Chapter 9 Newton's Method

An Introduction to Optimization

Spring, 2014

Wei-Ta Chu

Introduction

- The steepest descent method uses only first derivatives in selecting a suitable search direction.
- Newton's method (sometimes called Newton-Raphson method) uses first and second derivatives and indeed performs better.
- Given a starting point, construct a quadratic approximation to the objective function that matches the first and second derivative values at that point. We then minimize the approximate (quadratic function) instead of the original objective function. The minimizer of the approximate function is used as the starting point in the next step and repeat the procedure iteratively.

$$f(b) = f(a) + \frac{h}{1!}f^{(1)}(a) + \frac{h^2}{2!}f^{(2)}(a) + \cdots + \frac{h^{m-1}}{(m-1)!}f^{(m-1)}(a) + R_m$$

Introduction

- We can obtain a quadratic approximation to the twice continuously differentiable function $f : R^n \rightarrow R$ using the Taylor series expansion of f about the current point $\boldsymbol{x}^{(k)}$, neglecting terms of order three and higher.

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}^{(k)}) + (\boldsymbol{x} - \boldsymbol{x}^{(k)})^T \boldsymbol{g}^{(k)} + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{(k)})^T F(\boldsymbol{x}^{(k)})(\boldsymbol{x} - \boldsymbol{x}^{(k)}) \triangleq q(\boldsymbol{x})$$

Where, for simplicity, we use the notation $\boldsymbol{g}^{(k)} = \bigtriangledown f(\boldsymbol{x}^{(k)})$

- Applying the FONC to q yields

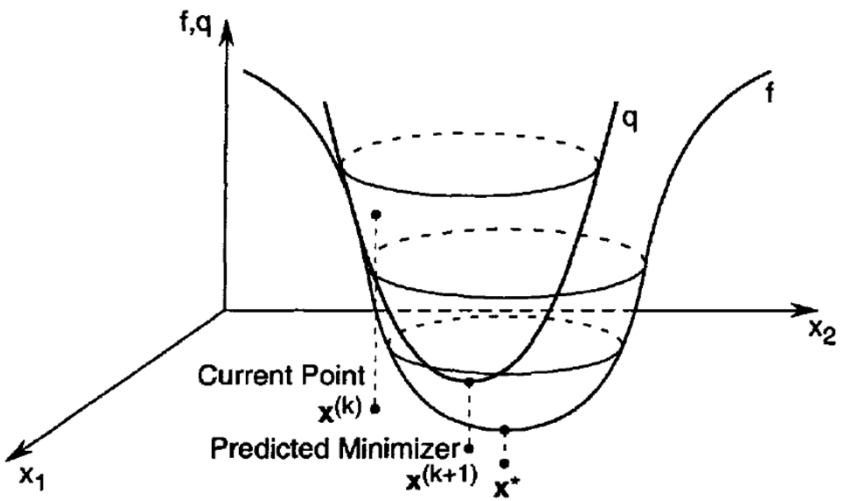
$$\mathbf{0} = \bigtriangledown q(\boldsymbol{x}) = \boldsymbol{g}^{(k)} + F(\boldsymbol{x}^{(k)})(\boldsymbol{x} - \boldsymbol{x}^{(k)})$$

- If $F(\boldsymbol{x}^{(k)}) > 0$, then q achieves a minimum at

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1}\boldsymbol{g}^{(k)}$$

$$0 = q'(x) = f'(x^{(k)}) + f''(x^{(k)})(x - x^{(k)})$$

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$$



Example

- Use Newton's method to minimize the Powell function:

$$f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

  Use as the starting point $\boldsymbol{x}^{(0)} = [3, -1, 0, 1]^T$. Perform three iterations.

