014arXiv1401.7714L (http://adsabs.harvard.edu/abs/2014arXiv1401.7714L)
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*. 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. Lipcshutz, S.; Lipson, M. (2009). "2". *Linear Algebra*. Schaum's Outlines (4th ed.). McGraw Hill (USA). ISBN 978-0-07-154352-1.
10. Horn, Johnson (2013). "0". *Matrix Analysis* (2nd ed.). Cambridge University Press. ISBN 978 0 521 54823 6.
11. Williams, Virginia Vassilevska. "Multiplying matrices faster than Coppersmith-Winograd" (http://www.cs.stanford.edu/~virgi/matrixmult-f.pdf) (PDF).
12. Raz, Ran (January 2003). "On the Complexity of Matrix Product" (http://epubs.siam.org/doi/10.1137/S0097539702402147). *SIAM Journal on Computing*. **32** (5): 1356–1369. doi:10.1137/s0097539702402147 (https://doi.org/10.1137%2Fs0097539702402147). ISSN 0097-5397 (https://www.worldcat.org/issn/0097-5397).

References

- Henry Cohn, Robert Kleinberg, Balázs Szegedy, and Chris Umans. Group-theoretic Algorithms for Matrix Multiplication. arXiv:math.GR/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. *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". *J. Symbolic Comput.* **9**: 251–280. doi:10.1016/s0747-7171(08)80013-2 (https://doi.org/10.1016%2Fs0747-7171%2808%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 (

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", *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). ACM: 887–898. doi:10.1145/2213977.2214056 (https://doi.org/10.1145%2F2213977.2214056). ISBN 9781450312455.
