Introduction to Engineering Optimization (ME6806)

Dr. Yogesh Kumar
Assistant Professor
Mechanical Engineering Department
National Institute of Technology Patna
Bihar - 800 005, India
yogesh.me@nitp.ac.in

Module 4

Multi-variable Optimization Algorithms

Outlines

- **Optimality Criteria**
- **Direct Search Methods**
  - Nelder and Mead (Simplex Search)
  - Hook and Jeeves (Pattern Search)
  - Powell’s Method (The Conjugated Direction Search)
- **Gradient Based Methods**
  - Steepest Descent (Cauchy’s) Method
  - Newton’s Method
  - Modified Newton’s Method
  - **Marquardt’s Method**
  - Conjugate Gradient Method
  - Quasi-Newton Method
  - Trust Regions
  - Gradient-Based Algorithm
  - Numerical Gradient Approximations

Marquardt Method

- The steepest descent method reduces the function value when the design vector X_i is away from the optimum point X^*.
- The Newton method, on the other hand, converges fast when the design vector X_i is close to the optimum point X^*.
- The Marquardt method attempts to take advantage of both the steepest descent and Newton methods. This method modifies the diagonal elements of the Hessian matrix, $[J_i]$, as

$$[\tilde{J}_i] = [J_i] + \alpha_i [I]$$

where $[I]$ is an identity matrix and α_i is a positive constant that ensures the positive definiteness of $[\tilde{J}_i]$ when $[J_i]$ is not positive definite. It can be noted that when α_i is sufficiently large (on the order of 10^4), the term $\alpha_i[I]$ dominates $[J_i]$ and the inverse of the matrix $[\tilde{J}_i]$ becomes

$$[\tilde{J}_i]^{-1} = [[J_i] + \alpha_i [I]]^{-1} \approx [\alpha_i [I]]^{-1} = \frac{1}{\alpha_i}[I]$$

Marquardt Method

$$[\tilde{J}_i]^{-1} = [[J_i] + \alpha_i[I]]^{-1} \approx [\alpha_i[I]]^{-1} = \frac{1}{\alpha_i}[I]$$

Thus if the search direction $\mathbf{S}_i$ is computed as

$$\mathbf{S}_i = -[\tilde{J}_i]^{-1}\nabla f_i$$

$\mathbf{S}_i$ becomes a steepest descent direction for large values of α_i. In the Marquardt method, the value of α_i is taken to be large at the beginning and then reduced to zero gradually as the iterative process progresses. Thus as the value of α_i decreases from a large value to zero, the characteristics of the search method change from those of a steepest descent method to those of the Newton method.

Marquardt Method

The iterative process of a modified version of Marquardt method can be described as follows:

1. Start with an arbitrary initial point $\mathbf{X}_1$ and constants α_1 (on the order of 10^4), $c_1 (0 < c_1 < 1)$, $c_2 (c_2 > 1)$, and ε (on the order of 10^{-2}). Set the iteration number as $i = 1$.

2. Compute the gradient of the function, $\nabla f_i = \nabla f(\mathbf{X}_i)$.

3. Test for optimality of the point $\mathbf{X}_i$. If $\|\nabla f_i\| = \|\nabla f(\mathbf{X}_i)\| \leq \varepsilon$, $\mathbf{X}_i$ is optimum and hence stop the process. Otherwise, go to step 4.

4. Find the new vector $\mathbf{X}_{i+1}$ as

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \mathbf{S}_i = \mathbf{X}_i - [[J_i]] + \alpha_i [I]]^{-1} \quad \nabla f_i$$

5. Compare the values of f_{i+1} and f_i. If $f_{i+1} < f_i$, go to, step 6. If $f_{i+1} \geq f_i$, go to step 7.

Marquardt Method

6. Set $\alpha_{i+1} = c_1\alpha_i$, $i = i + 1$, and go to step 2.

7. Set $\alpha_i = c_2\alpha_i$ and go to step 4.

An advantage of this method is the absence of the step size λ_i along the search direction $\mathbf{S}_i$. In fact, the algorithm above can be modified by introducing an optimal step length

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda_i^* \mathbf{S}_i = \mathbf{X}_i - \lambda_i^* [[J_i] + \alpha_i [I]]^{-1} \nabla f_i$$

Marquardt Method - Example

Minimize $f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$ from the starting point $\mathbf{X}_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$ using Marquardt method with $\alpha_1 = 10^4$, $c_1 = \frac{1}{4}$, $c_2 = 2$, and $\varepsilon = 10^{-2}$.

Solution

Iteration 1 (i = 1)

Here $f_1 = f(\mathbf{X}_1) = 0.0$ and

$$\nabla f_1 = \begin{Bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \end{Bmatrix}_{(0,0)} = \begin{Bmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{Bmatrix}_{(0,0)} = \begin{Bmatrix} 1 \\ -1 \end{Bmatrix}$$

Since $\|\nabla f_1\| = 1.4142 > \varepsilon$, we compute $[J_1]$

Marquardt Method - Example

$$[J_1] = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} \\ \frac{\partial^2}{\partial x_1 x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}_{(0,0)} = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$

$$\mathbf{X}_2 = \mathbf{X}_1 - [[J_1] + \alpha_1 [I]]^{-1} \nabla f_1$$

$$= \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} - \begin{bmatrix} 4 + 10^4 & 2 \\ 2 & 2 + 10^4 \end{bmatrix}^{-1} \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} = \begin{Bmatrix} -0.9998 \\ 1.0000 \end{Bmatrix} 10^{-4}$$

As $f_2 = f(\mathbf{X}_2) = -1.9997 \times 10^{-4} < f_1$, we set $\alpha_2 = c_1 \alpha_1 = 2500$, $i = 2$, and proceed to the next iteration.

Marquardt Method - Example

Iteration 2 (i = 2)

The gradient vector corresponding to $\mathbf{X}_2$ is given by $\nabla f_2 = \begin{Bmatrix} 0.9998 \\ -1.0000 \end{Bmatrix}$, $\|\nabla f_2\| = 1.4141 > \varepsilon$, and hence we compute

$$\mathbf{X}_3 = \mathbf{X}_2 - [[J_2] + \alpha_2[I]]^{-1}\nabla f_2$$

$$= \begin{Bmatrix} -0.9998 \times 10^{-4} \\ 1.0000 \times 10^{-4} \end{Bmatrix} - \begin{bmatrix} 2504 & 2 \\ 2 & 2502 \end{bmatrix}^{-1} \begin{Bmatrix} 0.9998 \\ -1.0000 \end{Bmatrix}$$

$$= \begin{Bmatrix} -4.9958 \times 10^{-4} \\ 5.0000 \times 10^{-4} \end{Bmatrix}$$

Since $f_3 = f(\mathbf{X}_3) = -0.9993 \times 10^{-3} < f_2$, we set $\alpha_3 = c_1\alpha_2 = 625$, $i = 3$, and proceed to the next iteration. The iterative process is to be continued until the convergence criterion, $\|\nabla f_i\| < \varepsilon$, is satisfied.