Matrix decompositions

How can we solve

$$A\mathbf{x} = \mathbf{b}$$

?

Linear algebra

Typical linear system of equations :

$$\begin{aligned} 5x_1 - x_2 + 2x_3 &= 7 \\ -2x_1 + 6x_2 + 9x_3 &= 0 \\ -7x_1 + 5x_2 - 3x_3 &= 5 \end{aligned}$$

The variables x_1, x_2, and x_3 only appear as linear terms (no powers or products).

Linear algebra

Where do linear systems come from?

- Fitting curves to data
- Polynomial approximation to functions
- Computational fluid dynamics
- Network flow
- Computer graphics
- Difference equations
- Differential equations
- Dynamical systems theory
- ...

Typical linear system

How does Matlab solve linear systems such as :

$$\begin{aligned} 5x_1 - x_2 + 2x_3 &= 7 \\ -2x_1 + 6x_2 + 9x_3 &= 0 \\ -7x_1 + 5x_2 - 3x_3 &= 5 \end{aligned}$$

- Does such a system always have a solution?
- Can such a system be solved efficiently for millions of equations?
- What does Matlab do if we have more equations than unknowns? More unknowns than equations?

Solving linear systems

We are already familiar with at least one type of linear system

$$\begin{aligned} 3x_1 + 5x_2 &= 3 \\ 2x_1 - 4x_2 &= 1 \end{aligned}$$

The solution is the intersection of the two lines represented by each equation. This solution is a point (x_1, x_2) that satisfies both equations *simultaneously*.

We could also view the solution as providing the correct linear combination of vectors $(3, 2)$ and $(5, -4)$ that give us $(3, 1)$.

$$x_1 \begin{bmatrix} 3 \\ 2 \end{bmatrix} + x_2 \begin{bmatrix} 5 \\ -4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

Linear algebra - a 2x2 system

We can *row-reduce* an augmented matrix to find the solution :

Use an elementary row operation to produce a "0" in the lower left corner.

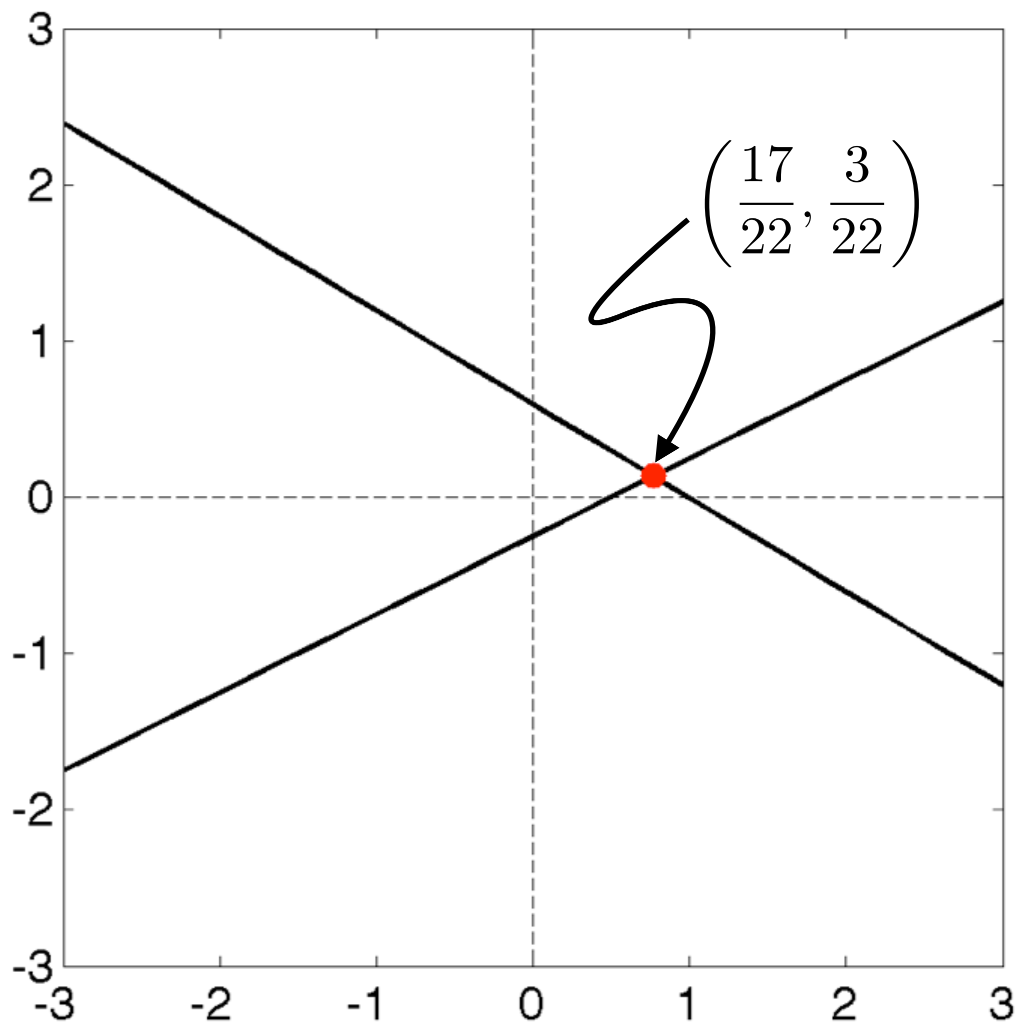
$$\left[\begin{array}{cc|c} 3 & 5 & 3 \\ 2 & -4 & 1 \end{array}\right] \rightarrow \left[\begin{array}{cc|c} 3 & 5 & 3 \\ 0 & -\frac{22}{3} & -1 \end{array}\right] \leftarrow (\text{eqn } 2) - \left(\frac{2}{3}\right)(\text{eqn } 1)$$

