A tutorial on inverting 3 by 3 matrices with cross products

By Cedrick Collomb[1]

Abstract. This tutorial introduces the idea of inverting a 3 by 3 matrix and calculating its determinant with cross products hopefully in a simple manner intelligible to any reader with minimal mathematical and engineering skills. The formulas are derived twice with different approaches and every step is purposely made unnecessarily detailed for ease of understanding. The goal of this document is to help the reader implement and use the ideas behind this tutorial immediately after reading it.

1. Introduction

The inversion of matrices is a recurring step in several Mathematics and Computer Science algorithms. Although there already exist several generic algorithms for the inversion of matrices, there are several domains such as Embedded software, Computer Graphics and Video Games for which performance is critical and for which generic algorithms used with low dimension matrices are not good enough. A subset of the performance critical problems encountered in those domains can be reduced and solved by a linear system of size 3 by 3; therefore the need for an efficient way to inverse a 3 by 3 matrix. On top of that hardware SIMD architectures have been providing for several years the means to compute cross products very efficiently, therefore making the technique introduced in this tutorial both a simple and efficient method.

The rest of the tutorial is organized as follow: Section 2 introduces the theory behind general square matrix inversion. Section 3 introduces the difference between theory and in use algorithms. Section 4 simplifies the results of Section 2 for small 3 by 3 matrices. Section 5 shows how to achieve the same results as Section 4 using cross products. Section 6 concludes this tutorial.

2. Basic linear algebra refresher

Let M be a square matrix of dimension n. We note $M = (m_{ij})$ with $i \in [\![1, n]\!]$ representing the row indices and $j \in [\![1, n]\!]$ representing the column indices. M can also be visualized in the following form $M = \begin{pmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{nn} \end{pmatrix}$. Notice that m_{1n} is the last element of the first line and m_{n1} is the first element of the last line.

[1] ccollomb@yahoo.com / http://ccollomb.free.fr/

Although outside the scope of this tutorial[2] we can demonstrate that M is invertible if and only if its determinant that will be referenced as $det(M)$ is not zero. When M is invertible, its inverse is written as M^{-1} and can be determined by the following formula:

$$M^{-1} = \frac{1}{\det(M)} adj(M) \tag{1}$$

The notation $adj(M)$ represents the adjugate matrix of M. The formula means that the inverse of M is the adjugate scaled by the inverse of the determinant which is a scalar. The adjugate and determinant still need to be explained so that the previous formula makes sense and be at all useable.

The adjugate of M is also a square matrix of dimension n, and its elements a_{ij} are defined by the following formula: $a_{ij} = c_{ji}$ (notice that i and j are inverted on the right hand side of the equation so that to transpose the matrix). The coefficients c_{ij} are called cofactors and are defined with the determinant of the square matrix of dimension $n-1$ created by removing the line i and column j from M.

$$c_{ij} = (-1)^{i+j} \det \begin{pmatrix} m_{11} & \cdots & m_{1,j-1} & m_{1,j+1} & \cdots & m_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ m_{i-1,1} & \cdots & m_{i-1,j-1} & m_{i-1,j+1} & \cdots & m_{i-1,n} \\ m_{i+1,1} & \cdots & m_{i+1,j-1} & m_{i+1,j+1} & \cdots & m_{i+1,n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ m_{nn} & \cdots & m_{n,j-1} & m_{n,j+1} & \cdots & m_{nn} \end{pmatrix} \tag{2}$$

The determinant of M can be calculated with the following formula:

$$\det(M) = \sum_{i=1}^{n} m_{ij} c_{ij} \tag{3}$$

The choice of j in (3) does not matter and for a matrix of dimension one, $\det(M) = m_{11}$.

As you might have noticed this create a recursive formula since $\det(M)$ requires the value of the cofactors c_{ij}, and the cofactors are defined with determinants! However you can also notice that the size of the matrices involved in the cofactor is always one less than the size of the matrix of which we calculate the determinant. Therefore the recursion will reach matrices of size one and therefore will end.

[2] I recommend any good Linear Algebra course book, for French readers [1] is a good reference

3. Matrix inversion in practice

Although the inversion of matrices with adjugate has theoretical value, it is important to mention that this form is very rarely used in practice to inverse matrices.

For several classes of problem inversing the matrix just to find one x so that $Mx = y$ when you have M and y can be prohibitory expensive, and it makes more sense to create iterative algorithms that converge to x. The most common methods[3] to proceed this way are Jacobi, Gauss-Seidel and the Conjugate gradient.

