

What Is $A \setminus A$?

Nicholas J. Higham*

March 8, 2022

In a recent blog post [What is \$A \setminus A\$?](#), Cleve Moler asked what the MATLAB operation $A \setminus A$ returns. I will summarize what backslash does in general, for $A \setminus B$ and then consider the case $B = A$.

$A \setminus B$ is a solution, in some appropriate sense, of the equation

$$AX = B, \quad A \in \mathbb{C}^{m \times n} \quad X \in \mathbb{C}^{n \times p} \quad B \in \mathbb{C}^{m \times p}. \quad (1)$$

It suffices to consider the case $p = 1$, because backslash treats the columns independently, and we write this as

$$Ax = b, \quad A \in \mathbb{C}^{m \times n} \quad x \in \mathbb{C}^n \quad b \in \mathbb{C}^m.$$

The MATLAB backslash operator handles several cases depending on the relative sizes of the row and column dimensions of A and whether it is rank deficient.

Square Matrix: $m = n$

When A is square, backslash returns $x = A^{-1}b$, computed by LU factorization with partial pivoting (and of course without forming A^{-1}). There is no special treatment for singular matrices, so for them division by zero may occur and the output may contain NaNs (in practice, what happens will usually depend on the rounding errors). For example:

```
>> A = [1 0; 0 0], b = [1 0]', x = A\b
A =
     1     0
     0     0
b =
     1
     0
Warning: Matrix is singular to working precision.

x =
     1
    NaN
```

Backslash take advantage of various kinds of structure in A ; see MATLAB Guide (section 9.3) or `doc mldivide` in MATLAB.

*Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK (nick.higham@manchester.ac.uk).

Overdetermined System: $m > n$

An overdetermined system has no solutions, in general. Backslash yields a least squares (LS) solution, which is unique if A has full rank. If A is rank-deficient then there are infinitely many LS solutions, and backslash returns a *basic solution*: one with at most $\text{rank}(A)$ nonzeros. Such a solution is not, in general, unique.

Underdetermined System: $m < n$

An underdetermined system has fewer equations than unknowns, so either there is no solution or there are infinitely many. In the latter case $A \setminus b$ produces a basic solution and in the former case a basic LS solution. Example:

```
>> A = [1 1 1; 1 1 0]; b = [3 2]'; x = A\b
x =
    2.0000e+00
         0
    1.0000e+00
```

Another basic solution is $[0 \ 2 \ 1]^T$, and the minimum 2-norm solution is $[1 \ 1 \ 1]^T$.

$A \setminus A$

Now we turn to the special case $A \setminus A$, which in terms of equation (1) is a solution to $AX = A$. If $A = 0$ then $X = I$ is not a basic solution, so $A \setminus A \neq I$; in fact, $0 \setminus 0 = 0$ if $m \neq n$ and it is matrix of NaNs if $m = n$.

For an underdetermined system with full-rank A , $A \setminus A$ is not necessarily the identity matrix:

```
>> A = [1 0 1; 0 1 0], X = A\A
A =
     1     0     1
     0     1     0
X =
     1     0     1
     0     1     0
     0     0     0
```

But for an overdetermined system with full-rank A , $A \setminus A$ is the identity matrix:

```
>> A'\A'
ans =
    1.0000e+00         0
   -1.9185e-17    1.0000e+00
```

Minimum Frobenius Norm Solution

The MATLAB definition of $A \setminus B$ is a pragmatic one, as it computes a solution or LS solution to $AX = B$ in the most efficient way, using LU factorization ($m = n$) or QR factorization ($m \neq n$).

Often, one wants the solution of minimum 2-norm, which can be expressed as A^+b , where A^+ is the pseudoinverse of A . In MATLAB, A^+b can be computed by `lsqminnorm(A,b)` or `pinv(A)*b`, the former expression being preferred as it avoids the unnecessary computation of A^+ and it uses a complete orthogonal factorization instead of an SVD. Let us write the minimal 2-norm solution as $A \backslash \backslash B$. Then $A \backslash \backslash A = A^+A$, which is the orthogonal projector onto $\text{range}(A^*)$, and it is equal to the identity matrix when $m \geq n$ and A has full rank. For the matrix above:

```
>> A = [1 0 1; 0 1 0], X = lsqminnorm(A,A)
A =
     1     0     1
     0     1     0
X =
 5.0000e-01         0  5.0000e-01
         0  1.0000e+00         0
 5.0000e-01         0  5.0000e-01
```

Related Blog Posts

- What Is an LU Factorization? (2021)
- What Is a QR Factorization? (2020)
- What Is a Rank-Revealing Factorization? (2021)

This article is part of the “What Is” series, available from <https://nhigham.com/index-of-what-is-articles/> and in PDF form from the GitHub repository <https://github.com/higham/what-is>.