Introduction to Computational Mathematics (AMATH 242/CS 371)

University of Waterloo
Winter 2019

Linear Equations

One of the problems encountered most frequently in scientific computation is the solution of systems of simultaneous linear equations. In this lecture, the solution of linear systems by Gaussian elimination and the sensitivity of the solution to errors in the data and round-off errors in the computation is covered.

Linear Equations

With matrix notation, a system of simultaneous linear equations is written as $A\vec{x} = \vec{b}$. In general, A is a given square matrix of order n, $\vec{b}$ is a given column vector of n components, and $\vec{x}$ is an unknown column vector of n components.
In solving this system of linear equations we wish to find an **accurate solution** with an **efficient algorithm.**

Question: Is the given linear system problem well-posed i.e. can we find a single unique solution $\vec{x}$ that satisfies the linear system?

Theorem: Existence and Uniqueness

Consider $A\vec{x} = \vec{b}$.

- Case 1: $det(A) \neq 0$ (A has linearly independent rows/columns, or A is invertible) if and only if $\vec{x} = A^{-1}\vec{b}$ is the unique solution of $A\vec{x} = \vec{b}$.
- Case 2: $det(A) = 0$ (recall that range(A)=column space of A), then
  - Case 2a: If $b \in range(A)$ then $A\vec{x} = \vec{b}$ has infinitely many solutions.
  - Case 2b: If $b \notin range(A)$ then $A\vec{x} = \vec{b}$ has no solutions.

Note: Column space (range) of A is the span (set of all possible linear combinations) of its column vectors. If $A \in \mathbb{R}^{n\times n}$, then we can write it as $A\,[\vec{v_1}\ \vec{v_2}\ \cdots\ \vec{v_n}]$, $\vec{v_1}\ \vec{v_2}\ \cdots\ \vec{v_n} \in \mathbb{R}^n$. We define
$C(A) = span(\vec{v_1}, \vec{v_2}, \cdots, \vec{v_n}) = \{c_1\vec{v_1} + c_2\vec{v_2} + \cdots + c_n\vec{v_n} | c_i \in \mathbb{R}\}$

Therefore if $\vec{a} \in C(A)$, then we can write it as $\vec{a} = c_1\vec{v_1} + c_2\vec{v_2} + \cdots + c_n\vec{v_n}$ solution to $A\vec{x} = \vec{b}$ can be written as $x = A^{-1}b$, where A^{-1} is the inverse of A. However, in the computational problems, it is unnecessary to actually compute A^{-1}. As an illustrative example, consider a system consisting of just one equation, such as $7x = 21$. Use of the matrix inverse would lead to $x = 7^{-1} \times 21 = 0.142857 \times 21 = 2.99997$. The inverse requires more arithmetic, a division and a multiplication instead of just a division, and produces a less accurate answer.

Gaussian Elimination is used to solve a linear system.
Definition: $A \in \mathbb{R}^{n \times n}$ with components a_{ij} is said to be **upper-triangular** if $a_{ij} = 0$ for all $i > j$. Similarly, A is said to be **lower-triangular** if $a_{ij} = 0$ for all $i < j$. A is triangular if it is either upper-triangular or lower-triangular.
Gaussian elimination may be performed in two phases:

- Phase 1: Reduce the matrix A to upper triangular form.
- Phase 2: Solve the reduced system by backward substitution.

To illustrate the general linear equation solution algorithm, consider an example of order three:

Example:

- Solve the following linear system.

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \\ 6 \end{pmatrix}$$

To eliminate x_1 from equations 1 and 2:

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \xrightarrow[R_2+0.3R_1 \to R_2]{R_3-0.5R_1 \to R_3} A^{(1)} \begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 2.5 & 5 \end{pmatrix}$$

Then, the system is defined as

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 2.5 & 5 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.1 \\ 2.5 \end{pmatrix}$$

To eliminate x_2 from the third equation we should add $25R_2$ to R_3. This results in

$$\overset{A^{(2)}}{\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.1 & 6 \\ 0 & 0 & 155 \end{pmatrix}} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.1 \\ 155 \end{pmatrix}$$

Then solution is $\vec{x} = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$.

