The following algorithms use naive Gaussian elimination followed by back substitution to compute the solution of $Ax = b$, where $A$ is an $n \times n$ matrix with $ij$th entry $a_{ij}$ and $b$ is an $n$-vector with $i$th component $b_i$.

These are structured as most modern software library subroutines are. The first algorithm performs naive Gaussian elimination on $A$, overwriting each $a_{ik}$ in the lower triangular part of $A$ with the "multiplier" $-a_{ik}/a_{kk}$. In library software, this algorithm is usually coded in a subroutine separate from the other two; in an application, it usually requires most of the computation. The second and third algorithms are usually combined in one subroutine that uses the output of the first subroutine. The second algorithm performs the same row operations on $b$ that were performed on $A$. The third algorithm performs back substitution, overwriting $b$ with the solution. Note that these algorithms require no more storage than is required for $A$ and $b$.

**Naive Gaussian Elimination:**

For $k = 1, \ldots, n - 1$

For $i = k + 1, \ldots, n$

$a_{ik} \leftarrow -a_{ik}/a_{kk}$

For $j = k + 1, \ldots, n$

$a_{ij} \leftarrow a_{ij} + a_{ik}a_{kj}$

**Row Operations on $b$:**

For $k = 1, \ldots, n - 1$

For $i = k + 1, \ldots, n$

$b_i \leftarrow b_i + a_{ik}b_k$

**Back Substitution:**

$b_n \leftarrow b_n/a_{nn}$

For $i = n - 1, \ldots, 1$

$b_i \leftarrow \left(b_i - \sum_{j=i+1}^{n} a_{ij}b_j\right)/a_{ii}$