- Note that $f(\boldsymbol{x}^{(0)}) = 215$. We have

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} 2(x_1 + 10x_2) + 40(x_1 - x_4)^3 \\ 20(x_1 + 10x_2) + 4(x_2 - 2x_3)^3 \\ 10(x_3 - x_4) - 8(x_2 - 2x_3)^3 \\ -10(x_3 - x_4) - 40(x_1 - x_4)^3 \end{bmatrix}$$

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} 2 + 120(x_1 - x_4) & 20 & 0 & -120(x_1 - x_4) \\ 20 & 200 + 12(x_2 - 2x_3)^2 & -24(x_2 - 2x_3)^2 & 0 \\ 0 & -24(x_2 - 2x_3)^2 & 10 + 48(x_2 - 2x_3)^2 & -10 \\ -120(x_1 - x_4)^2 & 0 & -10 & 10 + 120(x_1 - x_4)^2 \end{bmatrix}$$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1}\boldsymbol{g}^{(k)}$$

Example

- Iteration 1.

$$\boldsymbol{g}^{(0)} = [306, -114, -2, -310]^T$$

$$\boldsymbol{F}(\boldsymbol{x}^{(0)}) = \begin{bmatrix} 482 & 20 & 0 & -480 \\ 20 & 212 & -24 & 0 \\ 0 & -24 & 58 & -10 \\ -480 & 0 & -10 & 490 \end{bmatrix} \quad \boldsymbol{F}(\boldsymbol{x}^{(0)})^{-1} = \begin{bmatrix} 0.1126 & -0.0089 & 0.0154 & 0.1106 \\ -0.0089 & 0.0057 & 0.0008 & -0.0087 \\ 0.0154 & 0.0008 & 0.0203 & 0.0155 \\ 0.1106 & -0.0087 & 0.0155 & 0.1107 \end{bmatrix}$$

$$\boldsymbol{F}(\boldsymbol{x}^{(0)})^{-1}\boldsymbol{g}^{(0)} = [1.4127, -0.1587, 0.2540, 0.2540]^T$$

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} - \boldsymbol{F}(\boldsymbol{x}^{(0)})^{-1}\boldsymbol{g}^{(0)} = [1.5873, -0.1587, 0.2540, 0.2540]^T$$

$$f(\boldsymbol{x}^{(1)}) = 31.8$$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1}\boldsymbol{g}^{(k)}$$

Example

- Iteration 2.

$$\boldsymbol{g}^{(1)} = [94.81, -1.179, 2.371, -94.81]^T$$

$$\boldsymbol{F}(\boldsymbol{x}^{(1)}) = \begin{bmatrix} 215.3 & 20 & 0 & -213.3 \\ 20 & 205.3 & -10.67 & 0 \\ 0 & -10.67 & 31.34 & -10 \\ -213.3 & 0 & -10 & 223.3 \end{bmatrix}$$

$$\boldsymbol{F}(\boldsymbol{x}^{(1)})^{-1}\boldsymbol{g}^{(1)} = [0.5291, -0.0529, 0.0846, 0.0846]^T$$

$$\boldsymbol{x}^{(2)} = \boldsymbol{x}^{(1)} - \boldsymbol{F}(\boldsymbol{x}^{(1)})^{-1}\boldsymbol{g}(1) = [1.0582, -0.1058, 0.1694, 0.1694]^T$$

$$f(\boldsymbol{x}^{(2)}) = 6.28$$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1}\boldsymbol{g}^{(k)}$$

Example

- Iteration 3.

$$\boldsymbol{g}^{(2)} = [28.09, -0.3475, 0.7031, -28.08]^T$$

$$\boldsymbol{F}(\boldsymbol{x}^{(2)}) = \begin{bmatrix} 96.80 & 20 & 0 & -94.80 \\ 20 & 202.4 & -4.744 & 0 \\ 0 & -4.744 & 19.49 & -10 \\ -94.80 & 0 & -10 & 104.80 \end{bmatrix}$$

$$\boldsymbol{x}^{(3)} = \boldsymbol{x}^{(2)} - \boldsymbol{F}(\boldsymbol{x}^{(2)})^{-1}\boldsymbol{g}^{(2)} = [0.7037, -0.0704, 0.1121, 0.1111]^T$$

$$f(\boldsymbol{x}^{(3)}) = 1.24$$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1}\boldsymbol{g}^{(k)}$$

