Learning Matrix Multiplication: A Step-by-Step Tutorial Using Google Sheets

Authored by

Mohammed looti

November 16, 2025

Mastering Matrix Multiplication within Google Sheets

Welcome to this comprehensive guide dedicated to executing complex **matrix multiplication** operations with precision and speed using **Google Sheets**. Google Sheets is far more than a basic spreadsheet tool; it functions as a robust platform capable of handling sophisticated data analysis and numerical computations. Matrix multiplication is a fundamental operation central to many technical fields, including **linear algebra**, **computer graphics**, **physics**, and **statistics**, enabling essential functions like data transformations and the solution of complex systems of equations.

Attempting to calculate matrix products manually, especially when dealing with high-dimensional or large datasets, is notoriously time-consuming and highly prone to human error. Recognizing this challenge, Google Sheets provides a powerful, specialized function designed specifically to automate this critical process. This article serves as your definitive roadmap to mastering the **MMULT** function, ensuring you achieve maximum accuracy and operational efficiency in all your matrix calculations. Proficiency in utilizing the **MMULT function** is indispensable for any user looking to utilize Google Sheets for advanced analytical and mathematical tasks that extend beyond standard spreadsheet capabilities.

Regardless of your field--be it academic research, engineering, or business intelligence--developing expertise in performing matrix multiplication using Google Sheets will dramatically enhance your analytical toolkit. We will meticulously break down the function's essential syntax, detail the strict mathematical prerequisites necessary for successful multiplication, and provide clear, hands-on examples illustrating its application across various **matrix dimensions**. By the conclusion of this tutorial, you will possess the confidence and skills required to tackle complex matrix products effortlessly.

Deconstructing the MMULT Function: Syntax and Compatibility Requirements

The entire framework for conducting matrix multiplication in **Google Sheets** relies entirely upon the **MMULT** function. This function is expertly engineered to calculate the true matrix product of two specified matrices while strictly adhering to the fundamental mathematical laws governing these operations. Before we move on to practical demonstrations, it is absolutely essential to establish a firm grasp of the function's core structure, its required arguments, and the non-negotiable compatibility rules that must be met.

The **MMULT** function utilizes a highly intuitive yet powerful syntax that clearly defines the two matrix factors involved in the calculation:

MMULT(matrix1, matrix2)

The following list provides a detailed explanation of the arguments necessary for the **MMULT**

function to execute successfully:

matrix1: This represents the first matrix in the multiplication sequence, often referred to as the pre-multiplier. It must be accurately input as either a defined **cell range** within your Google Sheet (e.g., A1:B5) or explicitly defined using an **array literal** (e.g., {1, 2; 3, 4}).
matrix2: This is the second matrix, or the post-multiplier. Similar to **matrix1**, it must be precisely specified as a numerical **cell range** or an array literal containing valid numerical data points.

The single most crucial mathematical prerequisite for successful matrix multiplication is the rule of "inner dimension compatibility": the number of columns in **matrix1** must exactly equal the number of rows in **matrix2**. Failure to satisfy this essential condition will immediately cause the **MMULT** function to return a **#VALUE! error**, signaling an incompatible mathematical setup. Furthermore, the resulting product matrix will always inherit the dimensions of the outer factors: the rows of **matrix1** and the columns of **matrix2**. For instance, multiplying a (2×3) matrix by a (3×4) matrix will correctly generate a resulting (2×4) matrix.

