Methods to Invert a Matrix

The approaches available to find the inverse of a matrix are extensive and diverse. All methods seek to solve a linear system of equations that can be expressed in a matrix format as

\[
[A] \{x\} = \{b\}
\]

for the unknowns contained in the vector \(\{x\}\), i.e.,

\[
\{x\} = [A]^{-1} \{b\}
\]

The methods used to accomplish this can be loosely grouped into the following three categories:

- methods that explicitly calculate \(\{x\}\)
- methods that implicitly calculate \(\{x\}\), and
- iterative methods that calculate \(\{x\}\)

Of course hybrid methods exist that are combinations of two or methods in the categories listed above.
Consider the following list of methods which is not comprehensive:

1. Explicit methods for sparse matrices – includes Cramer’s rule which is a specific case of using self adjoint matrices. Variations include:
   a. Gauss elimination
      - Row echelon form; all entries below a nonzero entry in the matrix are zero
      - Bareiss algorithm; every element computed is the determinant of a $[A]$
      - Tri-diagonal matrix algorithm; special form of Gauss elimination
   b. Gauss-Jordan elimination

2. LDU decomposition – an implicit method that factors $[A]$ into a product of a lower and upper triangular matrices and a diagonal matrix. Variations include
   a. LU reduction – a special parallelized version of a LDU decomposition algorithm
      - Crout matrix decomposition is a special type of LU decomposition

3. Cholesky LDL decomposition – an implicit method that decomposes $[A]$, when it is a positive-definite matrix, into the product of a lower triangular matrix, a diagonal matrix and the conjugate transpose
   a. Frontal solvers used in finite element methods
   b. Nested dissection – for symmetric matrices, based on graph partitioning
   c. Minimum degree algorithm
   d. Symbolic Cholesky decomposition
5. Iterative methods:
   a. Gauss-Seidel methods
      • Successive over relaxation (SOR)
      • Back fit algorithms
   b. Conjugate gradient methods (CG) – used often in optimization problems
      • Nonlinear conjugate gradient method
      • Biconjugate gradient method (BiCG)
      • Biconjugate gradient stabilized method (BiCGSTAB)
      • Conjugate residual method
   c. Jacobi method
   d. Modified Richardson iteration
   e. Generalized minimal residual method (GMRES) – based on the Arnoldi iteration
   f. Chebyshev iteration – avoids inner products but needs bounds on the spectrum
   g. Stone's method (SIP: Strongly Implicit Procedure) – uses an incomplete LU decomposition
   h. Kaczmarz method
   i. Iterative refinement – procedure to turn an inaccurate solution in a more accurate one

6. Levinson recursion – for Toeplitz matrices
7. SPIKE algorithm – hybrid parallel solver for narrow-banded matrices

Details and derivations are presented for several of the methods in this section of the notes. Concepts associated with determinants, cofactors and minors are presented first.
The Determinant of a Square Matrix

A square matrix of order $n$ (an $n \times n$ matrix), i.e.,

\[
[A] = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}
\]

possesses a uniquely defined scalar that is designated as the determinant of the matrix, or merely the determinant

\[
\text{det} [A] = |A|
\]

Observe that only square matrices possess determinants.
Vertical lines and not brackets designate a determinant, and while $det[A]$ is a number and has no elements, it is customary to represent it as an array of elements of the matrix

$$det[A] = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$

A general procedure for finding the value of a determinant sometimes is called “expansion by minors.” We will discuss this method after going over some ground rules for operating with determinants.
Rules for Operating with Determinants

Rules pertaining to the manipulation of determinants are presented in this section without formal proof. Their validity is demonstrated through examples presented at the end of the section.

**Rule #1:** Interchanging any row (or column) of a determinant with its immediate adjacent row (or column) flips the sign of the determinant.

**Rule #2:** The multiplication of any single row (column) of determinant by a scalar constant is equivalent to the multiplication of the determinant by the scalar.

**Rule #3:** If any two rows (columns) of a determinant are identical, the value of the determinant is zero and the matrix from which the determinant is derived is said to be singular.

**Rule #4:** If any row (column) of a determinant contains nothing but zeroes then the matrix from which the determinant is derived is singular.

**Rule #5:** If any two rows (two columns) of a determinant are proportional, i.e., the two rows (two columns) are linearly dependent, then the determinant is zero and the matrix from which the determinant is derived is singular.
Rule #6: If the elements of any row (column) of a determinant are added to or subtracted from the corresponding elements of another row (column) the value of the determinant is unchanged.

Rule #6a: If the elements of any row (column) of a determinant are multiplied by a constant and then added or subtracted from the corresponding elements of another row (column), the value of the determinant is unchanged.

Rule #7: The value of the determinant of a diagonal matrix is equal to the product of the terms on the diagonal.

Rule #8: The value for the determinant of a matrix is equal to the value of the determinant of the transpose of the matrix.