Introduction

- Observe that the kth iteration of Newton's method can be written in two steps as
  - 1. Solve $\boldsymbol{F}(\boldsymbol{x}^{(k)})\boldsymbol{d}^{(k)} = -\boldsymbol{g}^{(k)}$ for $\boldsymbol{d}^{(k)}$
  - 2. Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{d}^{(k)}$
- Step 1 requires the solution of an $n \times n$ system of linear equations. Thus, an efficient method for solving systems of linear equations is essential when using Newton's method.
- As in the one-variable case, Newton's method can be viewed as a technique for iteratively solving the equation

$$\boldsymbol{g}(\boldsymbol{x}) = \mathbf{0}$$

where $\boldsymbol{x} \in R^n$ and $\boldsymbol{g} : R^n \to R^n$. In this case $\boldsymbol{F}(\boldsymbol{x})$ is the Jacobian matrix of $\boldsymbol{g}$ at $\boldsymbol{x}$; that is, $\boldsymbol{F}(\boldsymbol{x})$ is the $n \times n$ matrix whose (i, j) entry is $(\partial g_i/\partial x_j)(\boldsymbol{x})$, $i, j = 1, 2, ..., n$

Analysis of Newton's Method

- As in the one-variable case there is no guarantee that Newton's algorithm heads in the direction of decreasing values of the objective function if $\boldsymbol{F}(\boldsymbol{x}^{(k)})$ is not positive definite (recall Figure 7.7)
- Even if $\boldsymbol{F}(\boldsymbol{x}^{(k)}) > 0$, Newton's method may not be a descent method; that is, it is possible that $f(\boldsymbol{x}^{(k+1)}) > f(\boldsymbol{x}^{(k)})$
  - This may occur if our starting point is far away from the solution
- Despite these drawbacks, Newton's method has superior convergence properties when the starting point is near the solution.

Newton's method works well if $f''(x) > 0$ everywhere. However, if $f''(x) < 0$ for some x, Newton's method may fail to converge to the minimizer.

Analysis of Newton's Method

- The convergence analysis of Newton's method when f is a quadratic function is straightforward. Newton's method reaches the point $\boldsymbol{x}^*$ such that $\bigtriangledown f(\boldsymbol{x}^*) = \mathbf{0}$ in just one step starting from any initial point $\boldsymbol{x}^{(0)}$.
- Suppose that $\boldsymbol{Q} = \boldsymbol{Q}^T$ is invertible and $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b}$ Then, $\boldsymbol{g}(\boldsymbol{x}) = \bigtriangledown f(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{b}$ and $\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{Q}$
- Hence, given any initial point $\boldsymbol{x}^{(0)}$, by Newton's algorithm

$$\begin{aligned}\boldsymbol{x}^{(1)} &= \boldsymbol{x}^{(0)} - \boldsymbol{F}(\boldsymbol{x}^{(0)})^{-1}\boldsymbol{g}^{(0)} \\ &= \boldsymbol{x}^{(0)} - \boldsymbol{Q}^{-1}[\boldsymbol{Q}\boldsymbol{x}^{(0)} - \boldsymbol{b}] \\ &= \boldsymbol{Q}^{-1}\boldsymbol{b} \\ &= \boldsymbol{x}^*\end{aligned}$$

- Therefore, for the quadratic case the order of convergence of Newton's algorithm is ∞ for any initial point $\boldsymbol{x}^{(0)}$

