1.2 Gaussian elimination

In its simplest form, Gaussian elimination proceeds just like the reduction to echelon form in the previous section.

Example 1: Solve the system
$$\begin{cases} x_1 & -2x_2 & +2x_3 & = 1 \\ -x_1 & -x_2 & +3x_3 & = 1 \\ 3x_1 & +x_2 & -2x_3 & = 2 \end{cases}$$

Writing down only the coefficients, and proceeding by removing the entries below the main diagonal in successive columns gives

$$\left[\begin{array}{ccc|c} 1 & -2 & 2 & 1 \\ -1 & -1 & 3 & 1 \\ 3 & 1 & -2 & 2 \end{array}\right] \Rightarrow \left\{\begin{array}{c}\text{add multiples 1 and} \\ \text{-3 resp. of top row} \\ \text{to two rows below}\end{array}\right\} \Rightarrow \left[\begin{array}{ccc|c} 1 & -2 & 2 & 1 \\ & -3 & 5 & 2 \\ & 7 & -8 & -1 \end{array}\right] \Rightarrow \left\{\begin{array}{c}\text{multiply} \\ \text{second row} \\ \text{by } -\frac{1}{3}\end{array}\right\} \Rightarrow \left[\begin{array}{ccc|c} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & 7 & -8 & -1 \end{array}\right] \Rightarrow \tag{1}$$

$$\Rightarrow \left\{\begin{array}{c}\text{add -7 times} \\ \text{second row to} \\ \text{third row}\end{array}\right\} \Rightarrow \left[\begin{array}{ccc|c} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & & \frac{11}{3} & \frac{11}{3} \end{array}\right] \Rightarrow \left\{\begin{array}{c}\text{multiply} \\ \text{last row} \\ \text{by } \frac{3}{11}\end{array}\right\} \Rightarrow \left[\begin{array}{ccc|c} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & & 1 & 1 \end{array}\right].$$

Back substitution now proceeds by noting that the last equation tells us that $x_3 = 1$; knowing this, middle equation tells that $x_2 = 1$, and top equation that also $x_1 = 1$.

◊

This basic elimination process is not quite satisfactory as it stands:

1. If the top left element (first *pivot*) had been zero, no multiple of that row could have eliminated entries lower down in the first column. Similar break-downs could have occurred at any later stage. Or if a pivot is very small, a very large multiple of that row would have to be used - this looses accuracy in floating point,

2. It is quite common that one needs to solve a system with many right hand sides (RHSs). If they are all known initially, we can just place them side-by-side when we eliminate as above. But it is also common that they become known only one after another. We then do not want to have to repeat the work on the coefficient part.

3. The coefficient matrix has often a lot of zero elements, e.g. it may be tri-diagonal or banded, maybe with some additional elements present as well. Such structures can often be exploited to greatly reduce both operation count and storage.

We now address these issues in turn:

1. The solution of a linear system is totally independent of in which order the equations have been written down. The usually satisfactory method of *partial pivoting* simply amounts to, at stage k, exchanging row k with whatever row below it that has the largest element in the k^{th} position (thus never needing to use multiples in magnitude greater than one in the elimination steps - this makes growth if intermediate quantities less likely). The elimination process leaves the determinant of the coefficient matrix unchanged - its value can be read off as $(-1)^p$ times the product of their pivots (the diagonal elements of the resulting lower triangular matrix); p denotes how many times we needed to swap order of the equations. Breakdown of partial pivoting will occur if and only if the system is singular.

2. Although not obvious from how we presented the solution of Example 1 above, what we achieved was in fact a factorization of the original matrix

$$A = \begin{bmatrix} 1 & -2 & 2 \\ -1 & -1 & 3 \\ 3 & 1 & -2 \end{bmatrix}$$

into a product $A = L\ U$ where L is lower triangular, and U upper triangular. We were left with the U-matrix

$$U = \begin{bmatrix} 1 & -2 & 2 \\ & 1 & -\frac{5}{3} \\ & & 1 \end{bmatrix} ,$$

and just did not pay attention to the fact that

$$L = \begin{bmatrix} 1 & & \\ -1 & -3 & \\ 3 & 7 & \frac{11}{3} \end{bmatrix}$$

could have been written down by just taking note that these entries appeared at just these places during the elimination (Verify that the matrix product $L\ U$ indeed recovers A!).