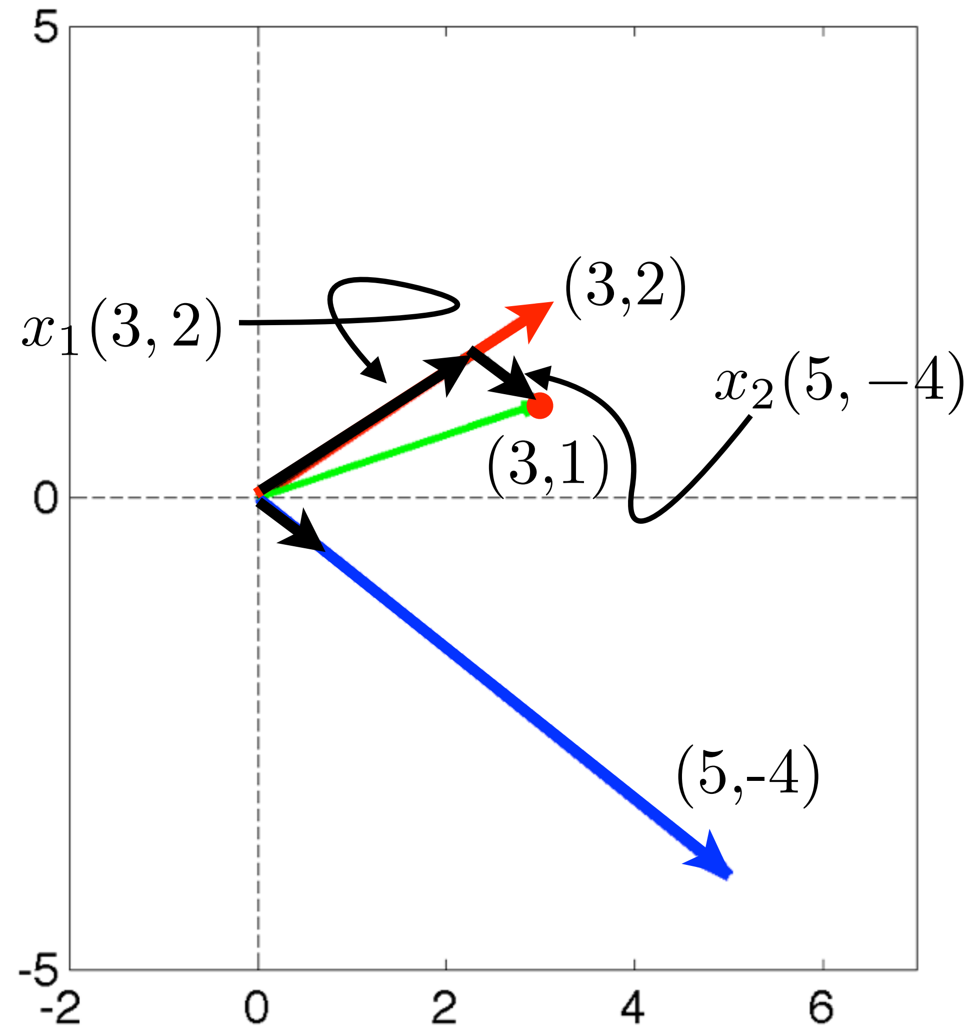
Use back-substitution to solve first for x_2 and then for x_1.