Analysis of Newton's Method

- Theorem 9.1: Suppose that $f \in \mathcal{C}^3$ and $\boldsymbol{x}^* \in R^n$ is a point such that $\bigtriangledown f(\boldsymbol{x}^*) = \mathbf{0}$ and $\boldsymbol{F}(\boldsymbol{x}^*)$ is invertible. Then, for all $\boldsymbol{x}^{(0)}$ sufficiently close to $\boldsymbol{x}^*$, Newton's method is well defined for all k and converge to $\boldsymbol{x}^*$ with an order of convergence at least 2.
- Proof: The Taylor series expansion of $\bigtriangledown f$ about $\boldsymbol{x}^{(0)}$ yields

$$\bigtriangledown f(\boldsymbol{x}) - \bigtriangledown f(\boldsymbol{x}^{(0)}) - \boldsymbol{F}(\boldsymbol{x} - \boldsymbol{x}^{(0)}) = O(\|\boldsymbol{x} - \boldsymbol{x}^{(0)}\|^2)$$

Because by assumption $f \in \mathcal{C}^3$ and $\boldsymbol{F}(\boldsymbol{x}^*)$ is invertible, there exist constants $\epsilon > 0$, $c_1 > 0$ and $c_2 > 0$ such that if $\boldsymbol{x}^{(0)}$, $\boldsymbol{x} \in \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \epsilon\}$, we have

$$\|\bigtriangledown f(\boldsymbol{x}) - \bigtriangledown f(\boldsymbol{x}^{(0)}) - \boldsymbol{F}(\boldsymbol{x} - \boldsymbol{x}^{(0)})\| \leq c_1 \|\boldsymbol{x} - \boldsymbol{x}^{(0)}\|^2$$

and by Lemma 5.3, $\boldsymbol{F}(\boldsymbol{x})^{-1}$ exists and satisfies

$$\|\boldsymbol{F}(\boldsymbol{x})^{-1}\| \leq c_2$$

Analysis of Newton's Method

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x}^{(0)}) - \boldsymbol{F}(\boldsymbol{x} - \boldsymbol{x}^{(0)})\| \leq c_1 \|\boldsymbol{x} - \boldsymbol{x}^{(0)}\|^2$$

$$\|\boldsymbol{F}(\boldsymbol{x})^{-1}\| \leq c_2$$

- The first inequality holds because the remainder term in the Taylor series expansion contains third derivatives of f that are continuous and hence bounded on $\{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \epsilon\}$
- Suppose that $\boldsymbol{x}^{(0)} \in \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq \epsilon\}$. Then, substituting $\boldsymbol{x} = \boldsymbol{x}^*$ in the inequality above and using the assumption that $\nabla f(\boldsymbol{x}^*) = \boldsymbol{0}$ we get

$$\|\boldsymbol{F}(\boldsymbol{x}^{(0)})(\boldsymbol{x}^{(0)} - \boldsymbol{x}^*) - \nabla f(\boldsymbol{x}^{(0)})\| \leq c_1 \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|^2$$

Analysis of Newton's Method

- Subtracting $\boldsymbol{x}^*$ from both sides of Newton's algorithm and taking norms yields

$$\begin{aligned}\|\boldsymbol{x}^{(1)} - \boldsymbol{x}^*\| &= \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^* - \boldsymbol{F}(\boldsymbol{x}^{(0)})^{-1} \bigtriangledown f(\boldsymbol{x}^{(0)})\| \\ &= \|\boldsymbol{F}(\boldsymbol{x}^{(0)})^{-1}(\boldsymbol{F}(\boldsymbol{x}^{(0)}))(\boldsymbol{x}^{(0)} - \boldsymbol{x}^*) - \bigtriangledown f(\boldsymbol{x}^{(0)})\| \\ &\leq \|\boldsymbol{F}(\boldsymbol{x}^{(0)})^{-1}\|\|(\boldsymbol{F}(\boldsymbol{x}^{(0)}))(\boldsymbol{x}^{(0)} - \boldsymbol{x}^*) - \bigtriangledown f(\boldsymbol{x}^{(0)})\|\end{aligned}$$

- Applying the inequalities above involving the constants c_1 and c_2

$$\|\boldsymbol{x}^{(1)} - \boldsymbol{x}^*\| \leq c_1 c_2 \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|^2$$