Even for the cases where the inverse is really needed there are usually better algorithms. The most common methods[4] are Gauss-Jordan, Shipley-Coleman [4], LU decomposition, QR decomposition.

4. The 3 by 3 case

If we apply (2) and (3) to the square matrix $M = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$, we get the following:

$$\det(M) = m_{11}c_{11} + m_{21}c_{21} + m_{31}c_{31} \tag{4}$$

$$c_{11} = (-1)^{1+1}\det\begin{pmatrix} m_{22} & m_{23} \\ m_{32} & m_{33} \end{pmatrix} = m_{22}m_{33} - m_{32}m_{23} \tag{5}$$

$$c_{21} = (-1)^{2+1}\det\begin{pmatrix} m_{12} & m_{13} \\ m_{32} & m_{33} \end{pmatrix} = m_{32}m_{13} - m_{12}m_{33} \tag{6}$$

$$c_{31} = (-1)^{3+1}\det\begin{pmatrix} m_{12} & m_{13} \\ m_{22} & m_{23} \end{pmatrix} = m_{12}m_{23} - m_{22}m_{13} \tag{7}$$

$$c_{12} = (-1)^{1+2}\det\begin{pmatrix} m_{21} & m_{23} \\ m_{31} & m_{33} \end{pmatrix} = m_{31}m_{23} - m_{21}m_{33} \tag{8}$$

$$c_{22} = (-1)^{2+2}\det\begin{pmatrix} m_{11} & m_{13} \\ m_{31} & m_{33} \end{pmatrix} = m_{11}m_{33} - m_{31}m_{13} \tag{9}$$

$$c_{32} = (-1)^{3+2}\det\begin{pmatrix} m_{11} & m_{13} \\ m_{21} & m_{23} \end{pmatrix} = m_{13}m_{21} - m_{11}m_{23} \tag{10}$$

---

[3] See [**2**] for a good start with common methods for solving linear systems and their descriptions.
[4] See [**3**] for a good start with matrix inversion methods.

$$c_{13} = (-1)^{1+3} \det\begin{pmatrix} m_{21} & m_{22} \\ m_{31} & m_{32} \end{pmatrix} = m_{21}m_{32} - m_{31}m_{22} \tag{11}$$

$$c_{23} = (-1)^{2+3} \det\begin{pmatrix} m_{11} & m_{12} \\ m_{31} & m_{32} \end{pmatrix} = m_{31}m_{12} - m_{11}m_{32} \tag{12}$$

$$c_{33} = (-1)^{3+3} \det\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} = m_{11}m_{22} - m_{21}m_{12} \tag{13}$$

Using (5), (6) and (7) with (4) gives:

$$\det(M) = m_{11}\left(m_{22}m_{33} - m_{32}m_{23}\right) + m_{21}\left(m_{32}m_{13} - m_{12}m_{33}\right) + m_{31}\left(m_{12}m_{23} - m_{22}m_{13}\right) \tag{14}$$

Using (5), (6), (7), (8), (9), (10), (11), (12) and (13) with (1) gives:

$$M^{-1} = \frac{1}{\det(M)}\begin{pmatrix} m_{22}m_{33} - m_{32}m_{23} & m_{32}m_{13} - m_{12}m_{33} & m_{12}m_{23} - m_{22}m_{13} \\ m_{31}m_{23} - m_{21}m_{33} & m_{11}m_{33} - m_{31}m_{13} & m_{13}m_{21} - m_{11}m_{23} \\ m_{21}m_{32} - m_{31}m_{22} & m_{31}m_{12} - m_{11}m_{32} & m_{11}m_{22} - m_{21}m_{12} \end{pmatrix} \tag{15}$$

You can verify by hand that $MM^{-1} = I_3$, by multiplying M by M^{-1}.

5. Calculating the determinant with a dot product and a cross product

Although the formula (15) can be optimized, it is based on scalar operations and do not take advantage of SIMD features commonly available on most CPUs. On some processor architectures an alternative formulation using SIMD can provide more efficient results.

A common feature with SIMD is the availability of vectors of several dimensions that stand for the MD of Multiple Data in SIMD, in most common platforms available today those vectors are four dimensional vectors storing floating point values or 32 bits integers.