Rule #9: The determinant of the product of two matrices is equal to the product of the determinants of the two matrices.

Rule #10: If the determinant of the product of two square matrices is zero, then at least one of the two matrices is singular.

Rule #11: If an $m \times n$ rectangular matrix $A$ is post-multiplied by an $n \times m$ rectangular matrix $B$, the resulting square matrix $[C] = [A][B]$ of order $m$ will, in general, be singular if $m > n$. 
In Class Example
Minors and Cofactors

Consider the $n^{th}$ order determinant:

$$
\det[A] = \begin{vmatrix}
    a_{11} & a_{12} & \cdots & a_{1n} \\
    a_{21} & a_{22} & \cdots & a_{2n} \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{n1} & a_{n2} & \cdots & a_{nn}
\end{vmatrix}
$$

The $m^{th}$ order minor of the $n^{th}$ order matrix is the determinant formed by deleting $(n - m)$ rows and $(n - m)$ columns in the $n^{th}$ order determinant. For example, the minor $|M|_{ir}$ of the determinant $|A|$ is formed by deleting the $i^{th}$ row and the $r^{th}$ column. Because $|A|$ is an $n^{th}$ order determinant, the minor $|M|_{ir}$ is of order $m = n - 1$ and contains $m^2$ elements.

In general, a minor formed by deleting $p$ rows and $p$ columns in the $n^{th}$ ordered determinant $|A|$ is an $(n - p)^{th}$ order minor. If $p = n - 1$, the minor is of first order and contains only a single element from $|A|$.

From this it is easy to see that the determinant $|A|$ contains $n^2$ elements of first order minors, each containing a single element.
When dealing with minors other than the \((n-1)\)th order, the designation of the eliminated rows and columns of the determinant \(|A|\) must be considered carefully. It is best to consider consecutive rows \(j, k, l, m \ldots\) and consecutive columns \(r, s, t, u \ldots\) so that the \((n-1)\)th, \((n-2)\)th, and \((n-3)\)th order minors would be designated, respectively, as \(|M|_{j,r}, |M|_{jk,rs}\) and \(|M|_{jkl,rsr}\).

The complementary minor, or the complement of the minor, is designated as \(|N|\) (with subscripts). This minor is the determinant formed by placing the elements that lie at the intersections of the deleted rows and columns of the original determinant into a square array in the same order that they appear in the original determinant. For example, given the determinant from the previous page, then

\[
|N|_{23} = a_{23}
\]

\[
|N|_{23,31} = \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix}
\]
The algebraic complement of the minor $|M|$ is the “signed” complementary minor. If a minor is obtained by deleting rows $i, k, l$ and columns $r, s, t$ from the determinant $|A|$ the minor is designated

$$|M|_{ikl,rst}$$

the complementary minor is designated

$$|N|_{ikl,rst}$$

and the algebraic complement is designated

$$(-1)^{i+k+l+r+s+t} |N|_{ikl,rst}$$

The cofactor, designated with capital letters and subscripts, is the signed $(n - 1)^{th}$ minor formed from the $n^{th}$ order determinant. Suppose the that the $(n - 1)^{th}$ order minor is formed by deleting the $i^{th}$ row and $j^{th}$ column from the determinant $|A|$. Then corresponding cofactor is

$$A_{ij} = (-1)^{i+j} |M|_{ij}$$
Lecture 3: Determinants & Matrix Inversion

Observe the cofactor has no meaning for minors with orders smaller than \((n - 1)\) unless the minor itself is being treated as a determinant of order one less than the determinant \(|A|\) from which it was derived.

Also observe that when the minor is order \((n - 1)\), the product of the cofactor and the complement is equal to the product of the minor and the algebraic complement.

We can assemble the cofactors of a square matrix of order \(n\) (an \(n \times n\) matrix) into a square cofactor matrix, i.e.,

\[
[A]^C = \begin{bmatrix}
A_{11} & A_{12} & \cdots & A_{1n} \\
A_{21} & A_{22} & \cdots & A_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
A_{n1} & A_{n2} & \cdots & A_{nn}
\end{bmatrix}
\]

So when the elements of a matrix are denoted with capital letters the matrix represents a matrix of cofactors for another matrix.
Lecture 3: Determinants & Matrix Inversion

In Class Example
**Rules for Operations with Cofactors**

The determinant for a three by three matrix can be computed via the expansion of the matrix by minors as follows:

\[
\det[A] = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{21} \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} + a_{31} \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix}
\]

This can be confirmed using the classic expansion technique for 3 x 3 determinants. This expression can be rewritten as:

\[
\det[A] = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} M_{11} - a_{21} M_{21} + a_{31} M_{31}
\]

or using cofactor notation:

\[
\det[A] = |A| = a_{11} A_{11} + a_{21} A_{21} + a_{31} A_{31}
\]
Rule #12: A determinant may be evaluated by summing the products of every element in any row or column by the respective cofactor. This is known as Laplace’s expansion.