- Suppose that $\boldsymbol{x}^{(0)}$ is such that

$$\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| \leq \frac{\alpha}{c_1 c_2} \qquad \alpha \in (0, 1)$$

Then

$$\|\boldsymbol{x}^{(1)} - \boldsymbol{x}^*\| \leq \alpha \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|$$

Analysis of Newton's Method

- By induction, we obtain

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\| \leq c_1 c_2 \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|^2$$
$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\| \leq \alpha \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*|$$

Hence, $\lim_{k\to\infty} \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\| = 0$ and therefore the sequence $\{\boldsymbol{x}_k\}$ converges to $\boldsymbol{x}^*$. The order of convergence is at least 2 because $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\| \leq c_1 c_2 \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|^2$. That is, $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\| = O(\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|^2)$

Analysis of Newton's Method

- Theorem 9.2: Let $\{\boldsymbol{x}_k\}$ be the sequence generated by Newton's method for minimizing a given objective function $f(\boldsymbol{x})$. If the Hessian $\boldsymbol{F}(\boldsymbol{x}^{(k)}) > 0$ and $\boldsymbol{g}^{(k)} = \bigtriangledown f(\boldsymbol{x}^{(k)}) \neq \mathbf{0}$, then the search direction

$$\boldsymbol{d}^{(k)} = -\boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1}\boldsymbol{g}^{(k)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$$

  from $\boldsymbol{x}^{(k)}$ to $\boldsymbol{x}^{(k+1)}$ is a descent direction for f in the sense that there exists an $\bar{\alpha} > 0$ such that for all $\alpha \in (0, \bar{\alpha})$

$$f(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}) < f(\boldsymbol{x}^{(k)})$$

Analysis of Newton's Method

- Proof: Let $\phi(\alpha) = f(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)})$, then using the chain rule, we obtain

$$\phi'(\alpha) = \bigtriangledown f(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)})^T \boldsymbol{d}^{(k)}$$

Hence, $\phi'(0) = \bigtriangledown f(\boldsymbol{x}^{(k)})^T \boldsymbol{d}^{(k)} = -\boldsymbol{g}^{(k)T} \boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1} \boldsymbol{g}^{(k)} < 0$

because $\boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1} > 0$ and $\boldsymbol{g}^{(k)} \neq \mathbf{0}$.
Thus, there exists an $\bar{\alpha} > 0$ so that for all $\alpha \in (0, \bar{\alpha})$, $\phi(\alpha) < \phi(0)$
This implies that for all $\alpha \in (0, \bar{\alpha})$

$$f(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}) < f(\boldsymbol{x}^{(k)})$$

Analysis of Newton's Method

- Theorem 9.2 motivates the following modification of Newton's method

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1}\boldsymbol{g}^{(k)}$$

  where $\alpha_k = \arg\min_{\alpha \geq 0} f(\boldsymbol{x}^{(k)} - \alpha \boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1}\boldsymbol{g}^{(k)})$
  that is, at each iteration, we perform a line search in the direction $-\boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1}\boldsymbol{g}^{(k)}$

- A drawback of Newton's method is that evaluation of $\boldsymbol{F}(\boldsymbol{x}^{(k)})$ for large n can be computationally expensive. Furthermore, we have to solve the set of n linear equations $\boldsymbol{F}(\boldsymbol{x}^{(k)})\boldsymbol{d}^{(k)} = -\boldsymbol{g}^{(k)}$. In Chapters 10 and 11 we discuss this issue.

- The Hessian matrix may not be positive definite. In the next we describe a simple modification to overcome this problem.

