6 Optimization Based Methods

Unconstrained optimization is another tool employed to solve large scale linear algebraic equations. Gradient and conjugate gradient methods for numerically solving unconstrained optimization problems can be tailored to solve a set of linear algebraic equations. The development of the gradient search method for unconstrained optimization for any scalar objective function $\psi(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is presented in Appendix C. In this section, we present how this method can be tailored for solving $Ax = b$.

6.1 Gradient Search Method

Consider system of linear algebraic equations of the form

$$Ax = b \; , \; x, b \in \mathbb{R}^n \; \text{(127)}$$

where $A$ is a non-singular matrix. Defining objective function

$$\phi(x) = \frac{1}{2} (Ax - b)^T (Ax - b) \; \text{(128)}$$

the necessary condition for optimality requires that

$$\frac{\partial \phi(x)}{\partial x} = A^T (Ax - b) = 0 \; \text{(129)}$$

Since $A$ is assumed to be nonsingular, the stationarity condition is satisfied only at the solution of $Ax = b$. The stationary point is also a minimum as $A^T A$ is a positive definite matrix.

Thus, we can compute the solution of $Ax = b$ by minimizing

$$\phi(x) = \frac{1}{2} x^T A^T A x - (A^T b)^T x \; \text{(130)}$$

When $A$ is positive definite, the minimization problem can be formulated as follows

$$\phi(x) = \frac{1}{2} x^T A x - b^T x \; \text{(131)}$$

as it can be shown that the above function achieves a minimum for $x^*$ where $Ax^* = b$. In the development that follows, for the sake of simplifying the notation, it is assumed that $A$ is symmetric and positive definite. When the original problem does not involve a symmetric positive definite matrix, then it can always be transformed by pre-multiplying both sides of the equation by $A^T$.

To arrive at the gradient search algorithm, given a guess solution $x^{(k)}$, consider the line search problem (ref. Appendix C for details)

$$\lambda_k = \min_{\lambda} \phi(x^{(k)} + \lambda g^{(k)})$$

where

$$g^{(k)} = -(Ax^{(k)} - b) \; \text{(132)}$$

Solving the one dimensional optimization problem yields
Thus, the gradient search method can be summarized in Table 5

Table 5: Gradient Search Method for Solving Linear Algebraic Equations

| INITIALIZE: $x^{(0)}, \varepsilon, k_{\text{max}}, \lambda^{(0)}, \delta$ |
| $\lambda = 0$ |
| $g^{(0)} = b - Ax^{(0)}$ |
| WHILE $[(\delta > \varepsilon) \wedge (k < k_{\text{max}})]$ |
| $\lambda_k = \frac{b^T g^{(k)}}{(g^{(k)})^T A g^{(k)}}$ |
| $x^{(k+1)} = x^{(k)} + \lambda_k g^{(k)}$ |
| $g^{(k+1)} = b - Ax^{(k+1)}$ |
| $\delta = \frac{\|g^{(k+1)} - g^{(k)}\|_2}{\|g^{(k+1)}\|_2}$ |
| $g^{(k)} = g^{(k+1)}$ |
| $k = k + 1$ |
| END WHILE |

6.2 Conjugate Gradient Method

The gradient method makes fast progress initially when the guess is away from the optimum. However, this method tends to slow down as iterations progress. It can be shown that directions there exist better descent directions than the negative of the gradient direction. One such approach is conjugate gradient method. In conjugate directions method, we take search directions $\{s^{(k)} : k = 0, 1, 2, \ldots\}$ such that they are orthogonal with respect to matrix $A$, i.e.

$$[s^{(k)}]^T A s^{(k-1)} = 0 \text{ for all } k \quad \text{(133)}$$

Such directions are called $A$-conjugate directions. To see how these directions are constructed, consider recursive scheme

$$s^{(k)} = \beta_k s^{(k-1)} + g^{(k)} \quad \text{(134 & 135)}$$

Premultiplying $[s^{(k)}]^T$ by $A s^{(k-1)}$, we have

$$[s^{(k)}]^T A s^{(k-1)} = \beta_k [s^{(k-1)}]^T A s^{(k-1)} + [g^{(k)}]^T A s^{(k-1)} \quad \text{(136)}$$