Rule #13: If all cofactors in a row or a column are zero, the determinant is zero and matrix from which they are derived is singular.

Rule #14: If the elements in a row or a column of a determinant are multiplied by cofactors of the corresponding elements of a different row or column, the resulting sum of these products are zero.
The Adjoint Matrix

The adjoint matrix is the matrix of transposed cofactors. If we have an \( n^{th} \) order matrix

\[
\begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1n} \\
a_{21} & a_{22} & \ldots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \ldots & a_{nn}
\end{bmatrix}
\]

this matrix possess the following matrix of cofactors

\[
\begin{bmatrix}
A_{11} & A_{12} & \ldots & A_{1n} \\
A_{21} & A_{22} & \ldots & A_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
A_{n1} & A_{n2} & \ldots & A_{nn}
\end{bmatrix}
\]
and the adjoint of the matrix is defined as the transpose of the cofactor matrix

\[
adj [A] = \left[ [A]^c \right]^T
\]

\[
= \begin{bmatrix}
A_{11} & A_{21} & \cdots & A_{n1} \\
A_{12} & A_{22} & \cdots & A_{n2} \\
\vdots & \vdots & \ddots & \vdots \\
A_{1n} & A_{2n} & \cdots & A_{nn}
\end{bmatrix}
\]

We will show in the next section that finding the inverse of a square matrix can be accomplished with the following expression:

\[
[A]^{-1} = \frac{adj [A]}{|A|}
\]

For a 3 x 3 matrix this is known as Cramer’s rule.
Direct Inversion Method

Suppose an $n \times n$ matrix is post multiplied by its adjoint and the resulting $n \times n$ matrix is identified as $[P]$

\[
[P] = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}
\begin{bmatrix}
A_{11} & A_{21} & \cdots & A_{1n} \\
A_{12} & A_{22} & \cdots & A_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
A_{n1} & A_{n2} & \cdots & A_{nn}
\end{bmatrix}
\]

The elements of matrix $[P]$ are divided into two categories, i.e., elements that lie along the diagonal

\[
p_{11} = a_{11} A_{11} + a_{12} A_{12} + \cdots + a_{1n} A_{1n} \\
p_{22} = a_{21} A_{21} + a_{22} A_{22} + \cdots + a_{2n} A_{2n} \\
\vdots \\
p_{nn} = a_{n1} A_{n1} + a_{n2} A_{n2} + \cdots + a_{nn} A_{nn}
\]
and those that do not

\[
\begin{align*}
p_{12} & = a_{11} A_{21} + a_{12} A_{22} + \ldots + a_{1n} A_{2n} \\
p_{13} & = a_{11} A_{31} + a_{12} A_{32} + \ldots + a_{1n} A_{3n} \\
\vdots & = \vdots \quad \vdots \quad \ldots \quad \vdots \\
p_{21} & = a_{21} A_{11} + a_{22} A_{12} + \ldots + a_{2n} A_{1n} \\
\vdots & = \vdots \quad \vdots \quad \ldots \quad \vdots \\
p_{32} & = a_{31} A_{21} + a_{32} A_{22} + \ldots + a_{3n} A_{2n} \\
\vdots & = \vdots \quad \vdots \quad \ldots \quad \vdots \\
p_{n3} & = a_{n1} A_{31} + a_{n2} A_{32} + \ldots + a_{nn} A_{3n} \\
\vdots & = \vdots \quad \vdots \quad \ldots \quad \vdots 
\end{align*}
\]

The elements of \([P]\) that lie on the diagonal are all equal to the determinant of \([A]\) (see Rule #12 and recognize the Laplace expansion for each diagonal value). Note that the non-diagonal elements will be equal to zero since they involve the expansion of one row of matrix A with the cofactors of an entirely different row (see Rule #14).
Lecture 3: Determinants & Matrix Inversion

Thus

\[
p_{11} = a_{11} A_{11} + a_{12} A_{12} + \ldots + a_{1n} A_{1n} = |A|
\]

\[
p_{22} = a_{21} A_{21} + a_{22} A_{22} + \ldots + a_{2n} A_{2n} = |A|
\]

\[
\vdots \quad \vdots \quad \vdots \quad \ldots \quad \vdots
\]

\[
p_{nn} = a_{n1} A_{n1} + a_{n2} A_{n2} + \ldots + a_{nn} A_{nn} = |A|
\]

and

\[
p_{12} = a_{11} A_{21} + a_{12} A_{22} + \ldots + a_{1n} A_{2n} = 0
\]

\[
p_{13} = a_{11} A_{31} + a_{12} A_{32} + \ldots + a_{1n} A_{3n} = 0
\]

\[
\vdots \quad \vdots \quad \vdots \quad \ldots \quad \vdots
\]