Each algebraic operation that gives $A^{(2)}$ from the original matrix A can be represented by a matrix called M such that $(M^{(2)} \cdot M^{(1)}) \cdot A = A^{(2)}$. In our example

$$M^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 0.3 & 1 & 0 \\ -0.5 & 0 & 1 \end{pmatrix} \Rightarrow M^{(1)} \cdot A = A^{(1)}$$

and

$$M^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 25 & 1 \end{pmatrix} \Rightarrow M^{(2)} \cdot A^{(1)} = A^{(2)}$$

We may write
$(M^{(2)}) \cdot (M^{(1)} \cdot A) = U \Rightarrow A = (M^{(2)} \cdot M^{(1)})^{-1} \cdot A = M^{(1)^{-1}} M^{(2)^{-1}} U = LU$
in which $L = M^{(1)^{-1}} M^{(2)^{-1}}$.
We define the matrix L_j as the inverse of the matrix $M^{(j)}$ (so $L_j M^{(j)} = I$).
Properties of matrices $M^{(j)}$ and L_j

- **Inversion Property:** L_j can be obtained from $M^{(j)}$ by swapping the signs of the off- diagonal elements.
- **Combination Property:** In general, $L = \prod_{j=1}^{n-1} L_j = L_1 \cdot L_2 \cdots L_{n-1}$. L can be obtained from the L_j by placing all of the off-diagonal elements of the matrices L_j in the corresponding position in L.

We note that the matrix L is a special type of lower-triangular matrix which is called **lower triangular matrix with unit diagonal**.
we may write the LU decomposition of A as
$M^{(2)} \cdot M^{(1)} \cdot A = U \Rightarrow A = M^{(1)^{-1}} M^{(2)^{-1}} U = LU$ with L unit lower triangular and U upper triangular. The technique discussed here may be generalized to square matrices of any size and is more generally known as **LU decomposition**.

$$A = M^{(1)^{-1}} \cdot M^{(2)^{-1}} \cdots M^{(n-1)^{-1}} \cdot U$$

in which $M^{(j)}$ is defined in general as

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & C_{ij} & \ddots & \vdots \\ 0 & & & 1 \end{pmatrix} \longrightarrow C_{ij} = -\frac{a_{ij}^{(j-1)}}{a_{jj}^{(j-1)}}$$

Since we defined $L_j = M^{(j)(-1)}$, we also have

$$A = L_1 \cdot L_2 \cdots L_{n-1} \cdot U = LU$$

Since L and U are triangular matrices, we may easily compute the solution to a linear system with either matrix L or U.

Procedure: Solving a Linear System by LU Decomposition. Consider a linear system given by $A\vec{x} = \vec{b}$. Since we now have a procedure for computing the LU decomposition of a matrix A, we may write an algorithm for performing Gaussian Elimination computationally.

- Phase 1: Decompose A = LU so we may write the linear system as $LU\vec{x} = \vec{b}$.
- Phase 2: Solve $L\vec{y} = \vec{b}$ for $\vec{y}$ by forward substitution.
- Phase 3: Solve $U\vec{x} = \vec{y}$ for $\vec{x}$ by backward substitution.

Pivoting: We observe that the LU decomposition algorithm breaks down when at some step i the pivot element a_{ii} is equal to zero. However, this problem does not necessarily imply that the system is unsolvable. Consider the following system:

$$\begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

We can not to proceed to solve the problem using the LU decomposition algorithm. However, we will have no problem proceeding if we swap the first and second rows before applying the LU decomposition.

$$\begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

More formally, the operation of swapping rows can be written as multiplication on the left with a permutation matrix P which is $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ in this example.

Then $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}$. Then we have $A\vec{x} = \vec{b} \Leftrightarrow PA\vec{x} = P\vec{b}$.

Definition. $P \in \mathbb{R}^{n \times n}$ is a **permutation matrix** if and only if P is obtained from the unit matrix I_n by swapping any number of rows.