Stepping back from the final form reached in the last formula and looking at (4), we can see that $\det(M)$ looks like a dot product of two three-dimensional vectors. Defining $C_1 = \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \end{bmatrix}$ which is the first column of M, and $R_1 = \begin{bmatrix} c_{11} & c_{21} & c_{31} \end{bmatrix}$ which is the first row of $adj(M)$, we notice that $\det(M) = R_1 C_1$ or using a dot product form:

$$\det(M) = R_1^T . C_1 \tag{16}$$

Now if we look carefully at R_1^T we get:

$$R_1^T = \begin{bmatrix} c_{11} \\ c_{21} \\ c_{31} \end{bmatrix} = \begin{bmatrix} m_{22}m_{33} - m_{32}m_{23} \\ m_{32}m_{13} - m_{12}m_{33} \\ m_{12}m_{23} - m_{22}m_{13} \end{bmatrix} = \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix} \wedge \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \end{bmatrix} = C_2 \wedge C_3 \tag{17}$$

C_2 and C_3 the second and third rows of M, therefore $C_2 = \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix}$ and $C_3 = \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \end{bmatrix}$.

Therefore using (17) in (16) gives:

$$\det(M) = C_1.\left(C_2 \wedge C_3\right) \tag{18}$$

6. Inverting the matrix with cross products

The formula (18) is quite simple and nice. Moreover through the calculus we also realized that the first row of $adj(M)$ was the cross product of the second and third columns of M, which was interesting and encourage to look at the other rows of $adj(M)$.

If we call respectively R_2 and R_3 the second and third rows of $adj(M)$, and call respectively C_2 and C_3 the second and third rows of $adj(M)$

$$R_2^T = \begin{bmatrix} c_{12} \\ c_{22} \\ c_{32} \end{bmatrix} = \begin{bmatrix} m_{31}m_{23} - m_{21}m_{33} \\ m_{11}m_{33} - m_{31}m_{13} \\ m_{13}m_{21} - m_{11}m_{23} \end{bmatrix} = \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \end{bmatrix} \wedge \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \end{bmatrix} = C_3 \wedge C_1 \tag{19}$$

$$R_3^T = \begin{bmatrix} c_{12} \\ c_{22} \\ c_{32} \end{bmatrix} = \begin{bmatrix} m_{21}m_{32} - m_{31}m_{22} \\ m_{31}m_{12} - m_{11}m_{32} \\ m_{11}m_{22} - m_{21}m_{12} \end{bmatrix} = \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \end{bmatrix} \wedge \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix} = C_1 \wedge C_2 \tag{20}$$

Using (17), (18), (19) and (20) we get the elegant and SIMD friendly formula:

$$M^{-1} = \frac{1}{C_1.\left(C_2 \wedge C_3\right)}\left(C_2 \wedge C_3 \quad C_3 \wedge C_1 \quad C_1 \wedge C_2\right)^T \tag{21}$$

7. An alternative way to get the cross product inversion formula

The cross product of two vectors is by definition normal to both vectors used to calculate the cross product, if we choose C_i and C_j with $(i, j) \in \{1,2,3\} \times \{1,2,3\}$ then

$$C_i.\left(C_i \wedge C_j\right) = 0 \text{ and } C_j.\left(C_i \wedge C_j\right) = 0 \tag{22}$$

If we write two matrices A and B with the following rows and columns notations:

$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix}$, where A_1, A_2 and A_3 are three dimensional horizontal vectors.

$B = \begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix}$, where B_1, B_2 and B_3 are three dimensional vertical vectors.

Then
$$AB = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} \begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix} = \begin{bmatrix} A_1^T.B_1 & A_1^T.B_2 & A_1^T.B_3 \\ A_2^T.B_1 & A_2^T.B_2 & A_2^T.B_3 \\ A_3^T.B_1 & A_3^T.B_2 & A_3^T.B_3 \end{bmatrix} \tag{23}$$

Imagine that B is M and we want A to be its inverse M^{-1}.

So we want $AB = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, therefore we need to make sure that $A_1^T.B_2 = 0$ and $A_1^T.B_3 = 0$. We know B_2 and B_3 since they are columns of M, and we want to find a vector that is normal to both.