\[
p_{21} = a_{21} A_{11} + a_{22} A_{12} + \ldots + a_{2n} A_{1n} = 0
\]

\[
\vdots \quad \vdots \quad \vdots \quad \ldots \quad \vdots
\]

\[
p_{32} = a_{31} A_{21} + a_{32} A_{22} + \ldots + a_{3n} A_{2n} = 0
\]

\[
\vdots \quad \vdots \quad \vdots \quad \ldots \quad \vdots
\]

\[
p_{n3} = a_{n1} A_{31} + a_{n2} A_{32} + \ldots + a_{nn} A_{3n} = 0
\]

\[
\vdots \quad \vdots \quad \vdots \quad \ldots \quad \vdots
\]
Lecture 3: Determinants & Matrix Inversion

which leads to

\[
[P] = [A] \text{adj}[A] = \begin{bmatrix}
|A| & 0 & \ldots & 0 \\
0 & |A| & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & |A|
\end{bmatrix} = |A|[I]
\]

or

\[
[I] = [A] \frac{\text{adj}[A]}{|A|}
\]

When this expression is compared to

\[
[I] = [A][A]^{-1}
\]

then it is evident that

\[
[A]^{-1} = \frac{\text{adj}[A]}{|A|}
\]

The inverse exists only when the determinant of \( A \) is not zero, i.e., when \( A \) is not singular.
Lecture 3: Determinants & Matrix Inversion

In Class Example
The direct inversion method presented above is referred to as a “brute force” approach. From a computational standpoint the method is inefficient (but doable) when the matrix is quite large. There are more efficient methods for solving large systems of linear equations that do not involve finding the inverse.

Generally these approaches are divided into the following two categories:

• Direct Elimination (not inversion) Methods (LDU decomposition, Gauss elimination, Cholesky)

• Iterative Methods (Gauss-Seidel, Jacobi)

We will look at methods from both categories.
Direct Elimination Methods

Elimination methods factor the matrix \([A]\) into products of triangular and diagonal matrices, i.e., the matrix can be expressed as

\[
[A] = [L] [D] [U]
\]

Where \([L]\) and \([U]\) are lower and upper triangular matrices with all diagonal entries equal to “1”. The matrix \([D]\) is a diagonal matrix.

Variations of this decomposition are obtained if the matrix \([D]\) is associated with either the matrix \([L]\) or the matrix \([U]\), i.e.,

\[
[A] = [L] [U]
\]

where \([L]\) and \([U]\) in this last expression are not necessarily the same as the matrices identified in the previous expression.
In an expanded format

\[
\begin{bmatrix}
  a_{11} & a_{12} & \ldots & a_{1n} \\
  a_{21} & a_{22} & \ldots & a_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{n1} & a_{n2} & \ldots & a_{nn}
\end{bmatrix}
\begin{bmatrix}
  l_{11} & 0 & \ldots & 0 \\
  l_{21} & l_{22} & \ldots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  l_{n1} & l_{n2} & \ldots & l_{nn}
\end{bmatrix}
\begin{bmatrix}
  u_{11} & u_{12} & \ldots & u_{1n} \\
  u_{22} & \ldots & u_{2n} \\
  \vdots & \ddots & \vdots \\
  u_{nn}
\end{bmatrix}
\]

The matrices \([L]\) and \([U]\) in this decomposition are not unique. Differences in the many variations of elimination methods are simply differences in how these two matrices are constructed.

In solving a system of linear equations we can now write

\[
[A] \{x\} = \{b\}
\]

as

\[
[A] \{x\} = [L][U] \{x\} = \{b\}
\]
Lecture 3: Determinants & Matrix Inversion

If we let

$$[U] \{x\} = \{y\}$$

then

$$[L][U] \{x\} = [L] \{y\} = \{b\}$$

which is an easier computation. Solving this last expression for each $y_i$ can be accomplished with the following expression

$$y_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} y_j}{l_{ii}} \quad i = 1, 2, \ldots, n$$

With the vector $\{y\}$ known, the vector of unknowns $\{x\}$ are computed from

$$x_i = \frac{y_i - \sum_{j=i+1}^{n} u_{ij} x_j}{l_{ii}} \quad i = n, \ldots, 1$$

The process for solving for the unknown vector quantities $\{x\}$ can be completed without computing the inverse of $[A]$. 
The Gauss Direct Elimination Method

The Gauss elimination method begins with a forward elimination process and finds unknowns by backward substitution.