The reason this happened ($LU = A$) can be seen if we interpret every step we took as a matrix multiplication. Described in this way, (1) reads

$$\begin{bmatrix} 1 & & \\ 1 & 1 & \\ -3 & & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 2 & 1 \\ -1 & -1 & 3 & 1 \\ 3 & 1 & -2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 & 1 \\ & -3 & 5 & 2 \\ & 7 & -8 & -1 \end{bmatrix}$$

$$\sim\sim\sim\sim \quad \leftarrow \quad \sim\sim\sim$$

$$\begin{bmatrix} 1 & & \\ & -\frac{1}{3} & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 2 & 1 \\ & -3 & 5 & 2 \\ & 7 & -8 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & 7 & -8 & -1 \end{bmatrix}$$

~ ~ ~ ~ $\leftarrow$ ~ ~ ~

$$\begin{bmatrix} 1 & & \\ & 1 & \\ & 7 & 1 \end{bmatrix}\begin{bmatrix} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & 7 & -8 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & & \frac{11}{3} & \frac{11}{3} \end{bmatrix}$$

~ ~ ~ ~ $\leftarrow$ ~ ~ ~

$$\begin{bmatrix} 1 & & \\ & 1 & \\ & & \frac{3}{11} \end{bmatrix}\begin{bmatrix} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & & \frac{11}{3} & \frac{11}{3} \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & & 1 & 1 \end{bmatrix}$$

In summary:

$$\begin{bmatrix} 1 & & \\ & 1 & \\ & & \frac{3}{11} \end{bmatrix}\begin{bmatrix} 1 & & \\ & 1 & \\ & -7 & 1 \end{bmatrix}\begin{bmatrix} 1 & & \\ & -\frac{1}{3} & \\ & & 1 \end{bmatrix}\begin{bmatrix} 1 & & \\ 1 & 1 & \\ -3 & & 1 \end{bmatrix}\begin{bmatrix} 1 & -2 & 2 & 1 \\ -1 & -1 & 3 & 1 \\ 3 & 1 & -2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & & 1 & 1 \end{bmatrix}$$

The inverses of each of the square matrices on the left are immediate: for a diagonal matrix, one inverts the diagonal elements. And for an identity matrix with one subdiagonal column, one just swaps the sign on those subdiagonal entries. Multiplying with the inverses from the left therefore gives

$$\begin{bmatrix} 1 & -2 & 2 & 1 \\ -1 & -1 & 3 & 1 \\ 3 & 1 & -2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ 3 & & 1 \end{bmatrix}\begin{bmatrix} 1 & & \\ & -3 & \\ & & 1 \end{bmatrix}\begin{bmatrix} 1 & & \\ & 1 & \\ & 7 & 1 \end{bmatrix}\begin{bmatrix} 1 & & \\ & 1 & \\ & & \frac{11}{3} \end{bmatrix}\begin{bmatrix} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & & 1 & 1 \end{bmatrix}$$

Last thing to note is that for the four square matrices which starting the right hand side, their structure and order is such that we get their product just by placing the nontrivial elements together in one single matrix. We have therefore

$$\begin{bmatrix} 1 & -2 & 2 & 1 \\ -1 & -1 & 3 & 1 \\ 3 & 1 & -2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & & \\ -1 & -3 & \\ 3 & 7 & \frac{11}{3} \end{bmatrix}\begin{bmatrix} 1 & -2 & 2 & 1 \\ & 1 & -\frac{5}{3} & -\frac{2}{3} \\ & & 1 & 1 \end{bmatrix}$$

This repeats just the same in the general case - we have obtained a genuine *LU*-factorization of *A* and the entries of L can indeed be picked straight out from the numbers 'left behind' during the Gaussian elimination.

In the form $L\,U\,\underline{x} = \underline{b}$, the linear system is quickly solved by two successive back substitutions: First $L\,\underline{x}_1 = \underline{b}$ and then $U\,\underline{x} = \underline{x}_1$. *L* and *U* can both be fitted back into the memory *A* used to occupy - the non-trivial elements of *L* are obtained at the same time as zeros are introduced in the corresponding positions of *A* (as it is turning into *U*). This factorization will take $O(n^3)$ operations for an $n\times n$-matrix, but from the form $A = L\,U$, the solution for each new RHS will take just $O(n^2)$ operations .