Theorem

For all $A \in \mathbb{R}^{n \times n}$ there exists a permutation matrix P, a unit lower triangular matrix L and an upper triangular matrix U (all of type $\mathbb{R}^{n \times n}$) such that PA = LU.

Corollary. If A is nonsingular then $A\vec{x} = \vec{b}$ can be solved by the LU decomposition algorithm applied to PA.

Computational cost of Gaussian elimination using the LU decomposition:

- Phase 1 (LU decomposition): Addition and subtraction=$[\frac{1}{3}n^3 + O(n^2)]$, Multiplication and division=$[\frac{1}{3}n^3 + O(n^2)]$
- Phase 2(forward substitution): $n^2 + O(n)$
- Phase 3(backward substitution): $n^2 + O(n)$

Big O definition: Suppose that f(x) is a polynomial in x. Then $f(x) = O(x^n)$ as $(x \to 0)$ if $\exists\ C > 0$ and $x_0 > 0$ such that $|f(x)| < C^n|x^n| \forall |x| < x_0$ i.e. $\lim_{x\to 0} \frac{|f(x)|}{|x^n|} = M_1$.

Similarly, Suppose that f(x) is a polynomial in x. Then $f(x) = O(x^n)$ as $(x \to \infty)$ if $\exists\ C > 0$ and $x_0 > 0$ such that $|f(x)| < C^n|x^n| \forall |x| > x_0$ i.e. $\lim_{x\to \infty} \frac{|f(x)|}{|x^n|} = M_2$.

Linear Equations

- For $f(x) = 3x^2 + 7x^3 + 10x^4 + 7x^{12}$ we have
$f(x) = O(x^2),\ f(x) \neq O(x^3),\ f(x) = 3x^2 + O(x^3)$ if $x \to 0$.(why?)
- For $f(x) = 3x^2 + 7x^3 + 10x^4 + 7x^{12}$ we have
$f(x) = O(x^{12}),\ f(x) = 7x^{12} + O(x^4)$ if $x \to \infty$.(why?)

Determinant: Determinant of $A \in \mathbb{R}^{n \times n}$ is given by

$$det(A) = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} det(A_{ij}) \qquad \text{for fixed i}$$

in which the matrix A_{ij} is an $(n-1) \times (n-1)$ matrix obtained by removing row i and column j from the original matrix A. This is the expansion of the determinant about row i for any $1 \leq i \leq n$. We may also consider the expansion of the determinant about column j for any $1 \leq j \leq n$ as follows:

$$det(A) = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} det(A_{ij}) \qquad \text{for fixed i}$$

Proposition: The following identities hold for determinants:

- $det(BC) = det(B) \cdot det(C), \quad (B, C \in \mathbb{R}^{n \times n})$
- $U \in \mathbb{R}^{n \times n}$ upper triangular $\Rightarrow \quad det(U) = \prod_{i=1}^{n} u_{ii}$
- $L \in \mathbb{R}^{n \times n}$ lower triangular $\Rightarrow \quad det(L) = \prod_{i=1}^{n} l_{ii}$
- $P \in \mathbb{R}^{n \times n}$ permutation matrix $\Rightarrow \quad det(P) =$

$$\begin{cases} +1 & \text{even number of row changes to obtain P from } I_n \\ -1 & \text{odd number of row changes to obtain P from } I_n \end{cases}$$

The algorithm for the LU decomposition can only be performed if there are no divisions by zero.

How can we guarantee that this will not occur?

Proposition. Consider a matrix $A \in \mathbb{R}^{n \times n}$. Then $det(A) \neq 0$ if and only if the decomposition $PA = LU$ has $u_{ii} \neq 0 \ \forall i$.

condition and stability:

We consider a particular set of matrix norms, namely those induced by a vector norm. These are also known as the set of "natural" matrix norms over the vector space of matrices (for matrices of the form $A \in \mathbb{R}^{n \times n}$, the set $V = \mathbb{R}^{n \times n}$ is a vector space over $\mathbb{R}$.)