Levenberg-Marquardt Modification

- If the Hessian matrix $\boldsymbol{F}(\boldsymbol{x}^{(k)})$ is not positive definite, then the search direction $\boldsymbol{d}^{(k)} = -\boldsymbol{F}(\boldsymbol{x}^{(k)})^{-1}\boldsymbol{g}^{(k)}$ may not point in a descent direction.
- ***Levenberg-Marquardt modification***:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - (\boldsymbol{F}(\boldsymbol{x}^{(k)}) + \mu_k \boldsymbol{I})^{-1}\boldsymbol{g}^{(k)} \qquad \mu_k \geq 0$$

- Consider a symmetric matrix $\boldsymbol{F}$, which may not be positive definite. Let $\lambda_1, ..., \lambda_n$ be the eigenvalues of $\boldsymbol{F}$ with corresponding eigenvectors $\boldsymbol{v}_1, ..., \boldsymbol{v}_n$. The eigenvalues are real, but may not all be positive.
- Consider the matrix $\boldsymbol{G} = \boldsymbol{F} + \mu \boldsymbol{I}$, where $\mu \geq 0$. Note that the eigenvalues of $\boldsymbol{G}$ are $\lambda_1 + \mu, ..., \lambda_n + \mu$.

Levenberg-Marquardt Modification

- Indeed,
$$\begin{aligned}\boldsymbol{G}\boldsymbol{v}_i &= (\boldsymbol{F} + \mu\boldsymbol{I})\boldsymbol{v}_i \\ &= \boldsymbol{F}\boldsymbol{v}_i + \mu\boldsymbol{I}\boldsymbol{v}_i \\ &= \lambda_i\boldsymbol{v}_i + \mu\boldsymbol{v}_i \\ &= (\lambda_i + \mu)\boldsymbol{v}_i\end{aligned}$$
which shows that for all $i = 1, ..., n$, $\boldsymbol{v}_i$ is also an eigenvector of $\boldsymbol{G}$ with eigenvalue $\lambda_i + \mu$.
- If μ is sufficiently large, then all the eigenvalues of $\boldsymbol{G}$ are positive and $\boldsymbol{G}$ is positive definite.
- Accordingly, if the parameter μ_k in the Levenberg-Marquardt modification of Newton's algorithm is sufficiently large, then the search direction $\boldsymbol{d}^{(k)} = -(\boldsymbol{F}(\boldsymbol{x}^{(k)}) + \mu_k\boldsymbol{I})^{-1}\boldsymbol{g}^{(k)}$ always points in a descent direction.

Levenberg-Marquardt Modification

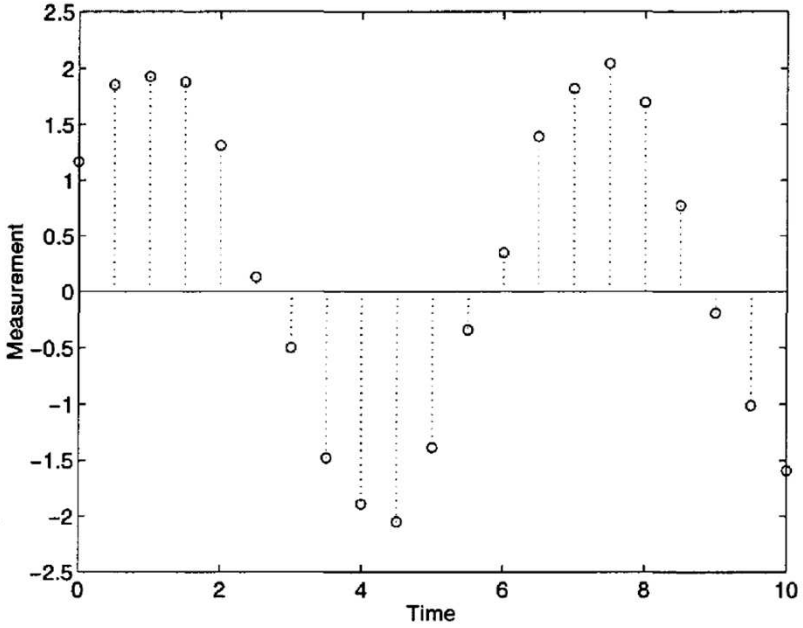
- If we further introduce a step size α_k

