to simplify the problem

We first add structure to the pattern of the elements in the matrix

How do we compute this in practice?

Independent.

A must be nonsingular, i.e., its columns must be linearly

where $A$ is an $n \times n$ matrix and $x$ and $q$ are $n$ vectors.

$q = Ax$

defined by the matrix identity

We have given general conditions for a system of linear equations

Solving Linear Systems
$u, \ldots, k = 1$ for $j \neq 1$ and $a_j \neq a_j$ for $j \neq 1$.

Next simplest case $A$ is a nonsingular diagonal matrix with $\alpha_j = 0$ for $j \neq 1$.

\[
q = x
\]
\[
q = xI
\]
\[
q = xA
\]

$$[u_1, \ldots, u_n] = I = A$$

Simplest case: $n = \ell \neq 1$.
Linear independence implies \( u', \ldots, u'_m \neq 0 \).

\[
\begin{aligned}
\forall \phi &= \forall \alpha \\
\exists \phi &= \exists \alpha \\
\exists \phi &= \exists \alpha \\
\forall \phi &= \forall \alpha \\
\text{This defines the following identities:} \\
\begin{pmatrix}
\forall \phi \\
\exists \phi \\
\exists \phi \\
\forall \phi
\end{pmatrix}
&=
\begin{pmatrix}
\forall \alpha \\
\exists \alpha \\
\exists \alpha \\
\forall \alpha
\end{pmatrix}
\begin{pmatrix}
\alpha \\
\alpha \\
\alpha \\
\alpha
\end{pmatrix}
\end{aligned}
\]

Example: \( \forall u = u \).
Triangular Systems

Consider the following set of equations with $n = 4$

\[
\begin{align*}
\lambda_{11}\xi_1 &= \phi_1 \\
\lambda_{21}\xi_1 + \lambda_{22}\xi_2 &= \phi_2 \\
\lambda_{31}\xi_1 + \lambda_{32}\xi_2 + \lambda_{33}\xi_3 &= \phi_3 \\
\lambda_{41}\xi_1 + \lambda_{42}\xi_2 + \lambda_{43}\xi_3 + \lambda_{44}\xi_4 &= \phi_4 
\end{align*}
\]

This can be written using matrix notation as the lower triangular system

\[Lx = f\]

where $L = [\lambda_{ij}] \in \mathbb{R}^{n \times n}$, $\lambda_{ij} = 0$ if $i < j$, $f = [\phi_i] \in \mathbb{R}^n$. 
\[
\begin{pmatrix}
\forall \phi \\
\exists \phi \\
\exists \phi \\
\exists \phi
\end{pmatrix} = \begin{pmatrix}
\forall \xi \\
\exists \xi \\
\exists \xi \\
\exists \xi
\end{pmatrix} \begin{pmatrix}
\forall \nu \gamma, \exists \nu \gamma, \exists \nu \gamma, \exists \nu \gamma \\
\exists \nu \gamma, \exists \nu \gamma, \exists \nu \gamma, \exists \nu \gamma \\
\exists \nu \gamma, \exists \nu \gamma, \exists \nu \gamma, \exists \nu \gamma \\
\exists \nu \gamma, \exists \nu \gamma, \exists \nu \gamma, \exists \nu \gamma
\end{pmatrix}
\]

'\forall, \exists, \cdots, \exists' = \exists \not\exists 0 \neq \exists \gamma \in \left[\gamma\right] = x
Rewriting the equations gives the following identities:
\[ \begin{align*}
\phi &= (I)A \\
\zeta &= (I)X \\
\chi &= (f', I)T \\
\end{align*} \]

First we must choose data structures to store the mathematical objects \( f \) and \( x \) that are one is oriented towards rows and the other columns.