Here the decomposition of the $n \times n$ matrix $[A]$

$$[A] = [L][U]$$

is accomplished as follows. For each value of $i$ (where $i$ ranges from 1 to $n$) compute

$$u_{ji} = \frac{l_{ij}}{l_{jj}} \quad j = 1, \ldots, i-1$$

$$u_{ii} = 1$$

and

$$l_{ji} = a_{ij} - \sum_{k=1}^{i-1} u_{ki} l_{jk} \quad j = 1, \ldots, n$$

If at any stage in the elimination process the coefficient of the first equation, i.e., $a_{jj}$ (often referred to as the pivot point) or $l_{jj}$ becomes zero the method fails.
Lecture 3: Determinants & Matrix Inversion

In Class Example
Cholesky’s Decomposition – A Direct Elimination Method

In linear algebra, the Cholesky decomposition or Cholesky triangle is a decomposition of a Hermitian, positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose. It was discovered by André-Louis Cholesky for real matrices (as opposed to matrices with elements that are complex). When it is applicable, the Cholesky decomposition is roughly twice as efficient as the LU decomposition for solving systems of linear equations.

The distinguishing feature of the Cholesky decomposition is that the matrix \([A]\), which is symmetrical and positive-definite, can be decomposed into upper and lower triangular matrices that are the transpose of each other, i.e.,

\[
[A] = [L][L]^T
\]

This can be loosely thought of as the matrix equivalent of taking the square root. Note that \([A]\) is a positive definite matrix if for all non-zero vectors \(\{z\}\) the inner product

\[
\{z\}^T [A]\{z\} > 0
\]

is always greater than zero. This is guaranteed if all the eigenvalues of the matrix are positive.
Once the decomposition has been performed, the solution of the system of equations proceeds by forward and backwards substitution in the same manner as the Gauss elimination method.

Convenience recurrence relationships for the Cholesky decomposition are as follows for each successive column (ith index)

\[
\begin{align*}
    l_{ii} &= \sqrt{a_{ii} - \sum_{k=1}^{i-1} (l_{ik})^2} \\
    l_{ji} &= \frac{a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik}}{l_{ii}} \\
    & \quad \quad j = i + 1, \ldots, n
\end{align*}
\]

These expressions can be modified where the expressions are free of needing to take a square root if the previous matrix expression is factored such that

\[
[A] = [L][D][L]^T
\]

where again [D] is a diagonal matrix.
Recurrence relationships for the Cholesky LDL decomposition. They are expressed as follows for each successive column (ith index)

\[ d_{ii} = a_{ii} - \sum_{k=1}^{i-1} d_{kk} (l_{ik})^2 \]

\[ l_{ii} = 1 \]

\[ l_{ji} = \frac{a_{ji} - \sum_{k=1}^{i-1} d_{kk} l_{jk} l_{ik}}{d_{ii}} \quad j = i+1, \ldots, n \]

With \([A]\) decomposed into a triple matrix product the solution to the system of equations proceeds with

\[ \{b\} = [A] \{x\} \]

\[ = [L][D][L]^T \{x\} \]

\[ = [L] \{y\} \]
Again

\[ y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_j \]

\[ l_{ii} \]

\[ i = 1, 2, \ldots, n \]

but now (verify for homework)

\[ x_i = \frac{y_i - \sum_{j=i+1}^{n} d_{ik} l_{kj} x_j}{l_{ii}} \]

\[ i = n, \ldots, 1 \]
Lecture 3: Determinants & Matrix Inversion

In Class Example
Gauss-Seidel: An Iterative Method

For large systems of equations the explicit and implicit elimination methods (backwards and forwards) can be inadequate from a memory storage perspective, execution time and/or round-off error issues. Iterative solution strategies are used when these problems arise. Solutions found using iterative methods can be obtained with a foreknowledge of the error tolerance associated with the method.

The most common iterative scheme is the Gauss-Seidel method. Here an initial estimate is made for the solution vector \( \{x\} \). Subsequent to this guess each equation in the system of linear equations is used to solve for (update) one of the unknowns from a previous calculation. For a linear system of three equations in three unknowns the updating procedure can be expressed by the following expressions

\[
\begin{align*}
-mx_1 &= \frac{b_1 - (a_{12})(m-1)x_2 - (a_{13})(m-1)x_3}{a_{11}} \\
mx_2 &= \frac{b_2 - (a_{21})(m-1)x_1 - (a_{23})(m-1)x_3}{a_{22}} \\
mx_3 &= \frac{b_3 - (a_{31})(m-1)x_1 - (a_{32})(m-1)x_2}{a_{33}}
\end{align*}
\]

Note that \( m \) and \( (m-1) \) represent the current and previous iteration respectively.
In general the Gauss-Seidel iteration format can be expressed as

\[
\left( \begin{array}{c} \mathbf{m} x_i \\ \end{array} \right) = \frac{1}{a_{ii}} \left\{ b_i - \sum_{k=1}^{i-1} (a_{ik}) (\mathbf{m} x_k) - \sum_{k=i+1}^{n} (a_{ik}) (\mathbf{m}^{-1} x_k) \right\}
\]

The original equations must be of a form where the diagonal elements of \([A]\) contain only nonzero values. For a stable structure, or a component with well-defined boundary conditions this will always be possible.