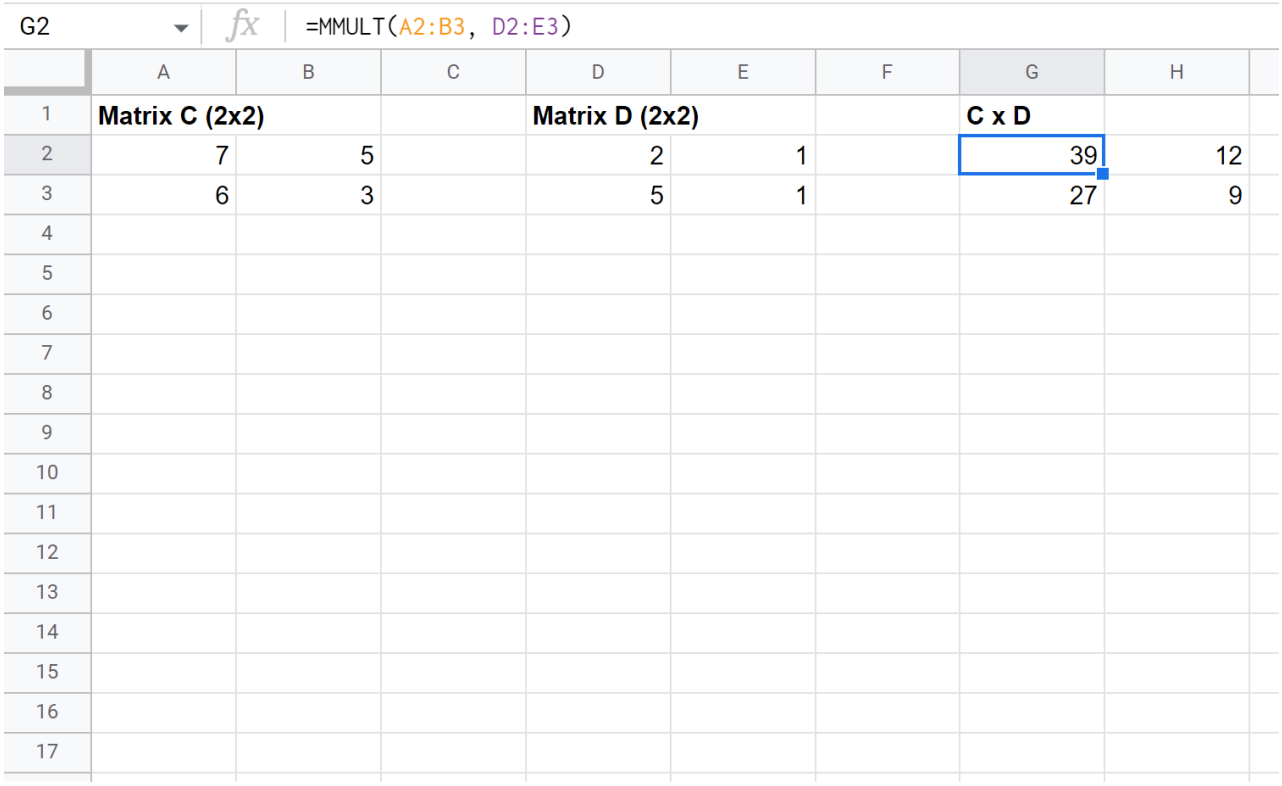
The subsequent examples are designed to thoroughly demonstrate the correct application of the **MMULT function** across various standard matrix configurations in Google Sheets, providing invaluable practical insights into performing different types of matrix products:

Multiplication of two square (2×2) matrices.
Multiplication of a (2×2) matrix by a non-square (2×3) matrix.
Multiplication of a (3×3) matrix by a rectangular (3×2) matrix.

Example 1: Multiplying Two Square (2×2) Matrices

Our first practical demonstration addresses the multiplication of two square matrices of equal size: a (2×2) matrix multiplied by another (2×2) matrix. This scenario serves as the clearest foundational illustration of the **MMULT** function's core mechanics and the structure of its output. The procedure is straightforward: first, define your two source matrices within distinct cell ranges, and then invoke the **MMULT** formula in the target output area to automatically calculate the product.

The accompanying screenshot visually depicts the two (2×2) matrices structured within our spreadsheet. The first matrix occupies the cell range **A2:B3**, while the second matrix is positioned in **D2:E3**. Since the number of columns in the first matrix (2) perfectly matches the number of rows in the second matrix (2), the inner dimension compatibility rule is satisfied, confirming the validity of the operation. Consequently, the result must also be a (2×2) matrix.



To execute this specific **matrix multiplication**, we simply enter the following formula into cell **G2**. A key characteristic of **MMULT** is its inherent behavior as an **array formula**: the output automatically "spills" or expands across the necessary adjacent cells to construct the complete resulting matrix. Thus, you only ever need to input the formula once, placing it in the top-left cell of your desired output location.