$$\begin{aligned} x_2 &= \left(\frac{-3}{22}\right)(-1) = \frac{3}{22} \\ x_1 &= \frac{1}{3}(3 - 5x_2) = \frac{1}{3}\left(3 - 5\left(\frac{-3}{22}\right)\right) = \frac{17}{22} \end{aligned}$$

Solution : $\quad x_1 = \frac{17}{22}, \qquad x_2 = \frac{3}{22}$

Linear algebra - a 2x2 system





Solution as the intersection of two lines

Solution as linear combination of vectors

Gaussian Elimination

Apply elementary row operations to the augmented matrix to zero out entries below the diagonal and reduce the system to an upper triangular system.

$$\left[\begin{array}{rrr|r} 5 & -1 & 2 & 7 \\ -2 & 6 & 9 & 0 \\ -7 & 5 & -3 & 5 \end{array}\right]$$

$$\left[\begin{array}{rrr|r} 5 & -1 & 2 & 7 \\ 0 & \frac{28}{5} & \frac{49}{5} & \frac{14}{5} \\ 0 & \frac{18}{5} & -\frac{1}{5} & \frac{74}{5} \end{array}\right] \begin{array}{l} \\ \leftarrow (\textbf{eqn 2}) - \left(\frac{-2}{5}\right)(\textbf{eqn 1}) \\ \leftarrow (\textbf{eqn 3}) - \left(\frac{-7}{5}\right)(\textbf{eqn 1}) \end{array}$$

$$\left[\begin{array}{rrr|r} 5 & -1 & 2 & 7 \\ 0 & \frac{28}{5} & \frac{49}{5} & \frac{14}{5} \\ 0 & \boxed{0} & -\frac{65}{10} & \frac{65}{5} \end{array}\right] \begin{array}{l} \\ \\ \leftarrow (\textbf{eqn 3}) - \left(\frac{9}{14}\right)(\textbf{eqn 2}) \end{array}$$

Solve upper triangular system for x

$$\begin{bmatrix} 5 & -1 & 2 \\ 0 & \frac{28}{5} & \frac{49}{5} \\ 0 & 0 & -\frac{65}{10} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ \frac{14}{5} \\ \frac{65}{5} \end{bmatrix}$$

Notice that the right hand side is not the right hand side of the original system

Use back-substitution to solve for x :

$$\textbf{Step 1 :}\ x_3 = \left(\frac{-10}{65}\right)\left(\frac{65}{5}\right) = -2$$

Known from previous steps

$$\textbf{Step 2 :}\ x_2 = \left(\frac{5}{28}\right)\left(\frac{14}{5} - \frac{49}{5}x_3\right) = 4$$

$$\textbf{Step 3 :}\ x_1 = \left(\frac{1}{5}\right)(7 - (-1)x_2 - (2)x_3) = 3$$

Some notation

$$\left[\begin{array}{rrr|r} 5 & -1 & 2 & 7 \\ -2 & 6 & 9 & 0 \\ -7 & 5 & -3 & 5 \end{array}\right]$$

Apply elementary row operations to the augmented matrix to zero out entries below the diagonal and reduce the system to an upper triangular system.