For convenience a vector of zeroes is often used as an initial guess at the solution \(\{\mathbf{x}\}\). Iterations are repeated until the solution converges to “close enough” to the true solution. Since we very seldom know the true solution a priori (why do the problem if we know the answer) iterations continue until the next “answer” obtained is only fractionally different than the “answer” from the previous iteration. Mathematically this is expressed as

\[
\varepsilon_a = \max_{i=1,...,n} \left| \frac{\left( \mathbf{m} x_i \right) - \left( \mathbf{m}^{-1} x_i \right)}{\left( \mathbf{m} x_i \right)} \right| (100) \quad (\%)
\]

with

\[
\varepsilon_a < \zeta
\]
where

\[ \zeta = 0.5 \times 10^{(2-q)} \] (\%)

and \( q \) is the desired number of accurate significant figures.

Many techniques have been developed in order to improve convergence of the Gauss-Seidel method. The simplest and the most widely used is successive over-relaxation method (\textit{SOR}). In this approach a newly computed value \( m^x_i \) is modified by taking a weighted average of the current and previous values in the following manner:

\[
\left( m^x_i \right)_{\text{updated}} = \beta \left( m^x_i \right) + (1 - \beta) \left( m^{-1}x_i \right)
\]

Here \( \beta \) is the relaxation or weighting factor. For over-relaxation methods \( \beta \) ranges between 1 and 2. The assumption is that using current value of \( m^x_i \) will slowly converge to the exact solution whereas the modification of \( m^x_i \) defined above will speed convergence along.

When \( \beta \) is between 0 and 1 the approach is termed a successive under-relaxation method (\textit{SUR}). Choice of a \( \beta \) value is problem dependent.
Lecture 3: Determinants & Matrix Inversion

In Class Example
Lecture 3: Determinants & Matrix Inversion

The Jacobi Iterative Method

The Jacobi method calculates all new values on the basis of the following iterative expression

\[
\begin{pmatrix} m x_i \end{pmatrix} = \frac{1}{a_{ii}} \left\{ b_i - \sum_{\substack{k=1\atop k \neq i}}^{n} (a_{ik}) (m^{-1} x_k) \right\}
\]

The Gauss-Seidel method typically converges to solutions faster than the Jacobi method, the Jacobi iteration scheme has certain advantages when solution of a system of equations is conducted with parallel processing (multiple computers solving the system at the same time).
The Conjugate Gradient Iterative Method

Conjugate gradient methods are the most popular for solving large systems of linear equations of the form

\[ [A] \{ x \} = \{ b \} \]

Here \( \{ x \} \) is an unknown vector and \( \{ b \} \) is a known vector and \( [A] \) is a known, square, symmetric, positive-definite matrix. Conjugate gradient methods are well suited for sparse matrices. If \( [A] \) is dense the best solution strategy is factoring \( [A] \) and using back substitution.

As noted earlier, a matrix is positive definite if for every nonzero vector \( \{ z \} \)

\[ \{ z \}^T [A] \{ z \} > 0 \]

This may mean little at this point because it is not a very intuitive idea. One has a hard time assembling a mental picture of a positive-definite matrix as opposed to a matrix that is not positive definite. A matrix that is positive definite is defined as a Hermitian matrix and there are a number of advantageous qualities associated with this category of matrices.
If a matrix is a positive definite Hermitian matrix then:

- all eigenvalues of the matrix are positive;
- all the leading principal minors of the matrix are positive; and
- there exists nonsingular square matrix $[B]$ such that

\[
\]

Quadratic forms are introduced next and we will see how the concept of positive definite matrices affects quadratic forms. The following scalar function

\[
f(x) = ax^2 + bx + c
\]

is said to have a quadratic form in scalar Algebra. If we change the formulation slightly such that

\[
f(x) = \frac{1}{2} Ax^2 - bx + 0
\]
Extending this equation format to at least two variables leads to equations of the form

\[
f(x_1, x_2) = \frac{1}{2} \left( a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2 \right) - (b_1x_1 + b_2x_2)
\]

and in general

\[
f(x_i) = \frac{1}{2} \left( \sum_{i \neq j}^{n} a_{ij}x_i x_j + 2 \sum_{i=j}^{n} a_{ij}x_i x_j \right) - \sum_{k=1}^{n} b_k x_k
\]

This last expression suggests that we can rewrite it in a matrix format as

\[
f([x]) = \frac{1}{2} [x]^T [A][x] - [b]^T [x]
\]