As a result we can derive two standard sequential methods for solving these systems in operations. They differ in the fact that one is oriented towards rows and the other columns.
\[(I'I)T / (I)\mathcal{F} = (I)X\]

(Endo)

\[(f)X (f'I)T - (I)\mathcal{F} = (I)\mathcal{F}\]

\[
I - I'I = f \circ p
\]

\[N \, \exists = I \circ p\]

\[(I'I)T / (I)\mathcal{F} = (I)X\]

Rounded
\[(N'N)^T / (N)X = (N) \Phi \]
\[\text{end do}\]
\[\text{end do}\]
\[(\Phi)X (\Phi'\Phi)^{-T} - (\Phi) = (\Phi) \Phi\]
\[N' + \Phi = I \text{ op}\]
\[N' - \Phi = I \text{ op}\]
\[\text{Column oriented} : \Phi \]
\[
\begin{pmatrix}
I^{-u_f} \\
I_{\phi}
\end{pmatrix} = 
\begin{pmatrix}
I^{-u_x} \\
I_{\xi}
\end{pmatrix} 
\begin{pmatrix}
I^{-u_T} & I^{-u_l} \\
L_0 & I_{11}
\end{pmatrix}
\]

Let \( I \) be an \( n \times n \) lower triangular matrix. Let \( I^{-u} \) be another point of view matrix. We can derive these algorithms from another point of view.
\[ 1^{-u} f = 1^{-u} \mathcal{I}_\Sigma - 1^{-u} f = 1^{-u} x 1^{-u} \mathcal{I} \]

\[ \Pi \chi / I \phi = \mathcal{I}_\Sigma \]

\[ 1^{-u} f = 1^{-u} x 1^{-u} \mathcal{I} + 1^{-u} \mathcal{I}_\Sigma \]

\[ I \phi = \mathcal{I}_\Sigma \Pi \chi \]

**NOTE:** This is the column-oriented algorithm.

- Dimension \( u - 1 \). The problem with the same structure as the \( u \times u \) problem produces a problem and produces a solution for the \( I \times I \) problem. Therefore, we can solve the \( I \times I \) problem and then produce a solution for the \( u \times u \) problem.
\[
\begin{pmatrix}
  u_\phi \\
  1 - u_f
\end{pmatrix} =
\begin{pmatrix}
  u_\Sigma \\
  1 - u_x
\end{pmatrix}
\begin{pmatrix}
  uu_{\chi} & 1 - u_l \\
  0 & 1 - u_T
\end{pmatrix}
\]

Alternatively, partition to solve a problem of order \( n - 1 \) and expand.
\[ uu \chi / (1 - u x^{1-u} I - u \phi) = u \theta \]

Problem of order \( u \) via \( \phi \)

Assuming the solution \( x^{-u} \) is available, we get the solution to the

\[
\begin{align*}
    u \phi &= u \theta uu \chi + 1 - u x^{1-u} I \\
    1 - u f &= 1 - u x^{1-u} I
\end{align*}
\]

These identities follow:
row-oriented algorithm.

Increasing, the resulting algorithms is equivalent to the $u = 1$ and incrementally produce the solutions for

If the identities are applied starting from a known solution for the system

This immediately generates a recursive algorithm to solve the

NOTE:
used to implement high-performance linear algebra libraries.

We next consider a factorization-based approach that yields a
matrix partitioning •
loop-based reasoning •
algorithms for triangular systems:
We have seen two ways of deriving the standard sequential
NOTE: A row form of $L'\mathbf{e}^i + I$ can also be defined.

At the $i$-th position, the vector which has 0 in all positions with the exception of a 1 in the $i$-th position, where $\mathbf{e}^i$ is 0 in the first $i$ positions and $\mathbf{e}^i$ is the standard basis.

$\begin{bmatrix} \mathbf{L}' \mathbf{e}^i \end{bmatrix} + I = \mathbf{W}$

is a matrix of the form:

**DEFINITION:** An elementary triangular matrix (column form)

**Elementary triangular matrices and solving triangular systems**
Example: $n = 4, i = 2,$
Adding 4 places 1 in the diagonal positions:

\[
\begin{pmatrix}
\varphi \\
\nu \\
0 \\
0
\end{pmatrix}
= 
\begin{bmatrix}
0 & 0 & \varphi & 0 \\
0 & \nu & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]
positions in the product, i.e., no operations need to be performed.