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k (\boldsymbol{F}(\boldsymbol{x}^{(k)}) + \mu_k \boldsymbol{I})^{-1} \boldsymbol{g}^{(k)}$$

  then we are guaranteed that the descent property holds.
- By letting $\mu_k \to 0$, the Levenberg-Marquardt modification approaches the behavior of the pure Newton's method.
- By letting $\mu_k \to \infty$, this algorithm approaches a pure gradient method with small step size.
- In practice, we may start with a small value of μ_k and increase it slowly until we find that the iteration is descent: $f(\boldsymbol{x}^{(k+1)}) < f(\boldsymbol{x}^{(k)})$

Newton's Method for Nonlinear Least Squares

- Consider $\text{minimize } \sum_{i=1}^{m}(r_i(\boldsymbol{x}))^2$, where $r_i : R^n \to R$, $i = 1, ..., m$ are given functions. This particular problem is called a ***nonlinear least-squares problem***.
- Suppose that we are given m measurements of a process at m points in time. Let $t_1, ..., t_m$ denote the measurement times and $y_1, ..., y_m$ the measurements values. Note that $t_1 = 0$ and $t_{21} = 10$ We wish to fit a sinusoid to the measurement data.



Newton's Method for Nonlinear Least Squares

- The equation of the sinusoid is
$$y = A\sin(\omega t + \phi)$$
with appropriate choices of the parameters A, ω, ϕ.
- To formulate the data-fitting problem, we construct the objective function
$$\sum_{i=1}^{m}(y_i - A\sin(\omega t_i + \phi))^2$$
representing the sum of the squared errors between the measurement values and the function values at the corresponding points in time.
- Let $\boldsymbol{x} = [A, \omega, \phi]^T$ represent the vector of decision variables. We obtain the least-squares problem with
$$r_i(\boldsymbol{x}) = y_i - A\sin(\omega t_i + \phi)$$

Newton's Method for Nonlinear Least Squares

- Defining $\boldsymbol{r} = [r_1, ..., r_m]^T$, we write the objective function as $f(\boldsymbol{x}) = \boldsymbol{r}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x})$. To apply Newton's method, we need to compute the gradient and the Hessian of f.
- The j th component of $\bigtriangledown f(\boldsymbol{x})$ is

$$(\bigtriangledown f(\boldsymbol{x}))_j = \frac{\partial f}{\partial x_j}(\boldsymbol{x}) = 2\sum_{i=1}^{m} r_i(\boldsymbol{x}) \frac{\partial r_i}{\partial x_j}(\boldsymbol{x})$$

- Denote the Jacobian matrix of $\boldsymbol{r}$ by

$$\boldsymbol{J}(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial r_1}{\partial x_1}(\boldsymbol{x}) & \cdots & \frac{\partial r_1}{\partial x_n}(\boldsymbol{x}) \\ \vdots & & \vdots \\ \frac{\partial r_m}{\partial x_1}(\boldsymbol{x}) & \cdots & \frac{\partial r_m}{\partial x_n}(\boldsymbol{x}) \end{bmatrix}$$

- Thus, the gradient of f can be represented as

$$\bigtriangledown f(\boldsymbol{x}) = 2\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x})$$

Newton's Method for Nonlinear Least Squares

- We compute the Hessian matrix of f. The (k,j)th component of the Hessian is given by

$$\begin{aligned}\frac{\partial^2 f}{\partial x_k \partial x_j}(\boldsymbol{x}) &= \frac{\partial}{\partial x_k}\left(\frac{\partial f}{\partial x_j}(\boldsymbol{x})\right) \\ &= \frac{\partial}{\partial x_k}\left(2\sum_{i=1}^m r_i(\boldsymbol{x})\frac{\partial r_i}{\partial x_j}(\boldsymbol{x})\right) \\ &= 2\sum_{i=1}^m \left(\frac{\partial r_i}{\partial x_k}(\boldsymbol{x})\frac{\partial r_i}{\partial x_j}(\boldsymbol{x}) + r_i(\boldsymbol{x})\frac{\partial^2 r_i}{\partial x_k \partial x_j}(\boldsymbol{x})\right)\end{aligned}$$