Where \([A]\) is a known \(n \times n\) matrix, the vector \([b]\) of known quantities is an \(n \times 1\) vector as is the vector of unknowns \([x]\). Several observations can be made. First, in the last two equations both sides of the equal sign represent scalar quantities. As a quick check on the matrix expression let \([x]\) and \([b]\) represent \(3 \times 1\) vectors, and \([A]\) is a \(3 \times 3\) square matrix. If \([A]\) is symmetric and positive-definite then the quadratic function \(f\) is minimized by the solution of

\[
[A][x] = [b]
\]
Let’s consider an example. If we have the following system of linear equations to solve

\[
\begin{align*}
3x_1 + 2x_2 &= 2 \\
2x_1 + 6x_2 &= -8
\end{align*}
\]

then the solution of this system, the quadratic form of the system, and the minimization of the quadratic form is depicted in the following three graphs.

\[
[A] \{x\} = \{b\}
\]

\[
[A] = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}
\]

\[
\{x\} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\]

\[
\{b\} = \begin{bmatrix} 2 \\ -8 \end{bmatrix}
\]
Lecture 3: Determinants & Matrix Inversion

\[ f({x}) = 3(x_1)^2 + 4x_2x_1 + 6(x_2)^2 - 2x_1 + 8x_2 \]

Contour plot depicting level surfaces of the quadratic function \( f({x}) \)

Surface plot of the quadratic function

\[ f({x}) = \frac{1}{2} {x}^T [A] {x} - {b}^T {x} \]
Lecture 3: Determinants & Matrix Inversion

Taking the partial derivative of a scalar valued function that depends on more than one variable results in partial derivatives with respect to each variable. Returning to the system with two unknown variables

\[ f(x_1, x_2) = \frac{1}{2} \left( a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2 \right) - (b_1x_1 + b_2x_2) \]

then

\[ \frac{\partial[f(x_1, x_2)]}{\partial x_1} = a_{11}x_1 + 2a_{12}x_2 - b_1 \]
\[ \frac{\partial[f(x_1, x_2)]}{\partial x_2} = 2a_{12}x_1 + a_{22}x_2 - b_2 \]

This suggest the following matrix format

\[
\begin{bmatrix}
\frac{\partial[f(x_1, x_2)]}{\partial x_1} \\
\frac{\partial[f(x_1, x_2)]}{\partial x_2}
\end{bmatrix} = \begin{bmatrix} a_{11} & 2a_{12} \\ 2a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}
\]
Lecture 3: Determinants & Matrix Inversion

Now set the last result equal to the zero vector, i.e.,

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{bmatrix} a_{11} & 2a_{12} \\ 2a_{12} & a_{22} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Recall from calculus that taking derivatives and setting the resulting expression(s) equal to zero yields a local minimum or local maximum. We can in general restate this in a matrix format as

$$\{0\} = [A]\{x\}^* - \{b\}$$

or

$$[A]\{x\}^* = \{b\}$$

Thus the minimum/maximum point on the quadratic surface

$$f(\{x\}) = \frac{1}{2}\{x\}^T[A]\{x\} - \{b\}^T\{x\}$$

occurs at $\{x\}^*$ which is coincidently the solution of the system of linear equations we wish to solve.
We now seek a way to obtain the solution vector \( \{x\}^* \) in an iterative fashion. Graphically the iterative solution would start at some arbitrary starting point (could be the origin, but in the figure below the iterative process starts at a point defined by a non-zero initial vector \( 0\{x\} \)).

Before establishing an iterative solution technique we define a residual vector \( i\{r\} \) and an error vector \( i\{e\} \) at the \( i^{th} \) step.
Before establishing an iterative solution technique we define a residual vector $i\{r\}$ and an error vector $i\{e\}$ at the $i$th step. The error vector is expressed as

$$i\{e\} = i\{x\} - \{x\}$$

and this is a vector that indicates how far we are from the solution. The residual vector is defined as

$$i\{r\} = \{b\} - [A]i\{x\}$$

and this residual vector indicates how far the matrix product $[A]i\{x\}$ is from correct vector $\{b\}$. Substitution of the expression for the error vector into the expression for the residual vector leads to the following relationship between the two vectors

$$i\{r\} = \{b\} - [A](i\{e\} + \{x\})$$

$$= \{b\} - [A]i\{e\} - [A]\{x\}$$

$$= \{b\} - [A]i\{e\} - \{b\}$$

$$i\{r\} = -[A]^i\{e\}$$
More importantly, the residual vector should be thought of in terms of the following expression

\[ \{ r \}_i = - \left( [A] \{ x \}_i - \{ b \} \right) = - \left\{ \frac{\partial [f(x, \ldots, x)]}{\partial x_1}, \ldots, \frac{\partial [f(x, \ldots, x)]}{\partial x_n} \right\}{x}_i \]