$A$-conjugacy of directions $\{s^{(k)} : k = 0, 1, 2, \ldots\}$ can be achieved if we choose

$$\beta_k = -\frac{[g^{(k)}]^T A s^{(k-1)}}{[s^{(k-1)}]^T A s^{(k-1)}} \quad \text{(137)}$$

Thus, given a search direction $s^{(k-1)}$, new search direction is constructed as follows

$$\text{--------(138)}$$
\[ s^{(k)} = -\frac{[g^{(k)}]^T A s^{(k-1)}}{[s^{(k-1)}]^T A s^{(k-1)}} + g^{(k)} \]
\[ = g^{(k)} - \langle g^{(k)}, \hat{s}^{(k-1)} \rangle_A \hat{s}^{(k-1)} \]

where
\[ \hat{s}^{(k-1)} = \frac{s^{(k-1)}}{\sqrt{[s^{(k-1)}]^T A s^{(k-1)}}} = \frac{s^{(k-1)}}{\sqrt{\langle s^{(k-1)}, s^{(k-1)} \rangle_A}} \]

It may be that matrix \( A \) is assumed to be symmetric and positive definite. Now, recursive use of equations (CJ-1-CJ-2) yields
\[
\begin{align*}
    s^{(0)} &= g^{(0)} \\
    s^{(1)} &= \beta_1 s^{(0)} + g^{(1)} = \beta_1 g^{(0)} + g^{(1)} \\
    s^{(2)} &= \beta_2 s^{(1)} + g^{(2)} = \beta_2 \beta_1 g^{(0)} + \beta_2 g^{(1)} + g^{(2)} \\
    &\vdots \\
    s^{(n)} &= (\beta_n \ldots \beta_1) g^{(0)} + (\beta_n \ldots \beta_2) g^{(1)} + \ldots + g^{(n)}
\end{align*}
\]

Thus, this procedure sets up each new search direction as a linear combination of all previous search directions and newly determined gradient.

Now, given the new search direction, the line search is formulated as follows
\[
\lambda_k = \min_{\lambda} \phi \left( s^{(k)} + \lambda \hat{s}^{(k)} \right) \quad \text{(139)}
\]

Solving the one dimensional optimization problem yields
\[
\lambda_k = \frac{b^T s^{(k)}}{\langle s^{(k)}, A s^{(k)} \rangle} \quad \text{(140)}
\]

Thus, the conjugate gradient search algorithm for solving \( Ax = b \) is summarized in Table 6

Table 6: Conjugate Gradient Algorithm to Solve Linear Algebraic Equations

| INITIALIZE: \( x^{(0)}, e, k_{\text{max}}, \lambda^{(0)}, \delta \) |
| \( k = 0 \) |
| \( g^{(0)} = b - Ax^{(0)} \) |
| \( s^{(-1)} = 0 \) |
| WHILE [\( (e > \varepsilon) \AND (k < k_{\text{max}}) \)] |
| \( \beta_k = -\frac{[g^{(k)}]^T A s^{(k-1)}}{[s^{(k-1)}]^T A s^{(k-1)}} \) |
| \( s^{(k)} = \beta_k s^{(k-1)} + g^{(k)} \) |
| \( \lambda_k = -\frac{b^T s^{(k)}}{\langle s^{(k)}, A s^{(k)} \rangle} \) |
| \( x^{(k+1)} = x^{(k)} + \lambda_k s^{(k)} \) |
| \( g^{(k+1)} = b - Ax^{(k+1)} \) |
If conjugate gradient method is used for solving the optimization problem, it can be theoretically shown that the minimum can be reached in $\nu$ steps. In practice, however, we require more than $\nu$ steps to achieve $\phi(x) < \varepsilon$ due to the rounding off errors in computation of the conjugate directions. Nevertheless, when $\nu$ is large, this approach can generate a reasonably accurate solution with considerably less computations.