1 November 10th, 2020

1.1 Iterative Methods to Solve Linear Systems

Recall that a linear system is of the form:

$$Ax = b, \quad A = (a_{ij}), \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}.$$

Where A is a $n \times n$ matrix. The standard approach is **Gaussian Elimination**. Recall that this is done by applying row operations to:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} & b_1 \\ a_{21} & a_{22} & & & \vdots \\ \vdots & & \ddots & & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} & b_N \end{bmatrix}.$$

into an upper triangular matrix, and then doing back-substitution. This can be done for any matrix, even non-square matrices. However, this method is slow, as it has a computational cost of $O(N^2)$, making it not practice for many calculations.

Example 1.1

Let us assume we are solving the laplacian on unit cube $[0,1]^3$:

$$\begin{cases} \Delta u = f = u_{xx} + u_{yy} + u_{zz} \\ u\big|_{\partial\Omega} = 0 \end{cases}$$

with 100 points in each dimension. As such, $N = 100^3$, meaning that the cost on the order 10^{12}.

As such, we need to develop methods to solving linear system that are more efficient. One such type of methods are iterative methods. Let us assume we want to solve the equation:

$$f(x) = x^5 + x^4 + 3x^2 + 1 = 0.$$

To solve for the roots of this equation, we can use the **Newton's Method**, for which we have an initial guess x_0. If $f(x_0) = y_0 = 0$, we are done. Otherwise, we use the Taylor expansion at the point by considering the tangent line at (x_0, y_0), and taking it's x-intercept, with:

$$y = f(x_0) + f'(x_0)(x - x_0) = 0 \implies x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

We repeat this process with:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

If $x_n \to x^*$, then:

$$x^* = x^* - \frac{f(x^{.*})}{f'(x^*)} \implies f(x^*) = 0.$$

meaning it converges to a root of $f(x)$.

Remark 1.2 — For the above case, we assume $f'(x^*) \neq 0$, this is a reasonable assumption. We can also slightly change this method to account for this situation.

Remark 1.3 — Newton's method depends on the initial guess x_0. In addition, it only converges (if it does) to one root, where as $f(x)$ above has 5 roots.

Remark 1.4 — Iterative methods generally have the following approach:

- We have a problem that is difficult to solve or too expensive
- We attempt to solve an approximate problem that is easy (e.g. tangent line for Newton's method)
- We repeatedly solve this simpler problem

Let us consider:

$$Ax = b.$$

where:

$$A = \begin{bmatrix} d_1 & & * \\ & \ddots & \\ * & & d_n \end{bmatrix} = \begin{bmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{bmatrix} - \begin{bmatrix} 0 & & 0 \\ & \ddots & \\ -* & & 0 \end{bmatrix} - \begin{bmatrix} 0 & & -* \\ & \ddots & \\ 0 & & 0 \end{bmatrix} = D - L - U.$$

Remark 1.5 — D is a diagonal matrix, L is a lower triangular matrix, and U is an upper triangular matrix

As such, we have:

$$\begin{aligned} Ax = Dx - Lx - Ux = b \\ \implies Dx =& b + Lx + Ux \end{aligned}$$

Given an initial guess x_0, we have:

.

$$\begin{aligned} Dx &= b + Lx_0 + Ux_0 \\ x_1 &= D^{-1}(b + Lx_0 + Ux_0) \\ \implies x_{n+1} &= D^{-1}(b + Lx_n + Ux_n). \end{aligned}$$

This method is called the **Jacobi iteration**, and it has $O(N)$ computation cost, since we only have compute the diagonal for D^{-1} and back-substitution.

Remark 1.6 — Note that D is a diagonal matrix, meaning that D^{-1} is very easy to solve:

$$D^{-1} = \begin{bmatrix} \frac{1}{d_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{d_n} \end{bmatrix}.$$

If $x_n \to x_*$, we have:

$$\begin{aligned} Dx_* &= D^{-1}(b + Lx_* + Ux_*) \\ \implies Ax_* &= Dx_* - Lx_* - Ux_* = b. \end{aligned}$$

meaning that x_* is a solution to $Ax = b$.

However, we still have to investigate its convergence, since if it requires $O(N)$ iterations to converge, then it is no better than Gaussian Elimination.

If A is upper triangular, then we can solve the problem using back-substitution in $O(N)$, meaning we only need $A = D - U$. As such, we have:

$$\begin{aligned} Ax = Dx - Lx - Ux &= b \\ Dx - Ux &= b + L_x \\ (D-U)x &= b + Lx. \end{aligned}$$

Given an initial guess x_0, we have:

$$\begin{aligned} (D-U)x_1 &= b + Lx_0 \\ \implies x_1 &= (D-U)^{-1}(b + Lx_0) \\ \implies x_{n+1} &= (D-U)^{-1}(d + Lx_n). \end{aligned}$$

This method is called the **Gauss-Seidel Iteration**, and also has $O(N)$ cost per iteration.