Thought must be given to this expression. This represents the derivative of the quadratic scalar equation evaluated at the components of the iteration vector at the \( i^{th} \) step. Thus the residual vector will always point in the direction of steepest descent, i.e., towards the solution point. This is extremely important in non-linear problems where iterative solution schemes easily bog down.
Now let’s pause and think about the solution vector \( \{x\} \) in general terms. A vector with \( n \) elements can be written in terms of a linear combination of basis vectors. When \( n \) is equal to 3 (i.e., Cartesian three space) we can express a vector as a linear combination of the three unit vectors \((1,0,0), (0,1,0)\) and \((0,0,1)\). But in general

\[
\{x\} = \sum_{i=1}^{n} i \alpha \{s_i\}
\]

For the Cartesian three space example

\[
\{x\} = \sum_{i=1}^{3} i \alpha \{s_i\} = 1 \alpha \{s_1\} + 2 \alpha \{s_2\} + 3 \alpha \{s_3\}
\]

\[
= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \alpha + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \alpha + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \alpha
\]

At the left we see indicated that the \( \alpha \)'s are scaling factors. This is an important idea to latch onto. We will not treat them as a vector.
Next consider the solution expressed as the following vector sum

\[ \{x\} = 0\{x\} + \sum_{i=1}^{n} \alpha \{s_i\} \]

The fuchsia line represents

\[ \sum_{i=1}^{n} \alpha \{s_i\} \]

i.e., a linear combination of the blue vectors. Each blue vector corresponds to a specific value of \(i\).

The last green vector is the solution and the first red vector represents the initial guess.

\[ x_0 \]
For the first iterative step in the last figure

\[ ^1\{x\} = ^0\{x\} + ^1\alpha \{s_i\} \]

This expression can be generalized into the following iterative form

\[ ^i\{x\} = ^{i-1}\{x\} + ^i\alpha \{s_i\} \]

The iteration can be continued until a solution is obtained that is “close enough.” The issues boil down to finding the \( \alpha \)'s, i.e., the scaling factors as well as the basis vectors, i.e., the \( s_i \)'s. Let’s see if we can use the concept of the direction of steepest descent to define the basis vectors and the scale factor at each step.
Return to the example and use the starting vector $^0\{x\} = (-2, -2)$. Our first step should be in the direction of steepest descent towards the minimum. That direction is the solid black line in the figure below.

The next issue is how far along the path of steepest descent should we travel? If we define the step as

$$
^1\{x\} = ^0\{x\} + ^0\alpha^0\{r\}
$$

or in general

$$
i+1\{x\} = ^i\{x\} + ^i\alpha^i\{r\}
$$

then how big should the step be? A line search is a procedure that chooses the $\alpha$ to minimize $f$ along a direction.

Also note that the residual vector is now being used as a basis vector in the expressions above.
Lecture 3: Determinants & Matrix Inversion

The following figure illustrates what we are trying to do. The process is restricted to selecting a point along the surface of the function $f$ and the plane that defines the direction of steepest descent from the point (-2, -2).

As we traverse along the line of steepest descent we can plot the magnitude and the direction of the gradient to the surface at any point on the line. We stop searching along the line when the gradient (derivative) vector is orthogonal to the line of steepest descent. The scaling factor $\alpha$ is now defined.

The new (red) search direction is an updated direction of deepest descent and will be perpendicular to the old (blue) direction.
From the geometry of the previous figure we can see that residual vector \( ^0r \) and the updated residual vector \( ^1r \) are orthogonal. Thus the numerical value for \( \alpha \) can be obtained from the following dot product taken between these two vectors

\[
0 = \begin{bmatrix} ^1r \end{bmatrix}^T \begin{bmatrix} ^0r \end{bmatrix} = \begin{bmatrix} b \end{bmatrix} - \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} ^1x \end{bmatrix}^T \begin{bmatrix} ^0r \end{bmatrix} = \begin{bmatrix} b \end{bmatrix} - \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} ^0x \end{bmatrix} + \begin{bmatrix} ^0\alpha \end{bmatrix} \begin{bmatrix} ^0r \end{bmatrix}]^T \begin{bmatrix} ^0r \end{bmatrix} = \begin{bmatrix} b \end{bmatrix} - \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} ^0x \end{bmatrix}^T \begin{bmatrix} ^0r \end{bmatrix} - \begin{bmatrix} ^0\alpha \end{bmatrix} \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} ^0r \end{bmatrix}]^T \begin{bmatrix} ^0r \end{bmatrix}
\]

or

\[
^0\alpha = \frac{\begin{bmatrix} ^0\alpha \end{bmatrix}^T \begin{bmatrix} ^0r \end{bmatrix}}{\begin{bmatrix} ^0r \end{bmatrix}^T \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} ^0r \end{bmatrix}}
\]
The iterations in their most basic format are as follows

\[
i \{ r \} = \{ b \} - [A] \{ x \}
\]

\[
i \alpha = \frac{i \{ r \}^T i \{ r \}}{i \{ r \}^T [A] i \{ r \}}
\]

\[
i+1 \{ x \} = i \{ x \} + i \alpha i \{ r \}
\]