$$\left[\begin{array}{rrr|r} 5 & -1 & 2 & 7 \\ 0 & \frac{28}{5} & \frac{49}{5} & \frac{14}{5} \\ 0 & \frac{18}{5} & -\frac{1}{5} & \frac{74}{5} \end{array}\right] \begin{array}{l} \\ \leftarrow (\textbf{eqn 2}) - \left(\frac{-2}{5}\right)(\textbf{eqn 1}) \\ \leftarrow (\textbf{eqn 3}) - \left(\frac{-7}{5}\right)(\textbf{eqn 1}) \end{array}$$

$$\left[\begin{array}{rrr|r} 5 & -1 & 2 & 7 \\ 0 & \frac{28}{5} & \frac{49}{5} & \frac{14}{5} \\ 0 & 0 & -\frac{65}{10} & \frac{65}{5} \end{array}\right] \begin{array}{l} \\ \\ \leftarrow (\textbf{eqn 3}) - \left(\frac{9}{14}\right)(\textbf{eqn 2}) \end{array}$$

"Pivots"

"Multipliers"

Cost of Gaussian elimination

The cost of eliminating entries below the 1^{st} pivot:

Pivot

$$\left[\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & b_4 \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & b_5 \end{array}\right] \qquad n = 5$$

Cost of Gaussian elimination

The cost of eliminating entries below the 1st pivot:

Pivot

$$\left[\begin{array}{ccccc|c} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & a_{14}^{(0)} & a_{15}^{(0)} & b_1^{(0)} \\ \color{red}{a_{21}^{(0)}} & a_{22}^{(0)} & a_{23}^{(0)} & a_{24}^{(0)} & a_{25}^{(0)} & b_2^{(0)} \\ \color{red}{a_{31}^{(0)}} & a_{32}^{(0)} & a_{33}^{(0)} & a_{34}^{(0)} & a_{35}^{(0)} & b_3^{(0)} \\ \color{red}{a_{41}^{(0)}} & a_{42}^{(0)} & a_{43}^{(0)} & a_{44}^{(0)} & a_{45}^{(0)} & b_4^{(0)} \\ \color{red}{a_{51}^{(0)}} & a_{52}^{(0)} & a_{53}^{(0)} & a_{54}^{(0)} & a_{55}^{(0)} & b_5^{(0)} \end{array}\right] \qquad n = 5$$

Each row operation costs $n+1$ multiplies/divides and n subtractions:

$$\text{Total operations per row: } 2n+1$$

There are $n-1$ rows to eliminate:

$$\text{Total operations: } (2n+1)(n-1) = 2n(n-1) + n - 1$$

Cost of Gaussian elimination

The cost of eliminating entries below the 2^{nd} pivot:

$$\left[\begin{array}{ccccc|c} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & a_{14}^{(0)} & a_{15}^{(0)} & b_1^{(0)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} & b_2^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & a_{34}^{(1)} & a_{35}^{(1)} & b_3^{(1)} \\ 0 & a_{42}^{(1)} & a_{43}^{(1)} & a_{44}^{(1)} & a_{45}^{(1)} & b_4^{(1)} \\ 0 & a_{52}^{(1)} & a_{53}^{(1)} & a_{54}^{(1)} & a_{55}^{(1)} & b_5^{(1)} \end{array}\right] \qquad n = 5$$

Pivot

Each row operation costs n multiplies/divides and $n-1$ subtractions:

$$\text{Total operations per row: } 2n-1$$

There are $n-2$ rows to eliminate:

$$\text{Total operations: } (2n-1)(n-2) = 2(n-1)(n-2) + n - 2$$

Cost of Gaussian elimination

The cost of eliminating entries below the 3^{rd} pivot:

Pivot

$$\left[\begin{array}{ccccc|c}
a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & a_{14}^{(0)} & a_{15}^{(0)} & b_1^{(0)} \\
0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} & b_2^{(1)} \\
0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} & b_3^{(2)} \\
0 & 0 & \color{red}{a_{43}^{(2)}} & a_{44}^{(2)} & a_{45}^{(2)} & b_4^{(2)} \\
0 & 0 & \color{red}{a_{53}^{(2)}} & a_{54}^{(2)} & a_{55}^{(2)} & b_5^{(2)}
\end{array}\right] \qquad n = 5$$

Each row operation costs $n-1$ multiplies/divides and $n-2$ subtracts:

$$\text{Total operations per row: } 2n-3$$

There are $n-3$ rows to eliminate:

$$\text{Total operations: } (2n-3)(n-3) = 2(n-2)(n-3) + n - 3$$

Cost of Gaussian elimination

The cost of eliminating entries below the 4^{th} pivot:

$$\left[\begin{array}{ccccc|c} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & a_{14}^{(0)} & a_{15}^{(0)} & b_1^{(0)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} & b_3^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} & b_4^{(3)} \\ 0 & 0 & 0 & a_{54}^{(3)} & \boxed{a_{55}^{(3)}} & \boxed{b_5^{(3)}} \end{array}\right] \qquad n = 5$$

Pivot

Each row operation costs $n-2$ multiplies/divides and $n-3$ subtracts:

$$\text{Total operations per row: } 2n-5$$

There are $n-4$ rows to eliminate:

$$\text{Total operations: } (2n-5)(n-4) = 2(n-3)(n-4) + n - 4$$

Cost of Gaussian elimination

After the $(n-1)^{\mathrm{st}}$ step the matrix is in upper triangular form:

$$\left[\begin{array}{ccccc|c} a_{11}^{(0)} & a_{12}^{(0)} & a_{13}^{(0)} & a_{14}^{(0)} & a_{15}^{(0)} & b_1^{(0)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & a_{24}^{(1)} & a_{25}^{(1)} & b_2^{(1)} \\ 0 & 0 & a_{33}^{(2)} & a_{34}^{(2)} & a_{35}^{(2)} & b_3^{(2)} \\ 0 & 0 & 0 & a_{44}^{(3)} & a_{45}^{(3)} & b_4^{(3)} \\ 0 & 0 & 0 & 0 & a_{55}^{(4)} & b_5^{(4)} \end{array}\right] \qquad n = 5$$

Cost of Gaussian elimination

We can count the total floating point operations (FLOPs):

$$
\begin{aligned}
\text{Total FLOPs} &= \sum_{k=1}^{n} \left(2(n-k+1)(n-k) + (n-k)\right) \\
&= 2\sum_{k=1}^{n-1}(n-k+1)(n-k) + \sum_{k=1}^{n-1}(n-k) \\
&= \frac{2}{3}n(n^2-1) + \frac{1}{2}n(n-1) \\
&= \frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n \\
&\approx \frac{2}{3}n^3
\end{aligned}
$$

we ignore lower order powers of n

We say that elimination is an n^3 algorithm, which is considered expensive.

What about the back solve?

$$\begin{bmatrix} a_{11} & a_{12} & a_{a13} & a_{14} & a_{15} \\ 0 & a_{22} & a_{23} & a_{24} & a_{25} \\ 0 & 0 & a_{33} & a_{34} & a_{35} \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & a_{55} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}$$

One 'op' is a multiplication or divide; Ignore subtractions for now; we will add them in momentarily.

$$x_5 = b_5/a_{55} \quad \rightarrow \quad 1 \text{ op}$$

$$x_4 = (b_4 - a_{45}x_5)/a_{44} \quad \rightarrow \quad 2 \text{ ops}$$

$$x_3 = (b_3 - a_{34}x_x - a_{35}x_5)/a_{33} \quad \rightarrow \quad 3 \text{ ops}$$

$$x_2 = (b_2 - a_{23}x_3 - a_{24}x_x - a_{25}x_5)/a_{22} \quad \rightarrow \quad 4 \text{ ops}$$

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3 - a_{14}x_x - a_{15}x_5)/a_{11} \quad \rightarrow \quad 5 \text{ ops}$$

What about the cost of the back solve?

Step k in the back solve requires k multiplications and k additions. So the total work for a back solve is :

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2} \approx \frac{1}{2}n^2$$

Ignore lower order powers of n, but pay attention to the coefficient of the highest order power of n.

or $2\left(\frac{1}{2}n^2\right) = n^2$ if we include subtractions, as well as divisions and multiplications.

We say that a back solve is an "order n^2" operation, which is considerably cheaper than the original elimination.

The LU decomposition

If we have more than one right hand side (as is often the case)

$$A\mathbf{x} = \mathbf{b}_i, \quad i = 1, 2, \ldots, M$$

We can actually store the work involved carrying out the elimination by storing the multipliers used to carry out the row operations.

LU Decomposition

$$U = \begin{bmatrix} 5 & -1 & 2 \\ 0 & \frac{28}{5} & \frac{49}{5} \\ 0 & 0 & -\frac{65}{10} \end{bmatrix}$$

Store the multipliers in a lower triangular matrix :

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{2}{5} & 1 & 0 \\ -\frac{7}{5} & \frac{9}{14} & 1 \end{bmatrix}$$

LU Decomposition

The product LU is equal to A :

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -\frac{2}{5} & 1 & 0 \\ -\frac{7}{5} & \frac{9}{14} & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 5 & -1 & 2 \\ 0 & \frac{28}{5} & \frac{49}{5} \\ 0 & 0 & -\frac{65}{10} \end{bmatrix}}_{U} = \underbrace{\begin{bmatrix} 5 & -1 & 2 \\ -2 & 6 & 9 \\ -7 & 5 & -3 \end{bmatrix}}_{A}$$

It does not cost us anything to store the multipliers. But by doing so, we can now solve many systems involving the matrix A.

Solution procedure given LU=A

How can we solve a system using the LU factorization?

$$A\mathbf{x} = \mathbf{b}$$

$$\begin{aligned}
&\textbf{Step 0 :} \quad \text{Factor } A \text{ into } LU && \leftarrow \text{Row-reduction} \\
&\textbf{Step 1 :} \quad \text{Solve } L\mathbf{y} = \mathbf{b} && \leftarrow \text{Forward substitution} \\
&\textbf{Step 2 :} \quad \text{Solve } U\mathbf{x} = \mathbf{y} && \leftarrow \text{Back substitution}
\end{aligned}$$

For each right hand side, we only need to do n^2 operations. The expensive part is forming the original LU decomposition.

Cost of a matrix inverse

To solve using the matrix inverse A^{-1} to get $\mathbf{x} = A^{-1}\mathbf{b}$.
To get a column c_j of the matrix A^{-1}, we solve

$$Ac_j = e_j$$

for each column e_j of an identity matrix.

$$\text{total cost} \approx \frac{2}{3}n^3 + 2n^3 = \frac{8}{3}n^3 \quad \text{operations}$$

It costs about 4 times as much to multiply by the inverse as it does to solve the linear system using Gaussian elimination.
The cost of the matrix vector multiply $A^{-1}\mathbf{b}$ is n^2.

Row exchanges

What if we start with a system that looks like :

$$A = \begin{bmatrix} 0 & 0 & -\frac{65}{10} \\ 0 & \frac{28}{5} & \frac{49}{5} \\ 5 & -1 & 2 \end{bmatrix}$$

All we need to do is exchange the rows of A, and do the decomposition on

$$LU = PA$$

where P is a *permutation* matrix, i.e.

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Partial pivoting

We can also do row exchanges not just to avoid a zero pivot, but also to make the pivot as large as possible. This is called "partial pivoting".

Find the largest pivot in the entire column, and do a row exchange.

$$\begin{bmatrix} X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \end{bmatrix}$$

One can also do "full pivoting" by looking for the largest pivot in the entire matrix. But this is rarely done.

Top 10 algorithms

The matrix decompositions are listed as one of the top 10 algorithms of the 20th century

from *SIAM News*, Volume 33, Number 4

The Best of the 20th Century: Editors Name Top 10 Algorithms

1951: Alston Householder of Oak Ridge National Laboratory formalizes the **decompositional approach to matrix computations**.

The ability to factor matrices into triangular, diagonal, orthogonal, and other special forms has turned out to be extremely useful. The decompositional approach has enabled software developers to produce flexible and efficient matrix packages. It also facilitates the analysis of rounding errors, one of the big bugbears of numerical linear algebra. (In 1961, James Wilkinson of the National Physical Laboratory in London published a seminal paper in the *Journal of the ACM*, titled "Error Analysis of Direct Methods of Matrix Inversion," based on the LU decomposition of a matrix as a product of lower and upper triangular factors.)