Definition. The natural matrix p-norm for $p = 1, 2$ or ∞ is defined by

$$\| A \|_p = \max_{\|\vec{x}\| \neq 0} \frac{\| A\vec{x} \|_p}{\| \vec{x} \|_p}$$

Propositions: Consider a matrix $A \in \mathbb{R}^{n \times n}$ with elements a_{ij}. Then

- $\| A\vec{x} \|_p \leq \| A \|_p \| \vec{x} \|_p$
- $\| A \|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}|$
- $\| A \|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|$
- $\| A \|_2 = \max_{1 \leq i \leq n} \lambda_i^{1/2}$ where λ_i are the eigenvalues of $A^T A$
- $\| A + B \|_p \leq \| A \|_p + \| B \|_p$

We can now show that the natural matrix norm satisfies the defining properties of a norm:

Proposition: $\| A \|_p$ is a norm:

- $\| A \|_p \geq 0, \quad \| A \|_p = 0 \Leftrightarrow A = 0$
- $\| \alpha A \|_p = |\alpha| \, \| A \|_p$
- $\| A + B \|_p \leq \| A \|_p + \| B \|_p$

Condition of the problem $A\vec{x} = \vec{b}$: The linear system problem can be reformulated as as: Find $\vec{x}$ from $A\vec{x} = \vec{b}$ (i.e., $\vec{x} = A^{-1}\vec{b}$).
From this statement of the problem, we may write $\vec{x} = \vec{f}(A, \vec{b}$. If we want to consider the condition of this problem, there are then two dependent variables which can be perturbed and with contribute to the condition number. We want to consider the change $\Delta\vec{x}$ if we have inputs $A + \Delta A$ and $\vec{b} + \Delta\vec{b}$.

- Consider the following linear system:

$$\begin{pmatrix} 1 & 2 \\ 0.499 & 1.001 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}$$

Solution of this system is $\vec{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. We perturb A by a small matrix ΔA to yield a new linear system and solution given by

$$\begin{pmatrix} 1 & 2 \\ 0.500 & 1.001 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1.5 \end{pmatrix}, \qquad \vec{x} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$

Thus a small change in A results in a large change in $\vec{x}$ this seems to imply that the problem is ill conditioned.

Linear Equations

In order to examine the condition of the initial problem **P** ($A\vec{x} = \vec{b}$) we need to consider a slight perturbation on the input data A and $\vec{b}$,

$$(A + \Delta A)(\vec{x} + \Delta\vec{x}) = \vec{b} + \Delta\vec{b}$$

To find $\kappa(A)$, we consider three cases:

- $\Delta A = 0$ and $\Delta\vec{b} \neq 0$
- $\Delta\vec{b} = 0$ and $\Delta A \neq 0$
- $\Delta\vec{b} \neq 0$ and $\Delta A \neq 0$

In all cases we get the condition number as $\kappa(A) = \| A \| \| A^{-1} \|$.
From this result, it appears that the condition number of a matrix $\kappa(A)$ is all we need to determine the condition number of the problem **P** defined by the linear system $A\vec{x} = \vec{b}$. In particular, we note that the 2-condition number of a matrix (defined by using the 2-norm) has a useful property that makes it unnecessary to compute the inverse A^{-1}:
Proposition: For a matrix $A \in \mathbb{R}^{n\times n}$

$$\kappa_2(A) = \| A \|_2 \| A^{-1} \|_2 = \sqrt{\frac{\lambda_{\max}(A^T)}{\lambda_{\min}(A^T A)}}$$

Proposition: For a matrix $A \in \mathbb{R}^{n \times n}$

$$\kappa_2(A) = \| A \|_2 \| A^{-1} \|_2 = \sqrt{\frac{\lambda_{\max}(A^T)}{\lambda_{\min}(A^T A)}}$$

- Compute the condition number of $\begin{pmatrix} 1 & 2 \\ 0.500 & 1.001 \end{pmatrix}$.

Stability of the LU Decomposition Algorithm.

