Sparse Matrices in Power Flow Calculations
Any matrix that has a high proportion of its elements equal to zero is a **sparse matrix**.

- **Most large matrices that arise in engineering applications are sparse.**
The Sparsity of a Matrix - A definition

A sparse matrix: A matrix, where most of the elements are zero, but a few are different from zero.

A full matrix: A matrix, where all the elements are non-zero.

The sparsity of a matrix = \frac{\text{Number of elements}}{\text{Total number of elements}} = 0 \cdot 100[\%]
Why and What Are Sparse Matrices in Power Systems?

• Matrices in power systems are often very large and very sparse!
• A large matrix: is for instance of dimension 100x100, 1000x1000 or 10000x10000!
• A sparse matrix: A matrix where, for instance, 99% of the elements are zero, but 1% are different from zero.
• Examples of these matrices are the $Y_{bus}$ matrix and the Jacobian matrix, which both are used in power flow calculations.
### Y_{bus} Matrix Composition

We review the rule for building the $Y_{bus}$ matrix:

$$
Y_{bus} = \begin{bmatrix}
    y_{11} & y_{12} & \cdots & y_{1n} \\
    y_{21} & y_{22} & \cdots & y_{1n} \\
    \vdots & \vdots & \ddots & \vdots \\
    y_{n1} & y_{n2} & \cdots & y_{nn}
\end{bmatrix}
$$

- **The diagonal elements of the $Y_{bus}$ matrix** consist of the sum of the (series and shunt) admittances connected to the bus in question.
- **These elements are always present!**
- **The off-diagonal elements of the $Y_{bus}$ matrix** consist of the negative value of the series admittances connecting the 2 buses in question. These elements will be zero when no direct connection is between buses. (The $Y_{bus}$ matrix is very sparse!)
- **These elements are present only when there is a link (a line or transformer)**

We review the rule for building the $Y_{bus}$ matrix:

$$y_{ik} = |y_{ik}| e^{\gamma_{ik}}$$
The Sparsity of a Matrix: Example 1:

An example: Assume, in an electrical power system, each bus is on the average connected to 1.5 other buses. Further assume that we have a 100 bus system. We get a 100 by 100 matrix with 100 \cdot 100 elements.

With the above assumption in the $Y_{\text{bus}}$ matrix there are 100 diagonal elements $\neq 0$ and 150 elements above the diagonal and 150 elements below the diagonal $\neq 0$. The sparsity of this matrix will therefore be:

$$\text{Sparsity} = \frac{100 \cdot 100 - 150 - 150 - 100}{100 \cdot 100}$$

$$\text{Sparsity} = 96.0\%$$

We can say that the matrix is 4% full.
Similarly for a 1000 bus system we get a matrix with 1000 \cdot 1000 elements. We have for the same assumption of how each bus is interconnected for the Y_{bus} matrix 1000 diagonal elements \neq 0 and 1500 elements above the diagonal and 1500 elements below the diagonal \neq 0. The sparsity of this matrix will be:

\[
\text{Sparsity} = \frac{1000 \cdot 1000 - 1500 - 1500 - 1000}{1000 \cdot 1000}
\]

\[
\text{Sparsity} = \frac{996}{1000} = 0.996 \%
\]

We can perhaps say that the matrix is 0.4% full.
The Sparsity of the Jacobi-Matrix

The $J_1$ sub-matrix:

$$\frac{\partial P_i}{\partial \delta_i} = - \sum_{k=1, k\neq i}^{n} |V_i||V_k||y_{ik}| \cdot \sin(\delta_i - \delta_k - \gamma_{ik})$$

$$\frac{\partial P_i}{\partial \delta_k} = |V_i||V_k||y_{ik}| \cdot \sin(\delta_i - \delta_k - \gamma_{ik}) \quad i \neq k$$

The $J_2$ sub-matrix:

$$\frac{\partial P_i}{\partial V_i} = 2|V_i||y_{ii}| \cdot \cos(-\gamma_{ii}) + \sum_{j=1, j\neq i}^{n} |V_j||y_{ij}| \cdot \cos(\delta_i - \delta_j - \gamma_{ij})$$

$$\frac{\partial P_i}{\partial V_k} = |V_i||y_{ik}| \cdot \cos(\delta_i - \delta_k - \gamma_{ik}) \quad k \neq i$$

The $J_3$ sub-matrix:

$$\frac{\partial Q_i}{\partial \delta_i} = \sum_{j=1, j\neq i}^{n} |V_i||V_j||y_{ij}| \cdot \cos(\delta_i - \delta_j - \gamma_{ij})$$

$$\frac{\partial Q_i}{\partial \delta_k} = -|V_i||V_k||y_{ik}| \cdot \cos(\delta_i - \delta_k - \gamma_{ik}) \quad k \neq i$$

The $J_4$ sub-matrix:

$$\frac{\partial Q_i}{\partial V_i} = 2|V_i||y_{ii}| \cdot \sin(-\gamma_{ii}) + \sum_{j=1, j\neq i}^{n} |V_j||y_{ij}| \cdot \sin(\delta_i - \delta_j - \gamma_{ij})$$

$$\frac{\partial Q_i}{\partial V_k} = |V_i||y_{ik}| \cdot \sin(\delta_i - \delta_k - \gamma_{ik}) \quad k \neq i$$

The elements of $Y_{bus}$ are also in the Jacobi matrices.

**Conclusion:** If $Y_{bus}$ is a sparse matrix, the Jacobi matrix is also sparse.
A Random Sparse Matrix

Sparse matrices can be of different nature
IEEE TEST systems

- IEEE test systems are standardized power systems to test research theories in power system analysis.
- We will look at different sparse $Y_{bus}$ matrices for different IEEE test systems along with their one-line diagram:
  - 14 bus
  - 30 bus
  - 57 bus
  - 118 bus
  - 300 bus
- These test systems can be found at:
  - http://www.ee.washington.edu/research/pstca/
  - (http://www.pserc.cornell.edu/matpower/)
IEEE 14 BUS TEST CASE

Use the SPY function in Matlab
IEEE 30 BUS TEST CASE

Use the “SPY” function in Matlab to obtain the graphical representation of the sparse matrix
IEEE 57 BUS TEST CASE

Use the SPY function in Matlab
Sparsity 93%
IEEE 118 bus test system
IEEE 300 bus test system

See details for the one-line diagrams for the 300 bus system on the following slides
300 bus IEEE test system (1) one line diagram
300 bus IEEE test system (2) one line diagram
300 bus IEEE test system (3) one line diagram
Sparsity structure of a 108 bus Icelandic system $Y_{bus}$ matrix
Solutions to the power system problem with sparse matrices

• For large systems (and power systems are large) we need to exploit the sparsity of the system matrices.

• Therefore solutions to power flow problem with Newton’s method involves solution to a set of linear equations rather than direct inversion of matrices.

• We use Gauss elimination as a standard set of solving large linear systems of equations where we exploit simultaneously the sparsity structure of the system matrix (Jacobi).
Sparse Matrices and Solutions to Sets of Linear Equations in Power Systems

\[
\begin{bmatrix}
    a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
    a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{n,1} & a_{n,2} & \cdots & a_{n,n}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_n
\end{bmatrix}
= 
\begin{bmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_n
\end{bmatrix}
\]

To solve: Find a vector of unknown x-es. Other quantities are known or the a-s and b-s are known.
Matrix Notation in Linear Equations

\[ A \cdot x = b \]

\[ A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \]

\( A \) is a known matrix
\( x \) is a vector with unknown variables. \( b \) is a known vector.
Solution methods for linear equations based on numerical analysis

Direct methods

**Gaussian elimination**
- Gauss-Jordan elimination
- LU factorization
- Cholesky factorization

Iterative methods

- Jacobi iterations
- Gauss-Seidel iteration
Solutions to a Set of Linear Equations

• If we can transform the matrix to a triangular form, it is possible to solve the equations
• “Elementary Operations” can be carried out without changing the solution
• Examples of “Elementary Operation”:
  – Multiply rows (equations) with a constant
  – Add rows (equations) to each other
• This is the Gauss elimination method
Assume for a moment that a matrix has the special triangular form case -- The matrix, \( A \) is triangular with all zeros below the diagonal:

\[
\begin{bmatrix}
  a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
  0 & a_{2,2} & \cdots & a_{2,n} \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & a_{n,n}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
\end{bmatrix} =
\begin{bmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_n
\end{bmatrix}
\]

Solution method: Back substitution
Back Substitution

\[
\begin{bmatrix}
  a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
  0 & a_{2,2} & \cdots & a_{2,n} \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & a_{n,n}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
\end{bmatrix}
= 
\begin{bmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_n
\end{bmatrix}
\]

\[
a_{n,n}x_n = b_n
\]
\[
x_n = \frac{b_n}{a_{n,n}}
\]

\[
a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1}
\]
\[
x_{n-1} = \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}}
\]
\[
a_{n-2,n-2}x_{n-2} + a_{n-2,n-1}x_{n-1} + a_{n-2,n}x_n = b_{n-2}
\]
\[
x_{n-2} = \frac{b_{n-2} - a_{n-2,n}x_n - a_{n-2,n-1}x_{n-1}}{a_{n-2,n-2}}
\]
\[
\vdots
\]
\[
\]

The last line
\[
b_k - \sum_{j=k+1}^{n} a_{k,j}x_j
\]
\[
x_k = \frac{b_k}{a_{k,k}}
\]
\[
\vdots
\]
\[
x_1 = \frac{b_1 - a_{1,2}x_2 - a_{1,3}x_3 - \cdots - a_{1,n}x_n}{a_{1,1}}
\]

The 3rd line from the end
Solution method

• How do we get any matrix to a triangular form?
• Transform the matrix by Gauss elimination to a lower triangular form matrix
• Solve such a matrix by back substitution

• We have already checked out back substitution. Let us look at Gauss elimination!!
Gaussian Elimination (1st step)

The original matrix:

\[
\begin{bmatrix}
  a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\
  a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n} \\
  a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,n} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  a_{n,1} & a_{n,2} & a_{n,3} & \cdots & a_{n,n}
\end{bmatrix}
\]

Multiply the 1st row with \( \frac{a_{2,1}}{a_{1,1}} \) and subtract from the 2nd row.

Multiply the 1st row with \( \frac{a_{3,1}}{a_{1,1}} \) and subtract from the 3rd row.

Multiply the 1st row with \( \frac{a_{n,1}}{a_{1,1}} \) and subtract from the last row.
Gaussian Elimination (2\textsuperscript{nd} step)

The matrix after the 1\textsuperscript{st} step:

The 1st column is with zeros except at the top row

\[
\begin{pmatrix}
\alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \cdots & \alpha_{1,n} \\
0 & \alpha_{2,2} & \alpha_{2,3} & \cdots & \alpha_{2,n} \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & \alpha_{n-1,2} & \alpha_{n-1,3} & \cdots & \alpha_{n-1,n} \\
0 & \alpha_{n,2} & \alpha_{n,3} & \cdots & \alpha_{n,n}
\end{pmatrix}
\]
Gaussian Elimination (2\textsuperscript{nd} step)

The matrix after the 1\textsuperscript{st} step:

Now the 2\textsuperscript{nd} row plays the same role as the 1\textsuperscript{st} row before:

Multiply the 2\textsuperscript{nd} row with \( \frac{a^{(1)}_{3,2}}{a^{(1)}_{2,2}} \)

and subtract from the 3\textsuperscript{rd} row

Multiply the 2\textsuperscript{nd} row with \( \frac{a^{(1)}_{n,2}}{a^{(1)}_{2,2}} \)

and subtract from the last row
Gaussian Elimination (3\textsuperscript{rd} step)

The matrix after the 2\textsuperscript{nd} step:

\[
\begin{pmatrix}
0 & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\
0 & a_{2,2}^{(1)} & a_{2,3}^{(1)} & \cdots & a_{2,n}^{(1)} \\
\vdots & \vdots & \ddots & \cdots & \vdots \\
0 & 0 & a_{n-1,3}^{(2)} & \cdots & a_{n-1,n}^{(2)} \\
0 & 0 & a_{n,3}^{(2)} & \cdots & a_{n,n}^{(2)}
\end{pmatrix}
\]

- The 1\textsuperscript{st} column with zeros except at the top
- The 2\textsuperscript{nd} column with zeros, except in the 2 top rows
Gauss-Elimination Transforms the Matrix to the Following Form:

\[ U \cdot x = G \]

\[
\begin{pmatrix}
1 & a'_{12} & \cdots & a'_{1n} \\
0 & 1 & \cdots & a'_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{pmatrix}
=
\begin{pmatrix}
b'_1 \\
b'_2 \\
\vdots \\
b'_n
\end{pmatrix}
\]

This matrix is \( U \) or an “upper” or “unit upper” triangular matrix. The elements are calculated with iteration, where each step is an “elementary operation”. It is possible to solve the equation with a simple “back substitution”. It is found by dividing each row with the diagonal element.
Gaussian Elimination

• After \(n-1\) steps the matrix has become triangular!

• Computer time grows fast with the matrix size (increasing \(n\))

• Computer time grows in proportion to \(n^3\), if the matrix is not sparse

• We can obtain a “1” on the diagonal by dividing by the diagonal element in each step