=MMULT(A2:B3, D2:E3)

Each individual element within the product matrix is derived by calculating the **dot product** between the rows of the first matrix and the corresponding columns of the second matrix. For a (2×2) multiplied by a (2×2) matrix, this precise process involves computing four distinct individual dot products. The image provided below offers a visual clarification of the underlying calculations that Google Sheets performs behind the scenes to derive each component of the resulting product matrix, giving invaluable insight into the mathematical computation process.

$$\mathbf{C \times D} = \begin{bmatrix} 7*2+5*5 & 7*1+5*1 \\ 6*2+3*5 & 6*1+3*1 \end{bmatrix}$$

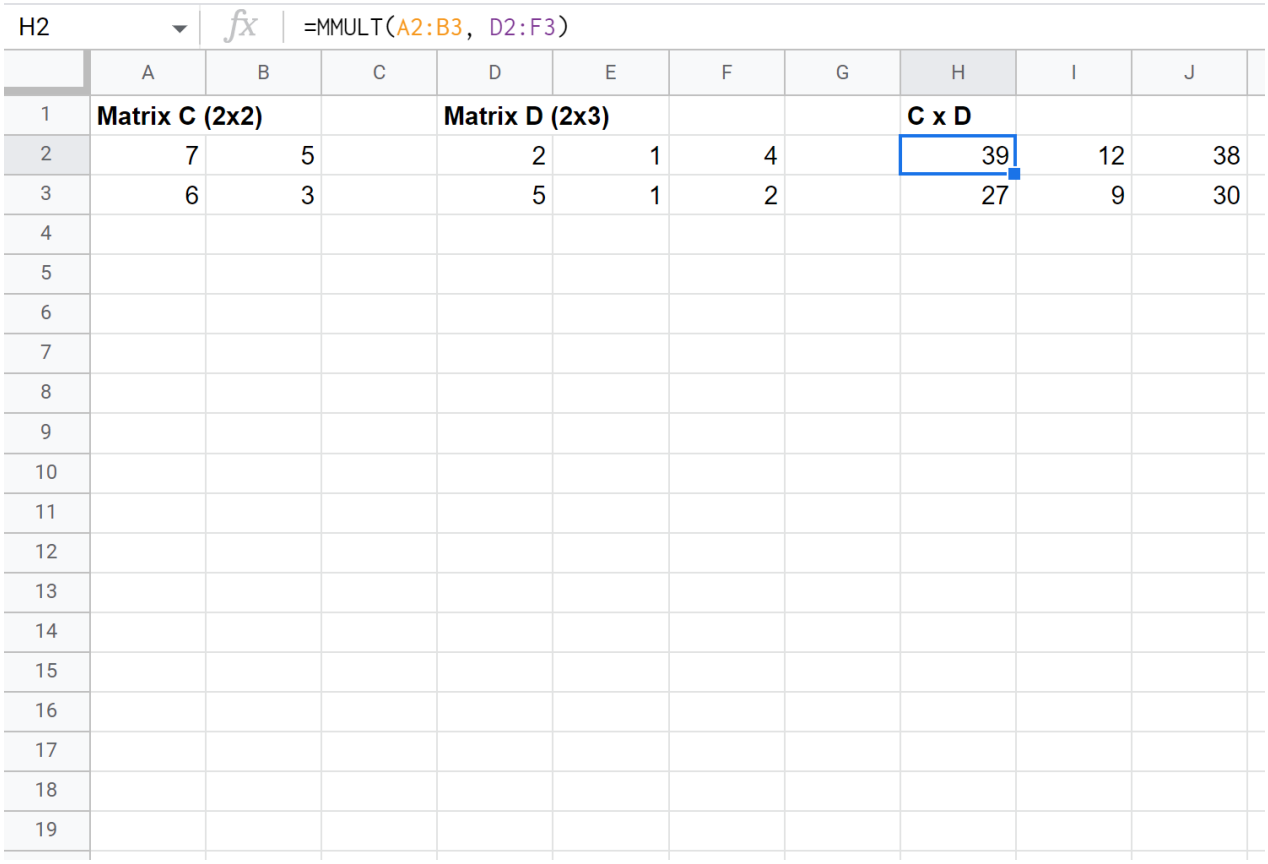
Upon successful execution of the **MMULT** function, Google Sheets populates the designated output range with the correct elements of the product matrix. As mathematically predicted, the multiplication of two (2×2) matrices yields another (2×2) matrix, which is clearly illustrated in the visual representation below. This demonstrates the seamless and reliable integration of complex matrix operations directly into your spreadsheet environment.