Problem. Consider the mathematical problem defined by $z(x) = \frac{a}{x}$ with a constant. We wish to know the absolute and relative condition number of this problem. The absolute condition number is obtained as $\kappa_A = \frac{|a|}{x^2}$. Thus, if x is small then this problem is ill-conditioned with respect to the absolute error. The relative condition number is computed as $\kappa_R = 1$ so the problem is well-conditioned with respect to the relative error. These results indicate that dividing by a small number (or multiplying by a large number) is ill-conditioned.

We conclude that the problem of linear systems will be ill-conditioned if we have many divisions by small numbers. Consider the following example, with δ small.

$$\begin{pmatrix} \delta & 1 \\ 1 & 1 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Applying Gaussian elimination and solving the resulted equations for $\vec{x}$ yields $x_1 \approx 1$ and $x_2 \approx 1$.
Under the finite precision system $F[b = 10, m = 4, e = 5]$ with$\delta = 10^{-5}$ we have by backward substitution $\hat{x}_1 = 0$ and $\hat{x}_2 = 1$ and so have generated a large error in $\hat{x}_1$.
To solve the issue, we first interchange the two equations (and so use pivot 1 instead of δ). After applying Gaussian elimination $\hat{x}_1 = 1$ and $\hat{x}_2 = 1$ and so the large error is avoided.

In general, in order to minimize the error, we should rearrange the rows in every step of the algorithm so that we get the largest possible pivot element (in absolute value). This will give us the most stable algorithm for computing the solution to the linear system since we avoid divisions by small pivot elements. This approach is called LU decomposition with partial pivoting.

Iterative Methods for solving $A\vec{x} = \vec{b}$:

Definition: $A \in \mathbb{R}^{n \times n}$ is a sparse matrix if and only if the number of nonzero elements in A is much smaller than n^2. We can often store matrices of this type very efficiently for example in compressed sparse row (CSR) format or diagonal format.

Definition: $A \in \mathbb{R}^{n \times n}$ is strictly diagonally dominant if and only if $|a_{ii}| > \sum_{j=1, j \neq i}^{n} |a_{ij}|$ for all rows of i.

Proposition: A strictly diagonally dominant matrix is non-singular.

In general if A is strictly diagonally dominant, the transpose of A does not necessarily retain the same property.

Jacobi and Gauss-Seidel Methods. To solve the linear system $A\vec{x} = \vec{b}$ using an iterative algorithm, we write at each step $\vec{x}^{\text{old}} = \begin{pmatrix} x_1^{\text{old}} \\ x_2^{\text{old}} \\ x_3^{\text{old}} \end{pmatrix}$ (assuming our system is of order 3), and determine x^{new} from x^{old} by either the Jacobi or Gauss-Seidel algorithm.

Jacobi: We assume $A \in \mathbb{R}^{3\times 3}$ and construct a system of equations as follows:

$$\begin{aligned} a_{11}x_1^{\text{new}} + a_{12}x_2^{\text{old}} + a_{13}x_3^{\text{old}} &= b_1 \\ a_{21}x_1^{\text{old}} + a_{22}x_2^{\text{new}} + a_{23}x_3^{\text{old}} &= b_2 \\ a_{13}x_1^{\text{old}} + a_{32}x_2^{\text{old}} + a_{33}x_3^{\text{new}} &= b_3 \end{aligned}$$

Therefore,

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{\text{old}} \right)$$

Gauss-Seidel: in this method we use the elements of x^{new} derived earlier in the same step and construct a linear system as:

$$\begin{aligned} a_{11}x_1^{\text{new}} + a_{12}x_2^{\text{old}} + a_{13}x_3^{\text{old}} &= b_1 \\ a_{21}x_1^{\text{new}} + a_{22}x_2^{\text{new}} + a_{23}x_3^{\text{old}} &= b_2 \\ a_{13}x_1^{\text{new}} + a_{32}x_2^{\text{new}} + a_{33}x_3^{\text{new}} &= b_3 \end{aligned}$$

Rearranging yields the defining equation for the Gauss-Seidel method:

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{\text{new}} - \sum_{j=i+1}^{n} a_{ij} x_j^{\text{old}} \right)$$

