Solutions of 3×3 Systems of Linear Equations Presented on a Golden Plate

Author: Lamjahdi, Mohamed El Mami

Email: lamjahdimohamedelmami@gmail.com, lamjahdi@stud.uni-hannover.de

Date: 05.07.2025

1. Abstract

This paper presents a direct solution method for 3×3 systems of linear equations. The derivation is based on a sketch of the Reverse-2X Method introduced in [LAM25], as well as the Reverse Gauss-Jordan Algorithm. Owing to the requirement of only a single mathematical operation to determine the third variable, the proposed solution outperforms all state-of-the-art methods, including the 4X Method described in [LAM25].

2. Introduction

Solving systems of linear equations is one of the most common and repetitive tasks encountered in technical university programs. Many electrical circuits and mechanical structures can be modeled as systems of linear equations, typically in 2×2 or 3×3 form. For instance, control processes—such as mechanical deep rolling—often require the solution of such systems. Controllers frequently depend on predictors to compensate for dead time between system input and process response. These predictors rely on continuously solving linear relationships among historical signal data. Controlling processes subject to unpredictable changes and external disturbances demands real-time solutions. The need for such solutions highlights the algorithmic complexity involved, especially when time is the most valuable resource.
In high-precision industries, such as nanometer-scale circuit printing on semiconductor wafers, the controller's response time is critical. In both of his papers, [LAM25] presents four novel methods for solving systems of linear equations in constant time. Particular attention is given to the 4X Method, which generalizes the 2X Method to $n \times n$ systems (e.g., the 6X Algorithm for 4×4 systems in [LAM24]). However, the 4X Method involves a sequence of operations: variable shifting, system reduction, term equating, variable solving, and back-substitution. While theoretically achievable in constant time, meeting high standards of precision and execution speed renders the shifting and equating steps increasingly time-consuming.

In computational environments such as MATLAB or Octave, these steps typically require symbolic variables from the *syms* library and additional algorithms to perform the solving process—often without explicitly accessing the numerical matrix elements. In contrast, this paper introduces a direct solution approach that eliminates the need for variable shifting and symbolic manipulation. The method relies solely on direct access to the numerical array, enabling a more efficient and precise solution process.

3. The Direct Solutions

Given is a 3x3 Linear System in Matrix Form:

$$M \cdot \vec{x}=\vec{e}, \text{ with:} \quad M=\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}, \quad \vec{x}=\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad \vec{e}=\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix}$$

The solutions are:

$$x_1=\frac{(a_2\cdot c_3-c_2\cdot a_3)\cdot(e_2\cdot c_3-e_3\cdot b_3)-(b_2\cdot c_3-c_2\cdot b_3)\cdot(e_1\cdot c_3-e_3\cdot a_3)}{(a_2\cdot c_3-c_2\cdot a_3)\cdot(b_1\cdot a_3-c_1\cdot b_3)-(b_2\cdot c_3-c_2\cdot b_3)\cdot(a_1\cdot c_3-c_1\cdot a_3)}$$

$$x_2=\frac{(a_1\cdot c_3-c_1\cdot a_3)\cdot(e_2\cdot c_3-e_3\cdot b_3)-(b_1\cdot c_3-c_1\cdot b_3)\cdot(e_1\cdot c_3-e_3\cdot a_3)}{(a_1\cdot c_3-c_1\cdot a_3)\cdot(b_2\cdot a_3-c_1\cdot b_3)-(b_1\cdot c_3-c_1\cdot b_3)\cdot(a_2\cdot c_3-c_2\cdot a_3)}$$

Once x_1 and x_2 have been determined, x_3 can be computed directly.

$$x_3=\frac{1}{c_3}\cdot(e_3-c_1 x_1-c_2\cdot x_2)$$

To minimize memory reuse and redundant computations—and thereby further reduce the time complexity—it is appropriate to define new constants:

$$T_1=a_1\cdot c_3-c_1\cdot a_3, \quad T_2=a_2\cdot c_3-c_1\cdot a_3, \quad T_3=e_1\cdot c_3-e_3\cdot a_3,$$

$$T_4=b_1\cdot c_3-c_1\cdot b_3, \quad T_5=b_2\cdot c_3-c_2\cdot b_3, \quad T_6=e_2\cdot c_3-e_3\cdot b_2.$$

The solutions can then be computed as follows:

$$x_1=\frac{(T_2\cdot T_6)-(T_5\cdot T_3)}{(T_2\cdot T_4)-(T_5\cdot T_1)}, \qquad x_2=\frac{(T_1\cdot T_6)-(T_4\cdot T_3)}{(T_1\cdot T_5)-(T_4\cdot T_2)}$$

The value of x3x_3x3 can then be computed as previously described, and the solvability of the system is reflected in the denominators.