Incorporating partial pivoting (necessary for numerical stability) complicates the factorization only a little bit - we must now instead aim for $A = P\,L\,U$, where *P* is a permutation matrix (one "1" in each row and column, "0"s otherwise). Besides *A* (which will turn into a composite of *L* and *U*, we also start by putting 1,2,3,4,... into a vector $\underline{c}$, and permute the entries in this vector at the same time as we permute in A (but we do no elimination in this vector). We give just brief comments on the steps:

$$\begin{bmatrix} 1 & -2 & 2 \\ -1 & -1 & 3 \\ 3 & 1 & -2 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \underset{\text{pivot}}{\Rightarrow} \begin{bmatrix} \boxed{3} & \boxed{1} & \boxed{-2} \\ -1 & -1 & 3 \\ \boxed{1} & \boxed{-2} & \boxed{2} \end{bmatrix}\begin{bmatrix} \boxed{3} \\ 2 \\ \boxed{1} \end{bmatrix} \underset{\text{divide top row}}{\Rightarrow} \begin{bmatrix} 3 & \boxed{\frac{1}{3}} & \boxed{-\frac{2}{3}} \\ -1 & -1 & 3 \\ 1 & -2 & 2 \end{bmatrix}\begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \underset{\text{eliminate}}{\Rightarrow} \begin{bmatrix} 3 & \frac{1}{3} & -\frac{2}{3} \\ -1 & \boxed{-\frac{2}{3}} & \boxed{\frac{7}{3}} \\ 1 & \boxed{-\frac{7}{3}} & \boxed{\frac{8}{3}} \end{bmatrix}\begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \underset{\text{pivot}}{\Rightarrow}$$

$$\begin{bmatrix} 3 & \frac{1}{3} & -\frac{2}{3} \\ \boxed{1} & \boxed{-\frac{7}{3}} & \boxed{\frac{8}{3}} \\ \boxed{-1} & \boxed{-\frac{2}{3}} & \boxed{\frac{7}{3}} \end{bmatrix}\begin{bmatrix} 3 \\ \boxed{1} \\ \boxed{2} \end{bmatrix} \underset{\text{divide pivot row}}{\Rightarrow} \begin{bmatrix} 3 & \frac{1}{3} & -\frac{2}{3} \\ 1 & -\frac{7}{3} & \boxed{-\frac{8}{7}} \\ -1 & -\frac{2}{3} & \frac{7}{3} \end{bmatrix}\begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} \underset{\text{eliminate}}{\Rightarrow} \begin{bmatrix} 3 & \frac{1}{3} & -\frac{2}{3} \\ 1 & -\frac{7}{3} & -\frac{8}{7} \\ -1 & -\frac{2}{3} & \boxed{\frac{11}{7}} \end{bmatrix}\begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} \ \text{finished}$$

Notes: 1. The shading illustrates how the L-matrix emerges - it is obtained by just leaving behind old values that otherwise would have been set to "1" or "0" in the elimination,

2. Elements that undergo any change in a particular step are 'boxed',

3. When we pivot, we pivot right through everything, including the L-matrix and the vector $\underline{c}$.

Looking at the very last matrix, we read off the factorization of A as

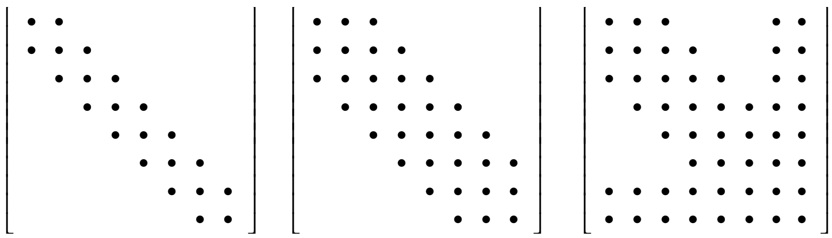
$$A = \begin{bmatrix} & 1 & \\ & & 1 \\ 1 & & \end{bmatrix}\begin{bmatrix} 3 & & \\ 1 & -\frac{7}{3} & \\ -1 & -\frac{2}{3} & \frac{11}{7} \end{bmatrix}\begin{bmatrix} 1 & \frac{1}{3} & -\frac{2}{3} \\ & 1 & -\frac{8}{7} \\ & & 1 \end{bmatrix}$$

P	L	U
Enter the "1"s in rows 3, 1, 2 resp. since $\underline{c}$ contains these numbers as entries	Shaded part of matrix above	Top right corner of end matrix, with "1"s inserted in diagonal

The first step in solving $A\ \underline{x} = \underline{b}$ is to solve $P\ \underline{x}_1 = \underline{b}$; simply a re-ordering of the RHS elements (no need to ever store the P-matrix as $n\times n$ matrix!). Then solve (back substitute) $L\ \underline{x}_2 = \underline{x}_1$, and finally $U\ \underline{x} = \underline{x}_2$.