$$C \times D = \begin{bmatrix} 39 & 12 \\ 27 & 9 \end{bmatrix}$$

Example 2: Calculating the Product of a (2×2) by a (2×3) Matrix

Our second example focuses on a scenario involving matrices with non-identical **dimensions**. In this case, we will multiply a (2×2) matrix by a (2×3) matrix. This particular setup remains mathematically valid because the number of columns in the first matrix (2) perfectly aligns with the number of rows in the second matrix (2). As a direct consequence, the resulting product matrix will adopt the number of rows from the first matrix (2) and the number of columns from the second matrix (3), culminating in a final (2×3) matrix result.

The following screenshot illustrates the exact arrangement of the input data: a (2×2) matrix defined across the range **A2:B3** and a (2×3) matrix occupying the range **D2:F3** within Google Sheets. This visual configuration is crucial for correctly structuring the input data prior to utilizing the **MMULT** function. The expected output, the (2×3) product matrix, will automatically populate the cells starting from where the formula is entered, expanding precisely to fill the required space.



To execute this rectangular matrix multiplication, we input the following **MMULT** formula into cell **H2**. We emphasize the vital detail that only the top-left cell of the intended output range requires the formula. Google Sheets efficiently manages the automatic expansion of the entire result across the necessary adjacent cells, significantly streamlining the overall calculation process.

=MMULT(A2:B3, D2:F3)

For those interested in the underlying computational mechanics, the diagram below clarifies the six individual calculations Google Sheets performs to compute each unique element of the (2×3) product matrix. Every cell's value is precisely computed as the **dot product** of a row vector from the first matrix and a corresponding column vector from the second matrix--a complex set of operations managed flawlessly by the **MMULT** function.

$$\mathbf{C \times D} = \begin{bmatrix} 7*2+5*5 & 7*1+5*1 & 7*4+5*2 \\ 6*2+3*5 & 6*1+3*1 & 6*4+3*2 \end{bmatrix}$$

The final outcome of this operation is, accurately, a (2×3) matrix, which perfectly aligns with the established rules of **matrix multiplication** concerning varying input dimensions. This result, visually confirmed in the image provided below, validates the successful and robust application of the **MMULT** function and confirms its capability to handle matrices of different sizes, provided the fundamental dimension compatibility rule is strictly upheld.

$$C \times D = \begin{bmatrix} 39 & 12 & 38 \\ 27 & 9 & 30 \end{bmatrix}$$

Example 3: Multiplying a (3×3) by a (3×2) Matrix

In this third and final practical example, we increase the complexity by demonstrating the multiplication of a (3×3) matrix by a (3×2) matrix. This scenario is crucial as it further emphasizes the versatility and scalability of the **MMULT** function within the Google Sheets environment, proving its effectiveness for larger datasets. As established in previous sections, the core prerequisite for success remains the matching of the inner **dimensions**: the number of columns in the first matrix (3) must precisely equal the number of rows in the second matrix (3). Consequently, the resultant matrix will correctly assume the shape of the outer dimensions, specifically yielding a (3×2) matrix.

The screenshot below offers a clear visual representation of the input matrices for this advanced calculation. The first matrix, a (3×3) square matrix, is defined across the **cell range** **A2:C4**, while the second matrix, a (3×2) rectangular matrix, is situated in **E2:F4**. This precise setup is fundamental for correctly applying the **MMULT** function and ensuring the accurate calculation of the matrix product.

H2 | fx | =MMULT(A2:C4, E2:F4)

	A	B	C	D	E	F	G	H	I
1	**Matrix C (3x3)**				**Matrix D (3x2)**			**C x D**	
2	-3	5	4		2	1		19	-2
3	1	2	3		5	1		12	0
4	-1	0	2		0	-1		-2	-3
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

To perform this specific **matrix multiplication**, we input the familiar **MMULT** formula into cell **H2**. It is essential to reiterate that Google Sheets handles the automatic expansion of the resulting **array formula** into the appropriate number of cells required to fully form the product matrix. This powerful array functionality removes the need for cumbersome manual formula dragging or pre-selecting the entire output range.