Elements below the diagonal of \( M \) and \( \hat{M} \) into the corresponding

If then \( a = 0 \) and the product merely copies the nonzero

\[ \ell \in \mathcal{E} \cap \mathcal{F} \]

where \( a = 0 \). Then \( \ell \in \mathcal{E} \cap \mathcal{F} \)

\[ \ell \in \mathcal{E} \cap \mathcal{F} \]

\[ \ell \in \mathcal{E} \cap \mathcal{F} \]

\[ \ell \in \mathcal{E} \cap \mathcal{F} \]

Consider the following:

\[ \ell \in \mathcal{E} \cap \mathcal{F} \]
\[
\begin{pmatrix}
1 & 0 & \varepsilon_0 \\
1 & \varepsilon_0 & 0 \\
1 & \varepsilon_0 & 0 \\
1 & 0 & 0
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & \varepsilon_0 \\
1 & \varepsilon_0 & 0 \\
1 & \varepsilon_0 & 0 \\
1 & 0 & 0
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & \varepsilon_0 \\
1 & \varepsilon_0 & 0 \\
1 & \varepsilon_0 & 0 \\
1 & 0 & 0
\end{pmatrix}
\]

\[\vartheta = u\]
The computations required are a single vector triad. The nonzero patterns below the diagonal in columns \( i \) and \( j \) only. Therefore, the nonzero pattern of the product is the union of the

\[
\ell \in (\ell \in \alpha + \ell \in \gamma) + \ell \in \gamma + I = \ell \in \mathcal{W} \cap \mathcal{W}
\]

If \( 0 \neq \ell \in \mathcal{L} \cap \mathcal{E} \Rightarrow \ell \in \alpha \).
For example, if $W'$ and $W$ have $\not\exists$ non-trivial columns each then

continuous columns which are nonzero below the diagonal.

extend immediately to the case where $W$ and $W'$ have a set of

These facts about multiplication of elementary triangular matrices.
$\mathbf{T}$ of form defined by the nonzeros below the diagonal in the $i$-th column.

where $\mathbf{C}$ is the elementary triangular matrix in column $i$.

$$\prod_{i=1}^{\eta} \mathbf{C}_{i} = \mathbf{T}$$

**Corollary:** A unit lower triangular matrix $\mathbf{T}$ can be written as previous facts:

The following results follow in a straightforward fashion from the
Due to the nonzero structure in $l^i$ the trial is of length $n - l^i$.

**NOTE:**

\[
\begin{align*}
\theta^i \gamma + x &= \\
x(\theta^i \gamma l + I) &= \theta^i x
\end{align*}
\]

Corollary: The multiplication of an elementary triangular matrix in column form by a vector is equivalent to a trial.
NOTE: Due to the nonzero structure in $\ell^i$ the dot product is of length $n - i$.

Only one element of $x$ is updated: $x^i$

\[
x \ell^i < \sigma + x =
\]

\[
x(\ell^i < \sigma + I) = x^i H
\]

COROLLARY: The multiplication of an elementary triangular matrix in row form by a vector is equivalent to a dot product.
Inversion of elementary triangular matrices.

Inverses of $R$ and $L$ are elementary triangular matrices, equivalent to a Rijndael (or a dot product for the row version) and the

$q = x^T \mathbf{C} \mathbf{R}$. Therefore, solving a system is

trivial, $\mathbf{C}^{-1} L - I = \mathbf{C}^{-1}$.

**Corollary:** Inversion of elementary triangular matrices is
The inverse of a (unit) lower triangular matrix is also a (unit) lower triangular matrix.
Similarly construct the row \( \ell \) of \( T \). 

To column \( \ell \) in \( T \) with the \( I \) on the diagonal removed and \( \ell \) is in the vector corresponding

where \( \ell \) is the vector corresponding

\[
\prod_{u}^{\ell} \prod_{1}^{\ell} = T
\]

Let

a straightforward manner based on this algebraic characterization.

The column and row algorithms to solve can be derived in \( f = xT \) approach to the column and row algorithms

An algebraic approach to the column and row
unless more standard factorizations are involved in defining this factorization.

\[ \ell \mathcal{H} = \ell \mathcal{W} \quad \text{and} \quad \ell \mathcal{O} = \ell \mathcal{N} \]

where

\[ f(\ell \mathcal{W} \prod_{z}^{u=\ell}) = f(\ell \mathcal{N} \prod_{1-u=\ell}^{1}) = x \]

Clearly
since $N$ is a tripod and $\mathcal{W}$ is a dot product. This can in turn be used to determine the computational primitives

$\cdot (((((((f^2\mathcal{W})^3\mathcal{W})^4\mathcal{W})^5\mathcal{W})^7\mathcal{W})^8\mathcal{W})$ 

and

$((((((f^1N)^3N)^4N)^5N)^6N)^7N)^8N)$

of the computations. For $u = 8$ the group is $\mathcal{G}$.

The column and row sweeps are generated by a particular grouping