4. The Proof

The direct solution comprises the Reverse 2X Method in the initial stage, followed by two steps derived from the Reverse Gauss-Jordan Algorithm.

To begin, consider reconstructing the 2X scheme based on the given solutions in the first part:

$$x_1=\frac{(T_2\cdot T_6)-(T_5\cdot T_3)}{(T_2\cdot T_4)-(T_5\cdot T_1)},\qquad x_2=\frac{(T_1\cdot T_6)-(T_4\cdot T_3)}{(T_1\cdot T_5)-(T_4\cdot T_2)}$$

$$\boxed{x_1=\left[\begin{array}{c|cc} T_3 & T_2 & T_1 \\ T_6 & T_5 & T_4 \end{array}\right]}\qquad \boxed{x_2=\left[\begin{array}{c|cc} T_3 & T_1 & T_2 \\ T_6 & T_4 & T_5 \end{array}\right]}$$

$$\Leftrightarrow\left[\begin{array}{c|cc} e_1\cdot c_3-e_3\cdot a_3 & a_1\cdot c_3-c_1\cdot a_3 & a_2\cdot c_3-c_1\cdot a_3 \\ e_2\cdot c_3-e_3\cdot b_2 & b_1\cdot c_3-c_1\cdot b_3 & b_2\cdot c_3-c_2\cdot b_3 \end{array}\right]$$

Adding x_3's equation:

$$\Leftrightarrow\left[\begin{array}{ccc|c} a_1\cdot c_3-c_1\cdot a_3 & a_2\cdot c_3-c_1\cdot a_3 & 0 & e_1\cdot c_3-e_3\cdot a_3 \\ b_1\cdot c_3-c_1\cdot b_3 & b_2\cdot c_3-c_2\cdot b_3 & 0 & e_2\cdot c_3-e_3\cdot b_2 \\ \frac{c_1}{c_3} & \frac{c_2}{c_3} & 1 & \frac{e_3}{c_3} \end{array}\right]$$

Dividing first row by $a_3\cdot c_3$ and second one by $b_3\cdot c_3$:

$$\Leftrightarrow\left[\begin{array}{ccc|c} \frac{a_1\cdot c_3-c_1\cdot a_3}{a_3\cdot c_3} & \frac{a_2\cdot c_3-c_1\cdot a_3}{a_3\cdot c_3} & 0 & \frac{e_1\cdot c_3-e_3\cdot a_3}{a_3\cdot c_3} \\ \frac{b_1\cdot c_3-c_1\cdot b_3}{b_3\cdot c_3} & \frac{b_2\cdot c_3-c_2\cdot b_3}{b_3\cdot c_3} & 0 & \frac{e_2\cdot c_3-e_3\cdot b_2}{b_3\cdot c_3} \\ \frac{c_1}{c_3} & \frac{c_2}{c_3} & 1 & \frac{e_3}{c_3} \end{array}\right]$$

Adding third row to first and second:

$$\Leftrightarrow \begin{bmatrix} \frac{a_1}{a_3} & \frac{a_2}{a_3} & 1 & | & \frac{e_1}{a_3} \\ \frac{b_1}{b_3} & \frac{b_2}{b_3} & 1 & | & \frac{e_2}{b_3} \\ \frac{c_1}{c_3} & \frac{c_2}{c_3} & 1 & | & \frac{e_3}{c_3} \end{bmatrix}$$

By multiplying each row by its corresponding third element, we obtain:

$$\Leftrightarrow \begin{bmatrix} a_1 & a_2 & a_3 & | & e_1 \\ b_1 & b_2 & b_3 & | & e_2 \\ c_1 & c_2 & c_3 & | & e_3 \end{bmatrix}$$

Which is the original system.

5. Conclusion

The method proposed in this paper offers control engineers the fastest possible direct solution for 3×3 systems of linear equations, with performance that cannot be surpassed. Additionally, it aligns with the formal definition of a 'direct' solution as established in [LAM25].

References

[LAM25] What If Linear Equations Could Be Solved Instantly? DOI: 10.5281/zenodo.15838111

6X-Method for solving 4x4 Linear Equations in O(1), DOI: 10.5281/zenodo.15838306