=MMULT(A2:C4, E2:F4)

For complete transparency regarding the calculation methodology, the image below visually illustrates the six distinct formulas that Google Sheets employs to compute each element of the resulting product matrix. Each cell's numerical value is derived from the **dot product** of a corresponding row from the first matrix and a column from the second matrix, comprehensively demonstrating the detailed computation executed efficiently by the **MMULT** function.

C x D =

-3*2 + 5*5 + 4*0	-3*1 + 5*1 + 4*-1
1*2 + 2*5 + 3*0	1*1 + 2*1 + 3*-1
-1*2 + 0*5 + 2*0	-1*1 + 0*1 + 2*-1

The outcome of this operation is, precisely and mathematically soundly, a (3×2) matrix, perfectly consistent with the laws of matrix algebra. The final product, visually confirmed in the image below, provides concrete evidence of the **MMULT** function's accurate and efficient capacity to handle complex matrix operations involving larger and non-square matrices within the spreadsheet environment.

$$C \times D = \begin{bmatrix} 19 & -2 \\ 12 & 0 \\ -2 & -3 \end{bmatrix}$$

Critical Considerations for Using MMULT Effectively

While the **MMULT** function is designed to simplify inherently complex numerical calculations, successful implementation in Google Sheets requires close attention to several critical considerations. Understanding these nuances is essential for preventing common errors and guaranteeing the accuracy and reliability of your results, whether your matrix operations are for academic modeling, professional reporting, or complex numerical projects.

Firstly, you must rigorously verify the **dimensions** of your input matrices before attempting execution. As consistently emphasized throughout these examples, the number of columns in the first matrix must strictly match the number of rows in the second matrix. Failure to comply with this fundamental requirement--the inner dimension compatibility rule--will invariably result in a **#VALUE! error**. This specific error message is Google Sheets' primary indicator of a mathematically impossible matrix setup, and it remains the most common issue encountered by users new to matrix multiplication.

Secondly, it is mandatory to ensure that the input ranges you define contain exclusively numerical values. The **MMULT** function is engineered strictly for numerical computation. If Google Sheets detects non-numerical data--such as text strings, logical values (TRUE/FALSE), blank cells, or any pre-existing error values--within the specified matrix ranges, the function will typically fail and return an error. Best practice dictates thoroughly cleaning and validating all source data before initiating any matrix multiplication operation.

Lastly, always remember that **MMULT** operates inherently as an **array formula**. This means the function returns an entire array of calculated values (the resulting product matrix) and automatically attempts to spill or expand into adjacent cells. You must ensure that the designated output range possesses a sufficient number of completely empty cells to fully accommodate the dimensions of

the resulting matrix. If existing data is present in the cells where the product matrix needs to expand, **MMULT** will produce a **#REF! error**, clearly signaling that the array result could not expand because it would overwrite non-empty cells.

Conclusion and Expanding Your Analytical Capabilities

Through this comprehensive guide, we have thoroughly explored the methodology for performing **matrix multiplication** using the powerful **MMULT function** in **Google Sheets**. From understanding the core mathematical requirements and syntax to successfully applying the function across matrices of various **dimensions**, we have covered all the fundamental knowledge necessary to execute these complex numerical calculations accurately and with remarkable efficiency. The ability to seamlessly integrate matrix multiplication into your workflow is a highly valuable skill that significantly elevates your data analysis potential within the spreadsheet environment.

The **MMULT** function efficiently streamlines what would otherwise be an intricate and extremely error-prone mathematical process, thereby making matrix operations accessible to a much broader user base. By diligently adhering to the guidelines regarding matrix compatibility, ensuring data integrity (using only numerical inputs), and correctly managing the array expansion, you can confidently incorporate highly sophisticated matrix operations into your daily tasks. Whether your requirements involve advanced statistical modeling, engineering calculations, data transformations, or complex financial analysis, Google Sheets consistently proves itself to be a versatile and robust platform for both basic and highly advanced computations.

For users seeking more in-depth technical specifications, troubleshooting advice, or official functional information, the complete documentation for the **MMULT function** is available directly on Google's official support pages. This resource offers essential comprehensive insights and tracks any relevant platform updates to ensure continuous operational success.

To further enhance your overall proficiency in Google Sheets, the following tutorials explain how to perform other essential and common tasks, allowing you to continually expand your analytical toolkit: