MA3227 Numerical Analysis II

Lecture 4: Sparse LU Factorisation

Simon Etter

Semester II, AY 2020/2021

Sparse LU Factorisation

Problem statement

Solve $Ax = b$ via LU factorisation in less than $O(n^3)$ operations, assuming $A \in \mathbb{R}^{n \times n}$ is sufficiently sparse.

Sparse LU Factorisation

Let us begin our discussion by recapitulating the main ideas of the standard, dense algorithms.

Thm: LU factorisation

Any matrix $A \in \mathbb{R}^{n \times n}$ can be written as $PA = LU$, where

- $P \in \mathbb{R}^{n \times n}$ is a permutation matrix,
- $L \in \mathbb{R}^{n \times n}$ is lower-triangular with unit diagonal, and
- $U \in \mathbb{R}^{n \times n}$ is upper-triangular.

The matrices L and U are unique for fixed P if A is invertible.

Proof. See any linear algebra textbook.

I will discuss the definition and meaning of the permutation factor P on slide 11. For now, let us ignore this factor and instead address the following questions:

- How do we compute $A = LU$?
- How do we use the $A = LU$ to solve $Ax = b$.

Sparse LU Factorisation

Computing the LU factorisation

Main ideas:

- Use the top left entry to eliminate all entries below it.
  This top left entry is called pivot.
- Use the L-factor to keep track of the elimination factors.

Example.

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 1 & -2 \\ -8 & 2 & 3 \\ 12 & 7 & -5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 1 & -2 \\ 0 & 4 & -1 \\ 0 & 4 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 1 & -2 \\ 0 & 4 & -1 \\ 0 & 4 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 1 & 1 \end{pmatrix} \begin{pmatrix} 4 & 1 & -2 \\ 0 & 4 & -1 \\ 0 & 0 & 2 \end{pmatrix} = LU$$

Sparse LU Factorisation

Solving linear systems

The purpose of the LU factorisation is to reduce an arbitrary linear systems to two triangular system,

$$Ax = LUx = b \qquad \iff \qquad Ly = b, \quad Ux = y$$

Doing so is useful because triangular systems are easy to solve.

Example

$$\begin{pmatrix} 4 & 1 & -2 \\ & 2 & -1 \\ & & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 6 \end{pmatrix}$$

$$\begin{array}{lrcl}
\text{Third eq.:} & 3x_3 = 6 & \implies & x_3 = 2 \\
\text{Second eq.:} & 2x_2 - x_3 = 4 & \implies & x_2 = 3 \\
\text{First eq.:} & 4x_1 + x_2 - 2x_3 = 3 & \implies & x_1 = 1
\end{array}$$

Terminology: Back substitution

The above algorithm is known as *back substitution*.

Remark. Some authors distinguish between back substitution for solving upper-triangular systems and forward substitution for solving lower-triangular systems. I will call both of these algorithms back substitution because they are fundamentally the same algorithm.

Sparse LU Factorisation

The runtimes of sparse LU factorisation and back substitution are easily derived from the above algorithm sketches.

Thm: Runtime of sparse LU factorisation

The runtime of computing the LU factorisation $LU = A$ is

$$O\left(\sum_{k=1}^{n} \mathrm{nnz}\big(L[:,k]\big)\, \mathrm{nnz}\big(U[k,:]\big)\right).$$

: is a shorthand notation for $\{1, \dots, n\}$.

Thm: Runtime of sparse back substitution

The runtime of solving $Ly = b$ and $Ux = y$ via back substitution is

$$O\Big(\mathrm{nnz}(L) + \mathrm{nnz}(U)\Big).$$

Proofs of these results will be provided on the following slides.

Sparse LU Factorisation

Proof of LU runtime.

Let us denote by $U^{(k)}$ the state of the U-factor just before eliminating the lower-triangular entries in the kth column (see slide 4).

The LU factorisation can then be formulated as follows.

for $k = 1, \dots, n$ **do**
    for $i = k+1, \dots, n$ such that $U^{(k)}[i,k] \neq 0$ **do**
        $L[i,k] = U^{(k)}[i,k] / U[k,k]$
        for $j = k+1, \dots, n$ such that $U[k,j] \neq 0$ **do**
            $U^{(k+1)}[i,j] = U^{(k)}[i,j] - L[i,k]\, U[k,j]$
        end for
    end for
end for

This immediately yields a runtime of

$$O\left(\sum_{k=1}^{n} \mathrm{nnz}\big(L[:,k]\big)\, \mathrm{nnz}\big(U[k,:]\big)\right).$$

Sparse LU Factorisation

Proof of back substitution runtime.

Back substitution applied to $Ux = y$ proceeds as follows.

```
1: for i = 1, ..., n do
2:     x[i] = y[i]
3:     for j = i + 1, ..., n such that U[i, j] ≠ 0 do
4:         x[i] = x[i] − U[i, j] x[j]
5:     end for
6:     x[i] = x[i]/U[i, i]
7: end for
```

We observe that line 4 is executed exactly once for each nonzero entry of U, and the runtimes of all other lines are negligible in the big-O sense. The runtime of solving $Ux = y$ is thus $O\big(\mathrm{nnz}(U)\big)$, and likewise we conclude that the runtime of $Ly = b$ is $O\big(\mathrm{nnz}(L)\big)$.

Sparse LU Factorisation

Remarks on the LU and back substitution runtime estimates

The above runtime estimates deserve a few more remarks.

- If L and U are dense, then the runtime of the LU factorisation is

$$O\left(\sum_{k=1}^{n} \operatorname{nnz}\big(L[:,k]\big)\operatorname{nnz}\big(U[k,:]\big)\right) = O\left(\sum_{k=1}^{n} k^2\right) = O\big(n^3\big),$$

  and the runtime of back substitution is

$$O\big(\operatorname{nnz}(L) + \operatorname{nnz}(U)\big) = O\big(n^2\big).$$

  The above results thus include the well-known runtimes for dense factorisation and back substitution as a special case.

- The runtime of LU factorisation is at least as large as that of back substitution since $\operatorname{nnz}(L[:,k]), \operatorname{nnz}(U[k,:]) \geq 1$ and thus

$$\sum_{k=1}^{n} \operatorname{nnz}\big(L[:,k]\big)\operatorname{nnz}\big(U[k,:]\big) \geq \begin{cases} \sum_{k=1}^{n} \operatorname{nnz}\big(L[:,k]\big) = \operatorname{nnz}(L), \\ \sum_{k=1}^{n} \operatorname{nnz}\big(U[k,:]\big) = \operatorname{nnz}(U). \end{cases}$$

  This point explains why we will be talking about LU factorisation rather than back substitution for most of this lecture.

Sparse LU Factorisation

Why the LU theorem involves a permutation matrix

Let us now return to the question of why the LU theorem from slide 3 involves a permutation factor P.

It turns out that this factor is needed to swap rows in case the top-left entry is zero and hence cannot be used as a pivot element.

Example.

$$\begin{pmatrix} 0 & 3 \\ 1 & 2 \end{pmatrix} \quad \longrightarrow \quad \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix}.$$

In the language of linear algebra, "swapping rows" corresponds to replacing

$$Ax = b \qquad \text{with} \qquad (PA)\,x = Pb$$

where P is a permutation matrix defined as follows.

Sparse LU Factorisation

Def: Permutations and permutation matrices

A *permutation* is a bijective map $\pi : \{1, \dots, n\} \to \{1, \dots, n\}$.

The *permutation matrix* associated with a permutation π is a matrix $P \in \mathbb{R}^{n \times n}$ such that $(Pb)[i] = b[\pi(i)]$ for all $b \in \mathbb{R}^n$.

Example

i	1	2	3
$\pi(i)$	1	3	2

$$\longleftrightarrow \qquad P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Useful facts

- A matrix P is a permutation matrix if and only if all its entries are zero except for exactly one 1 in each row and column.
- We have $(PA)[i, j] = A[\pi(i), j]$; hence $A \mapsto PA$ indeed corresponds to swapping rows.
- The product of two permutation matrices is again a permutation matrix.

Sparse LU Factorisation

Why the LU theorem involves a permutation matrix (continued)

The example on slide 11 demonstrates that swapping rows is sometimes necessary to make the LU factorisation work. One can further show that it is also sufficient, i.e. the row-permuted LU factorisation $LU = PA$ exists for all A. This is one of the key points of the LU theorem on slide 3.

Nevertheless, I will usually omit the P-factor in the following, i.e. I will say "let $LU = A$ be the LU factorisation of A" rather than "let $LU = PA$ be the LU factorisation of A". The intention in doing so is that we can always replace A with its row-permuted copy PA and thereby avoid having to carry an extra factor P around.

Outlook

We are now ready to start looking into how we can exploit sparsity in the LU algorithm. I will begin this journey on the next slide with an example which illustrates some of the complications arising in sparse LU factorisations.

Sparse LU Factorisation

Example

$$\begin{pmatrix} 1 & & 1 & \\ & 1 & 1 & \\ & & 1 & \\ 1 & 1 & & 1 \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 1 & 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & & 1 & \\ & 1 & 1 & \\ & & 1 & \\ & & & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & & 1 & \\ & -1 & 1 & \\ & & 1 & \\ 1 & 1 & & 1 \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 1 & 1 & & 1 \end{pmatrix} \begin{pmatrix} 1 & & 1 & \\ & -1 & 1 & \\ & & 1 & \\ & & & 1 \end{pmatrix}.$$

Observations:

- $A[i,j] = 0$ does not imply $L[i,j] = 0$ or $U[i,j] = 0$.
- Some entries of L, U are zero regardless of the values we assign to the nonzero entries of A. Others may be zero or nonzero depending on the values in A.

Sparse LU Factorisation

The above example prompts us to introduce some terminology.

Terminology: Sparsity pattern

The set of all indices $(i, j) \in \{1, \ldots, n\}^2$ such that $A[i, j] \neq 0$ is called the *sparsity pattern* or *structure* of a matrix $A \in \mathbb{R}^{n \times n}$.

Sparsity patterns are conveniently described by drawing a matrix where each zero entry is left empty and each nonzero entry is marked with a bullet ($\bullet$), see the middle matrix below.

In all the examples considered in this lecture, the diagonal will always be nonzero. I use this fact to write the numbers 1 to n on the diagonal rather than bullets, see the right matrix below. Doing so makes the sparsity patterns easier to read and talk about.

Example

$$\begin{pmatrix} 3 & 0 & 2 \\ 5 & 1 & 0 \\ 0 & 0 & 4 \end{pmatrix} = \left(\begin{array}{c|c|c} \bullet & & \bullet \\ \hline \bullet & \bullet & \\ \hline & & \bullet \end{array}\right) = \left(\begin{array}{c|c|c} 1 & & \bullet \\ \hline \bullet & 2 & \\ \hline & & 3 \end{array}\right).$$

Sparse LU Factorisation

Terminology: Structural nonzero

Let B be a matrix derived from some other matrix A (e.g. $B = A^2$, $B = A^{-1}$, or B is one of the LU factors).

An entry $B[i,j]$ is called *structurally nonzero* if $B[i,j]$ could be nonzero given the sparsity pattern of A, and *structurally zero* otherwise.

I will occasionally use $B[i,j] \neq 0$ as a shorthand notation for saying that $B[i,j]$ is structurally nonzero (but not necessarily actually nonzero).

Example

Consider

$$A = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}, \qquad A^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

- $A^2[2,1] = 0 \times 1 + -1 \times 0$ is structurally zero because the only way to make this number nonzero is to assign a nonzero value to $A[2,1] = 0$ and doing so would change the sparsity pattern of A.
- $A^2[1,2] = 1 \times 1 + 1 \times -1$ is structurally nonzero because we could make this number nonzero e.g. by changing $A[1,1] = 1$ to $A[1,1] = 2$ and doing so would not change the sparsity pattern of A.

Sparse LU Factorisation

Terminology: Derived sparsity pattern

Let B be a matrix derived from some other matrix A (e.g. $B = A^2$, $B = A^{-1}$, or B is one of the LU factors).

The set of all indices $(i, j) \in \{1, \ldots, n\}^2$ such that $B[i, j]$ is structurally nonzero is called the *sparsity pattern* or *structure* of the derived matrix B.

Example

Consider

$$A = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix}, \qquad A^2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The sparsity pattern of A^2 is

$$A^2 = \left(\begin{array}{c|c} 1 & \bullet \\ \hline & 2 \end{array}\right) \qquad \text{and not} \qquad A^2 = \left(\begin{array}{c|c} 1 & \\ \hline & 2 \end{array}\right)$$

because $A[1, 2]$ is structurally nonzero.

Sparse LU Factorisation

Resolving the ambiguity of "structure" for derived matrices

The above slides introduced two definitions for structure, namely one for matrices and another for derived matrices.

This strictly speaking means that "structure of a derived matrix B" can have two distinct meanings depending on whether we interpret B as a derived matrix or simply as a matrix.

Let us therefore establish the convention that the "derived" interpretation takes precedence whenever it is applicable.

Why study structure rather than values?

The fundamental idea behind the above notion of "structure" is that we only want to distinguish between whether an entry is guaranteed to be zero or could potentially be nonzero. Reasons for doing so include:

- Reasoning about structure is easier than reasoning about values.
- structure(B) provides a worst-case estimate for how much memory will be needed to store the derived matrix B.
- Cancellation (i.e. a structurally nonzero entry being zero) is unlikely.

Sparse LU Factorisation

The final term required to talk about sparsity is the following.

Terminology: Fill-in entry

Let B be a matrix derived from some other matrix A (e.g. $B = A^2$, $B = A^{-1}$, or B is one of the LU factors).

An index $(i, j) \in \text{structure}(B) \setminus \text{structure}(A)$ is called a *fill-in entry* of B.

Example 1

Consider

$$A = \left(\begin{array}{c|c|c} 1 & & \\ \hline \bullet & 2 & \\ \hline & \bullet & 3 \end{array}\right), \qquad A^2 = \left(\begin{array}{c|c|c} 1 & & \\ \hline \bullet & 2 & \\ \hline \bullet & \bullet & 3 \end{array}\right).$$

$B[3, 1]$ is a fill-in entry because $A[3, 1] = 0$ but $B[3, 1] \neq 0$.

This shows that matrix powers can create fill-in.

Example 2

$L[4, 3]$ in the LU factorisation from slide 14 is a fill-in entry.

This shows that the LU factorisation can create fill-in.

Sparse LU Factorisation

The importance of understanding fill-in

The fact that matrix powers and LU factorisations can create fill-in makes it hard to predict the runtime of such operations.

Example. `lu_benchmark()` shows that the runtime of computing an LU factorisation can differ by four orders of magnitude even if the input matrices have exactly the same size and number of nonzero entries.
"Order of magnitude" = power of 10.

This discrepancy arises because the factorisations of such matrices can have very different sparsity patterns, see `lu_structures()`.

The above example shows that it would be useful if we could estimate the amount of fill-in incurred by a sparse matrix operation before actually running that operation.

The following slides will show that we can do so by relating sparse matrices to graphs and then deducing the sparsity patterns of matrix powers, inverses and LU factorisations based on these graphs.

Sparse LU Factorisation

Let me begin by describing how we can translate matrices into graphs.

Def: Graph of a sparse matrix

The *graph* $G(A) = \big(V(A), E(A)\big)$ of a sparse matrix $A \in \mathbb{R}^{n\times n}$ is given by

$$V(A) = \{1, \ldots, n\}, \qquad E(A) = \big\{ j \to i \mid A[i,j] \neq 0 \big\}.$$

$V(A)$ denotes the set of vertices, $E(A)$ denotes the set of edges.
Note the transpose in $E(A)$: entry $A[i,j]$ corresponds to the edge $j \to i$.

Example

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) = 1 \quad 2 \quad 3 \quad 4$$

Note that the picture on the right omits the diagonal edges $i \to i$ for better readability.

Sparse LU Factorisation

Let me begin by describing how we can translate matrices into graphs.

Def: Graph of a sparse matrix

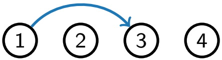
The *graph* $G(A) = \big(V(A), E(A)\big)$ of a sparse matrix $A \in \mathbb{R}^{n \times n}$ is given by

$$V(A) = \{1, \dots, n\}, \qquad E(A) = \big\{ j \to i \mid A[i,j] \neq 0 \big\}.$$

$V(A)$ denotes the set of vertices, $E(A)$ denotes the set of edges.
Note the transpose in $E(A)$: entry $A[i,j]$ corresponds to the edge $j \to i$.

Example

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



Note that the picture on the right omits the diagonal edges $i \to i$ for better readability.

Sparse LU Factorisation

Let me begin by describing how we can translate matrices into graphs.

Def: Graph of a sparse matrix

The *graph* $G(A) = \big(V(A), E(A)\big)$ of a sparse matrix $A \in \mathbb{R}^{n \times n}$ is given by

$$V(A) = \{1, \ldots, n\}, \qquad E(A) = \big\{ j \to i \mid A[i,j] \neq 0 \big\}.$$

$V(A)$ denotes the set of vertices, $E(A)$ denotes the set of edges.
Note the transpose in $E(A)$: entry $A[i,j]$ corresponds to the edge $j \to i$.

Example

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) = \begin{tikzcd} 1 \arrow[rr, bend left] & 2 \arrow[l, bend left] & 3 & 4 \end{tikzcd}$$

Note that the picture on the right omits the diagonal edges $i \to i$ for better readability.

Sparse LU Factorisation

Let me begin by describing how we can translate matrices into graphs.

Def: Graph of a sparse matrix

The *graph* $G(A) = \big(V(A), E(A)\big)$ of a sparse matrix $A \in \mathbb{R}^{n\times n}$ is given by

$$V(A) = \{1, \dots, n\}, \qquad E(A) = \big\{ j \to i \mid A[i,j] \neq 0 \big\}.$$

$V(A)$ denotes the set of vertices, $E(A)$ denotes the set of edges.
Note the transpose in $E(A)$: entry $A[i,j]$ corresponds to the edge $j \to i$.

Example

$$A = \begin{pmatrix} 1 & \bullet & & \\ & 2 & & \bullet \\ \bullet & & 3 & \\ & & \bullet & 4 \end{pmatrix} \qquad \longleftrightarrow \qquad G(A) =$$

$$\begin{tikzcd} 1 \arrow[rr, bend left] & 2 \arrow[l, bend left] & 3 \arrow[r, bend left] & 4 \end{tikzcd}$$

Note that the picture on the right omits the diagonal edges $i \to i$ for better readability.

Sparse LU Factorisation

Let me begin by describing how we can translate matrices into graphs.

Def: Graph of a sparse matrix

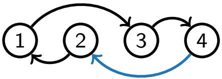
The *graph* $G(A) = \big(V(A), E(A)\big)$ of a sparse matrix $A \in \mathbb{R}^{n \times n}$ is given by

$$V(A) = \{1, \dots, n\}, \qquad E(A) = \big\{ j \to i \mid A[i,j] \neq 0 \big\}.$$

$V(A)$ denotes the set of vertices, $E(A)$ denotes the set of edges.
Note the transpose in $E(A)$: entry $A[i,j]$ corresponds to the edge $j \to i$.

Example

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



Note that the picture on the right omits the diagonal edges $i \to i$ for better readability.

Sparse LU Factorisation

Let me begin by describing how we can translate matrices into graphs.

Def: Graph of a sparse matrix

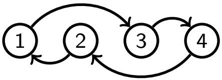
The *graph* $G(A) = \big(V(A), E(A)\big)$ of a sparse matrix $A \in \mathbb{R}^{n \times n}$ is given by

$$V(A) = \{1, \dots, n\}, \qquad E(A) = \big\{ j \to i \mid A[i,j] \neq 0 \big\}.$$

$V(A)$ denotes the set of vertices, $E(A)$ denotes the set of edges.
Note the transpose in $E(A)$: entry $A[i,j]$ corresponds to the edge $j \to i$.

Example

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



Note that the picture on the right omits the diagonal edges $i \to i$ for better readability.

Sparse LU Factorisation

Let me begin by describing how we can translate matrices into graphs.

Def: Graph of a sparse matrix

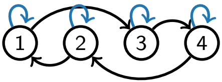
The *graph* $G(A) = \big(V(A), E(A)\big)$ of a sparse matrix $A \in \mathbb{R}^{n \times n}$ is given by

$$V(A) = \{1, \dots, n\}, \qquad E(A) = \big\{ j \to i \mid A[i,j] \neq 0 \big\}.$$

$V(A)$ denotes the set of vertices, $E(A)$ denotes the set of edges.
Note the transpose in $E(A)$: entry $A[i,j]$ corresponds to the edge $j \to i$.

Example

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



Note that the picture on the right omits the diagonal edges $i \to i$ for better readability.

Sparse LU Factorisation

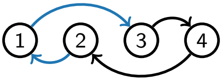
The following definition will be important to deduce the sparsity patterns of derived matrices from the graph of the original matrix.

Def: Paths

A *path* in a graph $G = (V, E)$ is an ordered sequence of vertices $k_0, \dots, k_p \in V$ such that $k_{q-1} \to k_q \in E$ for all $q \in \{1, \dots, p\}$.
The number of edges p is called the *length* of the path.

Example

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



$2 \to 1 \to 3$ is a path of length 2.

Sparse LU Factorisation

We now have the tools to formulate our first fill-in-describing result.

Path theorem for matrix powers

$$A^p[i,j] \neq 0 \quad \iff \quad \exists \text{ path } j \to i \text{ of length } p \text{ in } G(A).$$

Proof. We have

$$A^2[i,j] = \sum_k A[i,k]\, A[k,j].$$

Each term in this sum is nonzero iff $j \to k \to i$ is a path in $G(A)$.

Generalising this proof to arbitrary powers p, we observe that

$$A^p[i,j] = \sum_{k_{p-1}} \dots \sum_{k_1} A[i,k_{p-1}] \, \dots \, A[k_a,k_{a-1}] \, \dots \, A[k_1,j].$$

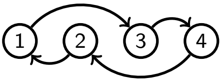
is nonzero iff $j \to k_1 \to \dots \to k_{p-1} \to i$ is a path in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



We observe:

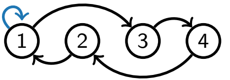
- $A^2[1,1] \neq 0$ because $1 \to 1 \to 1$ is a path of length 2 in $G(A)$.
- $A^2[2,1] = 0$ because the only path connecting 1 to 2, namely $1 \to 3 \to 4 \to 2$, has length 3.
  Note that strictly speaking there are more paths from 1 to 2, namely we can extend the above path with diagonal edges. Such paths will not create additional fill-in, however, because adding diagonal edges only makes a path longer.
- $A^2[3,1] \neq 0$ because $1 \to 1 \to 3$ is a path of length 2 in $G(A)$.
- $A^2[4,1] \neq 0$ because $1 \to 3 \to 2$ is a path of length 2 in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \begin{pmatrix} \color{blue}{1} & \bullet & & \\ & 2 & & \bullet \\ \bullet & & 3 & \\ & & \bullet & 4 \end{pmatrix} \qquad \longleftrightarrow \qquad G(A) =$$



We observe:

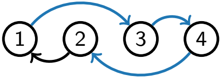
- $\color{blue}{A^2[1,1] \neq 0}$ because $\color{blue}{1 \to 1 \to 1}$ is a path of length 2 in $G(A)$.
- $A^2[2,1] = 0$ because the only path connecting 1 to 2, namely $1 \to 3 \to 4 \to 2$, has length 3.
  Note that strictly speaking there are more paths from 1 to 2, namely we can extend the above path with diagonal edges. Such paths will not create additional fill-in, however, because adding diagonal edges only makes a path longer.
- $A^2[3,1] \neq 0$ because $1 \to 1 \to 3$ is a path of length 2 in $G(A)$.
- $A^2[4,1] \neq 0$ because $1 \to 3 \to 2$ is a path of length 2 in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \begin{pmatrix} 1 & \bullet & & \\ & 2 & & \bullet \\ \bullet & & 3 & \\ & & \bullet & 4 \end{pmatrix} \quad \longleftrightarrow \quad G(A) =$$



We observe:

- $A^2[1,1] \neq 0$ because $1 \to 1 \to 1$ is a path of length 2 in $G(A)$.
- $A^2[2,1] = 0$ because the only path connecting 1 to 2, namely $1 \to 3 \to 4 \to 2$, has length 3.

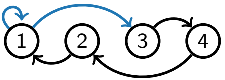
  Note that strictly speaking there are more paths from 1 to 2, namely we can extend the above path with diagonal edges. Such paths will not create additional fill-in, however, because adding diagonal edges only makes a path longer.
- $A^2[3,1] \neq 0$ because $1 \to 1 \to 3$ is a path of length 2 in $G(A)$.
- $A^2[4,1] \neq 0$ because $1 \to 3 \to 2$ is a path of length 2 in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



.

We observe:

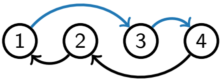
- $A^2[1,1] \neq 0$ because $1 \to 1 \to 1$ is a path of length 2 in $G(A)$.
- $A^2[2,1] = 0$ because the only path connecting 1 to 2, namely $1 \to 3 \to 4 \to 2$, has length 3.
  Note that strictly speaking there are more paths from 1 to 2, namely we can extend the above path with diagonal edges. Such paths will not create additional fill-in, however, because adding diagonal edges only makes a path longer.
- $A^2[3,1] \neq 0$ because $1 \to 1 \to 3$ is a path of length 2 in $G(A)$.
- $A^2[4,1] \neq 0$ because $1 \to 3 \to 2$ is a path of length 2 in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \begin{pmatrix} 1 & \bullet & & \\ & 2 & & \bullet \\ \bullet & & 3 & \\ & & \bullet & 4 \end{pmatrix} \quad \longleftrightarrow \quad G(A) =$$



.

We observe:

- $A^2[1,1] \neq 0$ because $1 \to 1 \to 1$ is a path of length 2 in $G(A)$.
- $A^2[2,1] = 0$ because the only path connecting 1 to 2, namely $1 \to 3 \to 4 \to 2$, has length 3.
  Note that strictly speaking there are more paths from 1 to 2, namely we can extend the above path with diagonal edges. Such paths will not create additional fill-in, however, because adding diagonal edges only makes a path longer.
- $A^2[3,1] \neq 0$ because $1 \to 1 \to 3$ is a path of length 2 in $G(A)$.
- $A^2[4,1] \neq 0$ because $1 \to 3 \to 2$ is a path of length 2 in $G(A)$.

Sparse LU Factorisation

Example (continued)

Continuing the above analysis for all the remaining entries, we conclude

$$A^2 = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \color{red}{\bullet} \\ \hline & 2 & \color{red}{\bullet} & \bullet \\ \hline \bullet & \color{red}{\bullet} & 3 & \\ \hline \color{red}{\bullet} & & \bullet & 4 \end{array}\right) .$$

The $\bullet$ indicate nonzero entries which were nonzero already in A.
The $\color{red}{\bullet}$ indicate fill-in.

Sparse LU Factorisation

Path theorem for inverses

$$A^{-1}[i,j] \neq 0 \quad \iff \quad \exists \text{ path } j \to i \text{ in } G(A).$$

Proof (not examinable).

Let $p(x) = \sum_{k=0}^{n-1} c_k \, x^k$ be the unique polynomial which interpolates $\frac{1}{x}$ in all the n eigenvalues of A. We then have

$$A^{-1} = p(A) = \sum_{k=0}^{n-1} c_k \, A^k,$$

which shows that $A^{-1}[i,j] \neq 0$ if and only if there is a path from j to i of any arbitrary length $k \in \{0, \ldots, n-1\}$.

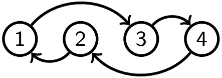
(Do not worry in case the claim $A^{-1} = p(A)$ is not clear to you. We will discuss matrix polynomials and their relation to inverses in Lecture 5.)

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



.

We observe that any vertex i is reachable from any other vertex j in $G(A)$; hence we have

$$A^{-1} = \left(\begin{array}{c|c|c|c} 1 & \bullet & {\color{red}\bullet} & {\color{red}\bullet} \\ \hline {\color{red}\bullet} & 2 & {\color{red}\bullet} & \bullet \\ \hline \bullet & {\color{red}\bullet} & 3 & {\color{red}\bullet} \\ \hline {\color{red}\bullet} & {\color{red}\bullet} & \bullet & 4 \end{array}\right) .$$

The $\bullet$ indicate nonzero entries which were nonzero already in A.
The ${\color{red}\bullet}$ indicate fill-in.

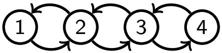
Sparse LU Factorisation

Corollaries of path theorem for inverses

The path theorem for inverses provides simple proofs for a number of useful results which would be much more tedious to show otherwise.

- The inverse of a tridiagonal matrix

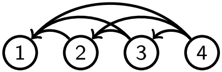
$$A = \begin{pmatrix} 1 & \bullet & & \\ \bullet & 2 & \bullet & \\ & \bullet & 3 & \bullet \\ & & \bullet & 4 \end{pmatrix} \qquad \longleftrightarrow \qquad G(A) =$$



  is dense because any pair of vertices i, j is connected.

- The inverse of an upper-triangular matrix

$$A = \begin{pmatrix} 1 & \bullet & \bullet & \bullet \\ & 2 & \bullet & \bullet \\ & & 3 & \bullet \\ & & & 4 \end{pmatrix} \qquad \longleftrightarrow \qquad G(A) =$$



  is upper triangular because all edges in $G(A)$ go only from right to left and hence paths j to i exist only if $i < j$.

Sparse LU Factorisation

We now know how to predict fill-in in matrix powers and inverses.
Let us therefore move on to predicting fill-in in the LU factorisation next.
Doing so requires a new notion of path defined as follows.

Def: Fill path

Path $j \to k_1 \to \ldots \to k_p \to i$ in $G(A)$ such that $k_1, \ldots, k_p < \min\{i, j\}$.

We then have the following result.

Fill path theorem

Let $LU = A$ be the LU factorisation of A. Then,

$$(L + U)[i, j] \neq 0 \quad \iff \quad \exists \text{ fill path } j \to i \text{ in } G(A).$$

Proof (not examinable). A proof of this result is provided on slide 33, but I will not discuss it in class. I recommend you ignore this proof unless you are truly interested.

Sparse LU Factorisation

Describing the sparsity pattern of an LU factorisation

The above result describes the sparsity pattern of $L + U$ rather than the two sparsity patterns of L and U. Doing so is convenient because it means we have to think about only one rather than two matrices, and it does not discard any information because we have

$$(L+U)[i,j] \neq 0 \quad \iff \quad \begin{cases} L[i,j] \neq 0 & \text{if } i \geq j, \\ U[i,j] \neq 0 & \text{if } i \leq j. \end{cases}$$

Note that we do not have to worry about cancellation in the diagonal entries

$$(L+U)[i,i] = L[i,i] + U[i,i]$$

because cancellation does not change whether an entry is structurally nonzero.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) = \begin{tikzcd} \textcircled{1} \arrow[rr, bend left] & \textcircled{2} \arrow[l, bend left] & \textcircled{3} \arrow[r, bend left] & \textcircled{4} \arrow[ll, bend left] \end{tikzcd}.$$

We observe:

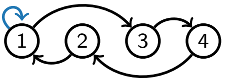
- $U[1,1] \neq 0$ because $1 \to 1$ is a fill path in $G(A)$.
- $L[2,1] = 0$ because $1 \to 3 \to 4 \to 2$ is not a fill path $(3, 4 > 1)$.
- $L[3,1] \neq 0$ because $1 \to 3$ is a fill path in $G(A)$.
- $L[4,1] = 0$ because $1 \to 3 \to 4$ is not a fill path $(3 > 1)$.
- $U[1,2], U[2,2] \neq 0$ because $2 \to 1$ and $2 \to 2$ are fill paths in $G(A)$.
- $L[3,2] \neq 0$ because $2 \to 1 \to 3$ is a fill path in $G(A)$.
- The only other fill-in entry is $U[3,4]$ because $4 \to 2 \to 1 \to 3$ is a fill path in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



.

We observe:

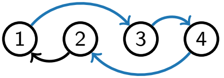
- $U[1,1] \neq 0$ because $1 \to 1$ is a fill path in $G(A)$.
- $L[2,1] = 0$ because $1 \to 3 \to 4 \to 2$ is not a fill path $(3, 4 > 1)$.
- $L[3,1] \neq 0$ because $1 \to 3$ is a fill path in $G(A)$.
- $L[4,1] = 0$ because $1 \to 3 \to 4$ is not a fill path $(3 > 1)$.
- $U[1,2], U[2,2] \neq 0$ because $2 \to 1$ and $2 \to 2$ are fill paths in $G(A)$.
- $L[3,2] \neq 0$ because $2 \to 1 \to 3$ is a fill path in $G(A)$.
- The only other fill-in entry is $U[3,4]$ because $4 \to 2 \to 1 \to 3$ is a fill path in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \begin{pmatrix} 1 & \bullet & & \\ & 2 & & \bullet \\ \bullet & & 3 & \\ & & \bullet & 4 \end{pmatrix} \quad \longleftrightarrow \quad G(A) =$$



We observe:

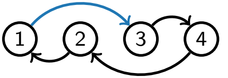
- $U[1,1] \neq 0$ because $1 \to 1$ is a fill path in $G(A)$.
- $L[2,1] = 0$ because $1 \to 3 \to 4 \to 2$ is not a fill path $(3,4 > 1)$.
- $L[3,1] \neq 0$ because $1 \to 3$ is a fill path in $G(A)$.
- $L[4,1] = 0$ because $1 \to 3 \to 4$ is not a fill path $(3 > 1)$.
- $U[1,2], U[2,2] \neq 0$ because $2 \to 1$ and $2 \to 2$ are fill paths in $G(A)$.
- $L[3,2] \neq 0$ because $2 \to 1 \to 3$ is a fill path in $G(A)$.
- The only other fill-in entry is $U[3,4]$ because $4 \to 2 \to 1 \to 3$ is a fill path in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



We observe:

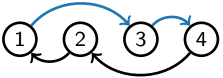
- $U[1,1] \neq 0$ because $1 \to 1$ is a fill path in $G(A)$.
- $L[2,1] = 0$ because $1 \to 3 \to 4 \to 2$ is not a fill path $(3, 4 > 1)$.
- $L[3,1] \neq 0$ because $1 \to 3$ is a fill path in $G(A)$.
- $L[4,1] = 0$ because $1 \to 3 \to 4$ is not a fill path $(3 > 1)$.
- $U[1,2], U[2,2] \neq 0$ because $2 \to 1$ and $2 \to 2$ are fill paths in $G(A)$.
- $L[3,2] \neq 0$ because $2 \to 1 \to 3$ is a fill path in $G(A)$.
- The only other fill-in entry is $U[3,4]$ because $4 \to 2 \to 1 \to 3$ is a fill path in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \begin{pmatrix} 1 & \bullet & & \\ & 2 & & \bullet \\ \bullet & & 3 & \\ & & \bullet & 4 \end{pmatrix} \qquad \longleftrightarrow \qquad G(A) =$$



.

We observe:

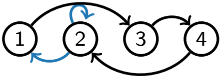
- $U[1,1] \neq 0$ because $1 \to 1$ is a fill path in $G(A)$.
- $L[2,1] = 0$ because $1 \to 3 \to 4 \to 2$ is not a fill path $(3, 4 > 1)$.
- $L[3,1] \neq 0$ because $1 \to 3$ is a fill path in $G(A)$.
- $L[4,1] = 0$ because $1 \to 3 \to 4$ is not a fill path $(3 > 1)$.
- $U[1,2], U[2,2] \neq 0$ because $2 \to 1$ and $2 \to 2$ are fill paths in $G(A)$.
- $L[3,2] \neq 0$ because $2 \to 1 \to 3$ is a fill path in $G(A)$.
- The only other fill-in entry is $U[3,4]$ because $4 \to 2 \to 1 \to 3$ is a fill path in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



We observe:

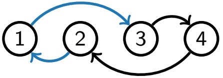
- $U[1,1] \neq 0$ because $1 \to 1$ is a fill path in $G(A)$.
- $L[2,1] = 0$ because $1 \to 3 \to 4 \to 2$ is not a fill path $(3, 4 > 1)$.
- $L[3,1] \neq 0$ because $1 \to 3$ is a fill path in $G(A)$.
- $L[4,1] = 0$ because $1 \to 3 \to 4$ is not a fill path $(3 > 1)$.
- $U[1,2], U[2,2] \neq 0$ because $2 \to 1$ and $2 \to 2$ are fill paths in $G(A)$.
- $L[3,2] \neq 0$ because $2 \to 1 \to 3$ is a fill path in $G(A)$.
- The only other fill-in entry is $U[3,4]$ because $4 \to 2 \to 1 \to 3$ is a fill path in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \begin{pmatrix} 1 & \bullet & & \\ & 2 & & \bullet \\ \bullet & & 3 & \\ & & \bullet & 4 \end{pmatrix} \qquad \longleftrightarrow \qquad G(A) =$$



We observe:

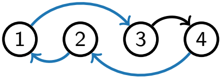
- $U[1,1] \neq 0$ because $1 \to 1$ is a fill path in $G(A)$.
- $L[2,1] = 0$ because $1 \to 3 \to 4 \to 2$ is not a fill path $(3, 4 > 1)$.
- $L[3,1] \neq 0$ because $1 \to 3$ is a fill path in $G(A)$.
- $L[4,1] = 0$ because $1 \to 3 \to 4$ is not a fill path $(3 > 1)$.
- $U[1,2], U[2,2] \neq 0$ because $2 \to 1$ and $2 \to 2$ are fill paths in $G(A)$.
- $L[3,2] \neq 0$ because $2 \to 1 \to 3$ is a fill path in $G(A)$.
- The only other fill-in entry is $U[3,4]$ because $4 \to 2 \to 1 \to 3$ is a fill path in $G(A)$.

Sparse LU Factorisation

Example

Recall from slide 21 the matrix

$$A = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & & 3 & \\ \hline & & \bullet & 4 \end{array}\right) \qquad \longleftrightarrow \qquad G(A) =$$



.

We observe:

- $U[1,1] \neq 0$ because $1 \to 1$ is a fill path in $G(A)$.
- $L[2,1] = 0$ because $1 \to 3 \to 4 \to 2$ is not a fill path $(3, 4 > 1)$.
- $L[3,1] \neq 0$ because $1 \to 3$ is a fill path in $G(A)$.
- $L[4,1] = 0$ because $1 \to 3 \to 4$ is not a fill path $(3 > 1)$.
- $U[1,2], U[2,2] \neq 0$ because $2 \to 1$ and $2 \to 2$ are fill paths in $G(A)$.
- $L[3,2] \neq 0$ because $2 \to 1 \to 3$ is a fill path in $G(A)$.
- The only other fill-in entry is $U[3,4]$ because $4 \to 2 \to 1 \to 3$ is a fill path in $G(A)$.

Sparse LU Factorisation

Example (continued)

We thus conclude that the sparsity pattern of $L + U$ is given by

$$L + U = \left(\begin{array}{c|c|c|c} 1 & \bullet & & \\ \hline & 2 & & \bullet \\ \hline \bullet & {\color{red}\bullet} & 3 & {\color{red}\bullet} \\ \hline & & \bullet & 4 \end{array}\right).$$

The $\bullet$ indicate nonzero entries which were nonzero already in A.
The ${\color{red}\bullet}$ indicate fill-in.

Sparse LU Factorisation

The proof of the fill path theorem is based on the following result.

Lemma (not examinable, ignore unless you are interested)
Let $LU = A$ be the LU factorisation of a matrix $A \in \mathbb{R}^{n\times n}$, let $i, j \in \{1, \ldots, n\}$ and set $\ell = \big\{1, \ldots, \min\{i, j\} - 1\big\}$. We then have,

$$\begin{aligned} U[i,j] &= A[i,j] - A[i,\ell]\,A[\ell,\ell]^{-1}\,A[\ell,j] \qquad &&\text{for } i \le j, \\ L[i,j]\,U[j,j] &= A[i,j] - A[i,\ell]\,A[\ell,\ell]^{-1}\,A[\ell,j] \qquad &&\text{for } i \ge j. \end{aligned}$$

Proof. Consider the block LU factorisation

$$\begin{pmatrix} A[\ell,\ell] & A[\ell,\bar r] \\ A[\bar r,\ell] & A[\bar r,\bar r] \end{pmatrix} = \begin{pmatrix} I & \\ A[\bar r,\ell]\,A[\ell,\ell]^{-1} & I \end{pmatrix} \begin{pmatrix} A[\ell,\ell] & A[\ell,\bar r] \\ & A[\bar r,\bar r] - A[\bar r,\ell]\,A[\ell,\ell]^{-1}\,A[\ell,\bar r] \end{pmatrix} \tag{1}$$

where $\bar r = \big\{\min\{i, j\}, \ldots, n\big\}$, The full factorisation is then given by

$$\begin{pmatrix} A[\ell,\ell] & A[\ell,\bar r] \\ A[\bar r,\ell] & A[\bar r,\bar r] \end{pmatrix} = \begin{pmatrix} L_1 & \\ A[\bar r,\ell]\,A[\ell,\ell]^{-1} L_1 & L_2 \end{pmatrix} \begin{pmatrix} U_1 & L_1^{-1}\,A[\ell,\bar r] \\ & U_2 \end{pmatrix}$$

where $\quad L_1 U_1 = A[\ell,\ell] \quad$ and $\quad L_2 U_2 = A[\bar r,\bar r] - A[\bar r,\ell]\,A[\ell,\ell]^{-1}\,A[\ell,\bar r]$ are the LU factorisations of the top-left and bottom-right blocks of the U-factor in (1). The claim follows by noting that $L[i,j] = L_2[i,j]$ and $U[i,j] = U_2[i,j]$ have the given form.

Sparse LU Factorisation

Proof of fill path theorem (not examinable, ignore unless you are interested).

According to the lemma on the previous slide, we have

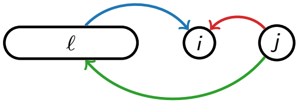
$$\begin{aligned} U[i,j] &= A[i,j] - A[i,\ell]\,A[\ell,\ell]^{-1}\,A[\ell,j] && \text{for } i \leq j, \\ L[i,j]\,U[j,j] &= A[i,j] - A[i,\ell]\,A[\ell,\ell]^{-1}\,A[\ell,j] && \text{for } i \geq j. \end{aligned}$$

The first term makes $U[i,j]$ / $L[i,j]\,U[j,j]$ nonzero if there is a fill path $j \to i$ of length 1, and the second term makes $U[i,j]$ / $L[i,j]\,U[j,j]$ nonzero if there is a fill path $j \to i$ of length > 1.

It then remains to observe that

$$L[i,j]\,U[j,j] \neq 0 \quad \iff \quad L[i,j]$$

since $U[j,j]$ is a pivot entry and hence we necessarily have $U[j,j] \neq 0$.



Sparse LU Factorisation

[To be continued]