Using (22) it is very easy with $A_1^T = k_1\left(B_2 \wedge B_3\right)$. Following the same method we can identify $A_2^T = k_2\left(B_3 \wedge B_1\right)$ and $A_3^T = k_3\left(B_1 \wedge B_2\right)$. Therefore

$$A = \left[k_1\left(B_2 \wedge B_3\right) \quad k_2\left(B_3 \wedge B_1\right) \quad k_3\left(B_1 \wedge B_2\right)\right]^T \tag{24}$$

We now need to identify k_1, k_2 and k_3 by using (24) in (23) which gives

$$AB = \begin{bmatrix} k_1 B_1.\left(B_2 \wedge B_3\right) & 0 & 0 \\ 0 & k_2 B_2.\left(B_3 \wedge B_1\right) & 0 \\ 0 & 0 & k_3 B_3.\left(B_1 \wedge B_2\right) \end{bmatrix}$$

Therefore

$$k_1 = \frac{1}{B_1.(B_2 \wedge B_3)} \tag{25}$$

$$k_2 = \frac{1}{B_2.(B_3 \wedge B_1)} \tag{26}$$

$$k_3 = \frac{1}{B_3.(B_1 \wedge B_2)} \tag{27}$$

Another interesting property if E, F, G are arbitrary vectors is that we can cycle vectors to the left or to the right in the equation below.

$$E.(F \wedge G) = F.(G \wedge E) = G.(E \wedge F) \tag{28}$$

Note that we only need to demonstrate one equality, since changing variable names leads to the third equality.

$E = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$ and $F \wedge G = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \wedge \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} f_2 g_3 - f_3 g_2 \\ f_3 g_1 - f_1 g_3 \\ f_1 g_2 - f_2 g_1 \end{bmatrix}$, therefore

$$E.(F \wedge G) = e_1 (f_2 g_3 - f_3 g_2) + e_2 (f_3 g_1 - f_1 g_3) + e_3 (f_1 g_2 - f_2 g_1) \tag{29}$$

Expanding (29) gives

$$E.(F \wedge G) = e_1 f_2 g_3 - e_1 f_3 g_2 + e_2 f_3 g_1 - e_2 f_1 g_3 + e_3 f_1 g_2 - e_3 f_2 g_1 \tag{30}$$

Factoring (30) by f_1, f_2 and f_3 gives

$$E.(F \wedge G) = f_1 (e_3 g_2 - e_2 g_3) + f_2 (e_1 g_3 - e_3 g_1) + f_3 (e_2 g_1 - e_1 g_2) \tag{31}$$

$G \wedge E = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \wedge \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} g_2 e_3 - g_3 e_2 \\ g_3 e_1 - g_1 e_3 \\ g_1 e_2 - g_2 e_1 \end{bmatrix}$ therefore see from (31) that $E.(F \wedge G) = F.(G \wedge E)$.

Using (28) with (25), (26) and (27) we get

$$k_1 = k_2 = k_3 = \frac{1}{B_1.(B_2 \wedge B_3)} \tag{32}$$

And finally using (32) and (24) we reach the same result as in the previous section

$$A = \frac{1}{B_1.\left(B_2 \wedge B_3\right)}\left[B_2 \wedge B_3 \quad B_3 \wedge B_1 \quad B_1 \wedge B_2\right]^T$$

8. Conclusion

This document has shown with two distinct approaches that it is possible to invert a three by three matrix with three cross products, one dot product and a matrix transpose. It has also shown that the determinant of the same matrix could be calculated with one cross product and one dot product.

Although inverting a matrix is a solved problem, finding optimal algorithms is still an unsolved problem. Of related importance is the fact that the computing complexity of matrix inversion is equivalent to matrix multiplication[5], and also equivalent to the complexity of solving a collection of linear equations and computing the determinant of a matrix[6]. Therefore any progress made by any of those area of research immediately will benefit the other areas.

A lot of progress has been made since the seminal discovery [**7**], made by Strassen in 1969. However there is still major interest to investigate and discover new and improved algorithms to address those areas, and the reader of this tutorial can take part in this research effort.

9. References

1. J.M Arnaudies, Cours de Mathematiques T.1 Algebre, Bordas (1993)
2. R. Barrett et al., Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition, SIAM (1994)
3. W.H. Press et al., Numerical Recipes in C++: The Art of Scientific Computing, Cambridge University Press (2002)
4. J.P. Pante, Computer Algorithms in Power System Analysis, World Wide Web site at the address http://www.geocities.com/SiliconValley/Lab/4223/fault/ach03.html, (1999)
5. T.H. Cormen, et al., *Introduction to Algorithms*, Second Edition, MIT Press and McGraw-Hill (2001)
6. V. Pan, How can we speed up matrix multiplication?, *SIAM Rev.* 26 (1984)
7. V. Strassen, Gaussian Elimination is not Optimal, Numer. Math. (1969)

---

[5] See [**5**] for more details.
[6] See [**6**] for more details.