For both of these methods, we must choose a starting vector $\vec{x}^{(0)}$ and generate the sequence $\vec{x}^{(1)}, \vec{x}^{(2)}, \ldots = \left\{ \vec{x}^{(i)} \right\}_{i=0}^{\infty}$

Using the decomposition $A = A_L + A_D + A_R$, in which A_L is an lower triangular matrix with zero diagonals, A_D is a diagonal matrix and A_R is an upper triangular matrix with zero diagonals, we formulate these methods in matrix form:

Jacobi: $\vec{x}^{\text{new}} = A_D^{-1}\left(\vec{b} - (A_L + A_R)\vec{x}^{\text{old}}\right)$

Gauss-Seidel: $\vec{x}^{\text{new}} = A_D^{-1}\left(\vec{b} - (A_L\vec{x}^{\text{new}} + A_R\vec{x}^{\text{new}})\right)$

Can we guarantee convergence of the iterative methods? For some classes of matrices the answer is yes. It depends on the matrix A whether the sequence of iterates converges and if so, how quickly it does. We can prove the following sufficient condition for convergence:

Theorem. Consider $A\vec{x} = \vec{b}$ and let $\vec{x}^{(0)}$ be any starting vector. Let $\left\{\vec{x}^{(i)}\right\}_{i=0}^{\infty}$ be the sequence generated by either Jacobi or Gauss-Seidel iterative methods. Then if A is strictly diagonally dominant the sequence converges to the unique solution of the system $A\vec{x} = \vec{b}$.

Convergence of iterative methods: To determine when to stop iteration for a given iterative method, we need a measure of how close we are to a correct solution.

Definition. The residual of a linear system $A\vec{x} = \vec{b}$ for some vector $\vec{u}$ is $\vec{r} = \vec{b} - A\vec{u}$. We write the residual at step i as $\vec{r}^{(i)}$. As $\vec{r}^{(i)}$ becomes smaller, we approach convergence of the system. Hence, for a given relative tolerance $t = 10^{-6}$ (for example), we compute the residual at each step and stop when $\frac{\| \vec{r}^{(i)} \|_2}{\| \vec{r}^{(0)} \|_2} \leq t$.

Linear Equations

For an iterative approximation $\vec{u}$, the error is given by $\vec{e} = \vec{x} - \vec{u}$. This leads to $A\vec{e} = \vec{r}$. Using the residual instead of error is beneficial and we can write $\vec{x} = \vec{u} + A^{-1}\vec{r}$. Unfortunately, inverting A is an expensive operation. Instead of using A^{-1}, we can choose a matrix B that is easy to invert such that B^{-1} is an approximation for A^{-1}. Then, $\vec{x} \approx \vec{u} + B^{-1}\vec{r}$. Repeated application of this formula leads to a standard general form for an iterative method:

$$\vec{x}^{(i+1)} = \vec{x}^{(i)} + B^{-1}\vec{r}^{(i)}$$

which is equal to $\vec{x}^{(i+1)} = \vec{x}^{i} + B^{-1}(\vec{b} - A\vec{x}^{i})$.
If B is chosen appropriately then $\vec{x}^{i}$ will converge to $\vec{x}$.

In general, we note that often Gauss-Seidel will converge faster than Jacobi, but this is not always the case. For sparse matrices, we often obtain $W = O(n^2)$ for both methods.

Theorem (Convergence of Iterative Methods).
Consider the iterative method $\vec{x}^{(i+1)} = \vec{x}^{i} + B^{-1}(\vec{b} - A\vec{x}^{i})$ with $det(A) \neq 0$. If there exists a p-norm for which $\| I - B^{-1}A \|_p < 1$ then the iterative method will converge for any starting value $\vec{x}^{(0)}$. Convergence will then hold in any p-norm.

Since this theorem is a general form for iterative methods, we should be able to determine B for the Jacobi and Gauss-Seidel methods:

Jacobi: $B^{-1} = A_D^{-1}$.

Gauss-Seidel: $B^{-1} = (A_D + A_L)^{-1}$