Remark 1.7 — Note that we can solve $(D-U)^{-1}$ efficiently.

Remark 1.8 — There is a different version where we put U on the right hand side instead of L, giving us:

$$x_{n+1} = (D-L)^{-1}(b + Ux_n).$$

There is another version called **SOR iteration** where instead of having $D - L$, we have $D - \omega L$ for $0 < \omega < 2$. This gives u:

$$\begin{aligned} Dx_{n+1} &= (1-\omega)Dx_n + \omega(b + Lx_{n+1} + Ux_n) \\ \implies (D - \omega L)x_{n+1} &= (1-\omega)Dx_n + \omega(b + Ux_n) \\ \implies x_{n+1} &= (D-\omega L)^{-1}\left[(1-\omega)Dx_n + \omega(b + Ux_n)\right]. \end{aligned}$$

This converges to a solution, as:

$$\begin{aligned} x_* &= (D-\omega L)^{-1}\left[(1-\omega)Dx_* + \omega(b + Ux_*)\right] \\ Dx_* - \omega L_* x_* &= (D - \omega D)x_* + \omega b + \omega U x_* \\ \implies \omega(D - L - U)x_* &= \omega b \\ \implies Ax_* &= b. \end{aligned}$$

Remark 1.9 — If $\omega = 1$, this would reduce to Gauss-Seidel.

We need to study two issues:

1. convergence, to see if it will return a solution
2. rate of convergence, as if it requires more than $O(N)$ iterations, then it is worse than Gaussian elimination.

Note that these iterative methods all have the following form:

$$x_{n+1} = Gx_n + C.$$

Example 1.10

For the above methods, we have:

1. Jacobi Iteration: $G = D^{-1}(L+U)$, $C = D^{-1}b$
2. Gauss-Seidel: $G = (D-L)^{-1}U$ or $G = (D-U)^{-1}L$
3. SOR: $G = (D-\omega L)^{-1}((1-\omega)D + \omega U)$

We know that the exact solution satisfies:

$$x^* = Gx^* + C.$$

and the iterative solution satisfies:

$$x_{n+1} = Gx_n + C.$$

Subtracting, we have:

$$x_{n+1} - x^* = G(x_n - x^*).$$

Denoting the difference as the error $e_n = x_n - x^*$, we have:

$$e_{n+1} = Ge_n = G^2 e_{n+1} = \ldots = G^{n+1} e_0.$$

Where $e_0 = x_0 - x^*$. As such, we want to see if $x_{n+1} = G^{n+1} e_0$ tends to zero.

Let us consider the simple case where G is diagonalizable, i.d. $G = T^{-1}\Lambda T$, where $\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix}$, we have:

$$G^{n+1} = T^{-1}\Lambda^{n+1}T, \quad \Lambda^{n+1} = \begin{bmatrix} \lambda_1^{n+1} & & \\ & \ddots & \\ & & \lambda_m^{n+1} \end{bmatrix} \to 0.$$

Lemma 1.11

The iteration converges as $n \to +\infty$ for all starting vector $x_0 \iff \rho(G) < 1$, where:

$$\rho(G) = \max_i |\lambda_i(G)|,$$

or in other words the norm of all eigenvalues $|\lambda_i| < 1$.

Definition 1.12. $\rho(G)$ is the **spectral radius** of G.

Remark 1.13 — For the diagonalization of G, Λ is a diagonal matrix of all of the eigenvalues, and T is all of the eigenvectors.

Returning to the three iterations, let's study the convergence using Lemma 1.11.

Theorem 1.14

The Jacobi iteration converges for an irreducibly diagonally dominant matrix.

Definition 1.15. $A = (A_{ij})$ is **diagonally dominant** if:

$$|A_{\ell\ell}| \geq \sum_{m \neq \ell} |A_{\ell m}|, \qquad \forall \ell.$$

or in other words, the norm of the diagonal entry is larger than the sum of the off diagonal entries in the row.

Definition 1.16. $A = (A_{ij})$ is **strictly diagonally dominant** if:

$$|A_{\ell\ell}| > \sum_{m \neq \ell} |A_{\ell m}|, \qquad \forall \ell.$$

Definition 1.17. $A = (A_{ij})$ is **irreducibly diagonally dominant** if A is diagonally dominant and the strict inequality holds for at least one row ℓ.

Example 1.18

Consider the identity matrix I. Note that it is strictly diagonally dominant for all rows. Thus it is irreducibly diagonally dominant.

Example 1.19

Consider the central difference matrix for $-u'' = f$:

$$A = -\frac{1}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & \end{bmatrix}.$$

Note that $|A_{\ell\ell}| = 2$, where as $\sum_{m \neq \ell} |A_{\ell m}| = 2$. for all $\ell \in \{2, \ldots, n-1\}$ However for $\ell = 1$ or n, we would the strict inequality. Thus it is irreducibly diagonally dominant.