3. The three sparsity patterns that arise most frequently are

a. tridiagonal,
b. banded, and
c. banded with some extra, full rows and/or columns (or corners).



a. b. c.

Figure 1. Three common matrix structures for linear systems.

Although much more complex sparsity patterns arise in many applications, strategies for them fall well outside the scope of this summary. For the three cases above:

a. If the matrix is not *diagonally dominant*

$$|a_{i,i}| \geq |a_{i,i-1}| + |a_{i,i+1}|$$

we need to pivot. Tridiagonal then becomes a special case of type b. - general banded matrix. If it happens to be diagonally dominant, pivoting can be shown to be unnecessary, and the very fast *Thomas algorithm* can be used. This is equivalent to unpivoted Gaussian elimination, but is usually described differently: We aim for a standard $A = L\,U$ - factorization, which will be of the form

$$\begin{bmatrix} a_{11} & a_{12} & & & \\ a_{21} & a_{22} & a_{23} & & \\ & a_{32} & a_{33} & a_{34} & \\ & & \ddots & \ddots & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ & l_{32} & l_{33} & & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} 1 & u_{12} & & & \\ & 1 & u_{23} & & \\ & & 1 & u_{34} & \\ & & & \ddots 1 & \ddots \\ & & & & \ddots \end{bmatrix}$$

For this $L\,U$ - product to become equal to A, we need

row 1:		$a_{11} = l_{11}$,	$a_{12} = l_{11}u_{12}$,
row 2:	$a_{21} = l_{21}$,	$a_{22} = l_{21}u_{12}+l_{22}$,	$a_{23} = l_{22}u_{23}$,
row 3:	$a_{32} = l_{32}$,	$a_{33} = l_{32}u_{23}+l_{33}$,	$a_{34} = l_{33}u_{34}$,
...	...	...	...

etc.

We could simply have copied the sub-diagonal of A over into the subdiagonal of L. The remaining equations give us recursively l_{11}, u_{12}, l_{22}, u_{23}, l_{33}, u_{34} etc. The two back substitutions becomes equally quick recursions.

In a computer program, one saves a lot of storage by just storing the diagonal elements in a few vectors - there obviously is no need to store all the zero entries.

b. In the banded case, we basically have to use the same strategy as for full matrices. The only difference is that L and U (even when partial pivoting is employed) will have non-zero entries only in some central diagonals. So again, it is much better to only store only these diagonals (in the columns of a matrix), and confine all the arithmetic to within it. The following schematic illustrations show how the reduction and fill-in would proceed in a case where the band is equally wide on both sides of the main diagonal, and partial (row-) pivoting is used. Removed entries are marked as "0" and locations where new entries are introduced by "$\times$":

$$
\begin{bmatrix}
\bullet & \bullet & \bullet & & & & & \\
\bullet & \bullet & \bullet & \bullet & & & & \\
\bullet & \bullet & \bullet & \bullet & \bullet & & & \\
 & \bullet & \bullet & \bullet & \bullet & \bullet & & \\
 & & \bullet & \bullet & \bullet & \bullet & \bullet & \\
 & & & \bullet & \bullet & \bullet & \bullet & \bullet \\
 & & & & \bullet & \bullet & \bullet & \bullet \\
 & & & & & \bullet & \bullet & \bullet
\end{bmatrix}
\sim
\begin{bmatrix}
\bullet & \bullet & \bullet & \times & \times & & & \\
0 & \bullet & \bullet & \bullet & \times & & & \\
0 & \bullet & \bullet & \bullet & \bullet & & & \\
 & \bullet & \bullet & \bullet & \bullet & \bullet & & \\
 & & \bullet & \bullet & \bullet & \bullet & \bullet & \\
 & & & \bullet & \bullet & \bullet & \bullet & \bullet \\
 & & & & \bullet & \bullet & \bullet & \bullet \\
 & & & & & \bullet & \bullet & \bullet
\end{bmatrix}
\sim
\begin{bmatrix}
\bullet & \bullet & \bullet & \times & \times & & & \\
0 & \bullet & \bullet & \bullet & \times & \times & & \\
0 & 0 & \bullet & \bullet & \bullet & \times & & \\
 & 0 & \bullet & \bullet & \bullet & \bullet & & \\
 & & \bullet & \bullet & \bullet & \bullet & \bullet & \\
 & & & \bullet & \bullet & \bullet & \bullet & \bullet \\
 & & & & \bullet & \bullet & \bullet & \bullet \\
 & & & & & \bullet & \bullet & \bullet
\end{bmatrix}
\sim \dots \sim
\begin{bmatrix}
\bullet & \bullet & \bullet & \times & \times & & & \\
0 & \bullet & \bullet & \bullet & \times & \times & & \\
0 & 0 & \bullet & \bullet & \bullet & \times & \times & \\
 & 0 & 0 & \bullet & \bullet & \bullet & \times & \times \\
 & & 0 & 0 & \bullet & \bullet & \bullet & \times \\
 & & & 0 & 0 & \bullet & \bullet & \bullet \\
 & & & & 0 & 0 & \bullet & \bullet \\
 & & & & & 0 & 0 & \bullet
\end{bmatrix}
$$

