• Compare three methods, Jacobi, Gauss-Seidel and Successive over relaxation (SOR).

• Conjugate gradient method.
We would like to compare the convergence of Jacobi, Gauss-Seidel, and SOR

- First, let us realize the Jacobi and the Gauss-Seidel method can be caseted into a general matrix form. Continue taking the notation of Poisson eq.

- In a matrix form, as before, Jacobi method reads

\[
\phi_i^{(k)} = - \sum_{j=1, j \neq i}^n (a_{ij} / a_{ii}) \phi_j^{(k-1)} + (b_i / a_{ii}),
\]

while the Gauss-Seidel method reads

\[
\phi_i^{(k)} = - \sum_{j=1}^{i-1} (a_{ij} / a_{ii}) \phi_j^{(k)} - \sum_{j=i+1}^n (a_{ij} / a_{ii}) \phi_j^{(k-1)} + (b_i / a_{ii}),
\]

- Then, with a parameter \(0 < \omega < 2\), Successive over relaxation (SOR) reads

\[
\phi_i^{(k)} = \omega \left[ - \sum_{j=1}^{i-1} (a_{ij} / a_{ii}) \phi_j^{(k)} - \sum_{j=i+1}^n (a_{ij} / a_{ii}) \phi_j^{(k-1)} + (b_i / a_{ii}) \right] + (1 - \omega) \phi_i^{(k-1)}
\]
A pseudo code for Gauss-Seidel method is given

- In Gauss-Seidel, if $x^{(k-1)}$ is not saved, then we can dispense with the superscripts in the pseudocode as follows:

  ```java
  for k = 1 to k_{max} do
    for i = 1 to n do
      x_i ← (b_i − \sum_{j=1,j\neq i}^{n} a_{ij} x_j) / a_{ii}
    end for
  end for
  ```

- Note that the convergence criteria (e.g. L2-norm) can be included into iteration process process.
• An example of Jacobi method.

Let us exercise “\(x^{(k+1)} = (I - Q^{-1}A) \cdot x^{(k)} + Q^{-1}b\).” Take,

\[
A = \begin{pmatrix}
2 & -1 & 0 \\
-1 & 3 & -1 \\
0 & -1 & 2
\end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 8 \\ -5 \end{pmatrix}
\]

Choose \(Q\) as the diagonal components of \(A\):

\[
Q = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad Q^{-1} = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/2 \end{pmatrix}, \quad Q^{-1}A = \begin{pmatrix} 1 & -1/2 & 0 \\ -1/3 & 1 & -1/3 \\ 0 & -1/2 & 1 \end{pmatrix}
\]

\[
I - Q^{-1}A = \begin{pmatrix} 0 & 1/2 & 0 \\ 1/3 & 0 & 1/3 \\ 0 & 1/2 & 0 \end{pmatrix}, \quad Q^{-1}b = \begin{pmatrix} 1/2 \\ 8/3 \\ -5/2 \end{pmatrix}
\]

Then

\[
x^{(0)} = (0.0, 0.0, 0.0) \\
x^{(1)} = (0.5, 2.67, -2.5)
\]

\[
\ldots \ldots
\]

\[
x^{(21)} = (2.0, 3.0, -1.0)
\]
• If A is diagonally dominant, then the Jacobi and Gauss-Seidel method converge for any starting vector \( x^{(0)} \).

• Suppose that the matrix A has positive diagonal elements and \( 0 < \omega < 2 \). The SOR method converges for any starting vector \( x^{(0)} \) if and only if A is symmetric and positive definite.\(^a\)

• J=21, GS=9, SOR=7, and Conjugate gradient method takes 3 iterations.

\(^a\)Symmetric positive definite ; if \( A = A^T \) and \( x^T Ax > 0 \) for all nonzero real vectors \( x \).
Conjugate gradient method is based on minimization of quadratic function

- As an introduction: Now imagine a solution for \( f(x) = ax - b = 0 \). We know how to solve of course, \( x = b/a \).

- Yet another way of looking at this problem is → we imagine a quadratic function \( F(x) = (a/2)x^2 - bx + c \) and find the saddle (minimum) point. \(^a\)

\(^a\)The saddle point \( x = b/a \) for \( F'(x) = ax - b = 0 \) is nothing but the solution for \( f(x) = 0 \).
Let us then apply this minimization idea to the matrix equation $Ax = b$

- Correspondingly a quadratic form:

$$F(x) = \frac{1}{2} \langle x, x \rangle_A - \langle b, x \rangle + c$$

where $\langle x, x \rangle_A = x^T A x$ and $\langle b, x \rangle$ is the scalar product.

- The gradient of the quadratic form is

$$F'(x) = \frac{1}{2} A^T x + \frac{1}{2} A x - b$$

If $A$ is a symmetric matrix, this becomes

$$F'(x) = A x - b$$

Thus the solution of $Ax = b$ is the saddle point of $F(x)$. If $A$ is symmetric and positive definite, then the minimum point will give us the solution.
Suppose $p^{(1)}, \ldots, p^{(k)}, \ldots, p^{(n)}$, are conjugate direction vectors. Then the true solution can be expanded as

$$x^* = \alpha_1 p^{(1)} + \ldots + \alpha_k p^{(k)} + \ldots + \alpha_n p^{(n)}$$

whose coefficients are given by

$$\alpha_k = \frac{\langle p^{(k)}, b \rangle}{\langle p^{(k)}, p^{(k)} \rangle_A}.$$ 

Now the concept of iteration enters. The key is to choose a small set of (not $n$) direction vectors $p^{(k)}$. Defining the $k$-th residual

$$r^{(k)} = b - Ax^{(k)}$$

The crude descent method moves in the direction of $r^{(k)}$. In CG, now taking the direction close to $r^{(k)}$ and imposing $p^{(k)}$ to be conjugate each other, we move by

$$x^{(k+1)} = x^{(k)} + \alpha_{k+1} p^{(k+1)} \quad \text{where} \quad p^{(k+1)} = r^{(k)} - \frac{\langle p^{(k)}, r^{(k)} \rangle_A}{\langle p^{(k)}, p^{(k)} \rangle_A} p^{(k)}.$$
Summary of discussions

- Diagonally dominant condition revisited.
- Comparison of three iterative methods.
- Conjugate gradient method (end of linear algebra).
- Will talk about PIC simulation on 4/23, 4/26, and 4/30. Plan to refer to FT anf FFT.