The U-part will not only fit back into the matrix that used to hold the diagonals - it can directly over-write in the space that the diagonals of A used to occupy. Had the RHS been available initially - and been manipulated along with the matrix - we would now be ready for back substitution. No additional storage would be needed.

If the RHS was not available at the time of the LU factorization, we also need to store one integer vector of pivot information - telling at each stage which row is moved into pivot position. Furthermore, we need to store the multiples used in the elimination. Using the same diagonal-based storage as above, the matrix structures would be

Input: Output:

$$
A = \begin{bmatrix}
 & & \bullet & \bullet & \bullet \\
 & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \\
\bullet & \bullet & \bullet & &
\end{bmatrix}, \qquad
L = \begin{bmatrix}
 & & \bullet \\
 & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\bullet & \bullet &
\end{bmatrix}, \quad
U = \begin{bmatrix}
1 & \bullet & \bullet & \bullet & \bullet \\
1 & \bullet & \bullet & \bullet & \bullet \\
1 & \bullet & \bullet & \bullet & \bullet \\
1 & \bullet & \bullet & \bullet & \bullet \\
1 & \bullet & \bullet & \bullet & \bullet \\
1 & \bullet & \bullet & \bullet & \bullet \\
1 & \bullet & \bullet & \bullet & \\
1 & \bullet & \bullet & &
\end{bmatrix}, \quad
P = \begin{bmatrix}
\bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\ \bullet \\
\end{bmatrix}
$$

With the information in L, U and P, we can rapidly - and repeatedly - solve the linear system $A\,\underline{x} = \underline{b}$ whenever a RHS vector $\underline{b}$ becomes available.

c. If one happens to have available solvers for general full and banded linear systems, one can solve this case of banded with extra rows/columns quickly. Schematically, we write the system to solve as (assuming for simplicity we have the RHS available from the start)

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \left[\begin{array}{c} \underline{x}_1 \\ \hline \underline{x}_2 \end{array}\right] = \left[\begin{array}{c} \underline{b}_1 \\ \hline \underline{b}_2 \end{array}\right] .$$

First, we solve (by a band solver) the many-RHS-system

$$\left[\begin{array}{c} A \end{array}\right] \left[\begin{array}{c|c} E & \underline{b}_3 \end{array}\right] = \left[\begin{array}{c|c} B & \underline{b}_1 \end{array}\right] .$$

The original system is now equivalent to

$$\left[\begin{array}{c|c} I & E \\ \hline C & D \end{array}\right] \left[\begin{array}{c} \underline{x}_1 \\ \hline \underline{x}_2 \end{array}\right] = \left[\begin{array}{c} \underline{b}_3 \\ \hline \underline{b}_2 \end{array}\right] ,$$

where I is the *identity matrix*, "1"s in the diagonal, and "0"s for the rest. This hold because we could have achieved the same thing (more costly) by multiplying the original system from the left by

$$\left[\begin{array}{c|c} A^{-1} & 0 \\ \hline 0 & I \end{array}\right] .$$

Now, we can easily - explicitly - add multiples of the top rows to clear out C, in the process turning $\underline{b}_2$ into $\underline{b}_4$:

$$\left[\begin{array}{c|c} I & E \\ \hline 0 & F \end{array}\right]\left[\begin{array}{c} \underline{x}_1 \\ \hline \underline{x}_2 \end{array}\right] = \left[\begin{array}{c} \underline{b}_3 \\ \hline \underline{b}_4 \end{array}\right] \quad .$$

Next we solve the square system (with one RHS)

$$\left[\, F \,\right]\left[\underline{x}_2\right] = \left[\underline{b}_4\right] \quad .$$

which has separated out at the bottom, independently of the rest of the equations. Knowing $\underline{x}_2$, we find the remaining unknowns $\underline{x}_1$ through

$$\left[\underline{x}_1\right] = \left[\underline{b}_3\right] - \left[\, E \,\right]\left[\underline{x}_2\right] \quad .$$