- Letting $\boldsymbol{S}(\boldsymbol{x})$ be the matrix whose (k,j)th component is

$$\sum_{i=1}^m r_i(\boldsymbol{x})\frac{\partial^2 r_i}{\partial x_k \partial x_j}(\boldsymbol{x})$$

- We write the Hessian matrix as

$$\boldsymbol{F}(\boldsymbol{x}) = 2(\boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{J}(\boldsymbol{x}) + \boldsymbol{S}(\boldsymbol{x}))$$

Newton's Method for Nonlinear Least Squares

- Therefore, Newton's method applied to the nonlinear least-squares problem is given by

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - (\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{J}(\boldsymbol{x}) + \boldsymbol{S}(\boldsymbol{x}))^{-1} \boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x})$$

- In some applications, the matrix $\boldsymbol{S}(\boldsymbol{x})$ involving the second derivatives of the function $\boldsymbol{r}$ can be ignored because its components are negligibly small.
- In this case Newton's algorithm reduces to what is commonly called the ***Gauss-Newton method***:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - (\boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{J}(\boldsymbol{x}))^{-1} \boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{r}(\boldsymbol{x})$$

  Note that the Gauss-Newton method does not require calculation of the second derivatives of $\boldsymbol{r}$

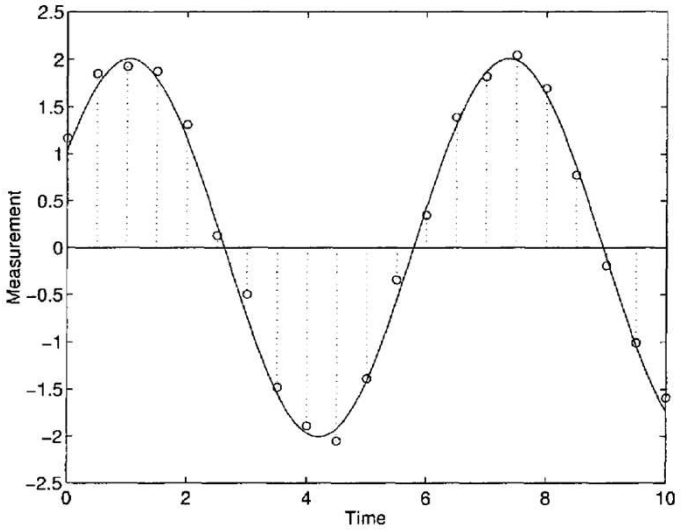
Example

$$r_i(\boldsymbol{x}) = y_i - A\sin(\omega t + \phi) \qquad i = 1, ..., 21$$

- The Jacobian matrix $J(\boldsymbol{x})$ in this problem is a 21×3 matrix with elements given by

$$\begin{aligned}
(J(\boldsymbol{x}))_{(i,1)} &= -\sin(\omega t_i + \phi) \\
(J(\boldsymbol{x}))_{(i,2)} &= -t_i A\cos(\omega t_i + \phi) \qquad i = 1, ..., 21 \\
(J(\boldsymbol{x}))_{(i,3)} &= -A\cos(\omega t_i + \phi)
\end{aligned}$$

- We apply the Gauss-Newton algorithm to find the sinusoid of best fit.
- The parameters of this sinusoid are $A = 2.01, \omega = 0.992, \phi = 0.541$



Newton's Method for Nonlinear Least Squares

- A potential problem with the Gauss-Newton method is that the matrix $\boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{J}(\boldsymbol{x})$ may not be positive definite.
- This problem can be overcome using a Levenberg-Marquardt modification:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - (\boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{J}(\boldsymbol{x}) + \mu_k\boldsymbol{I})^{-1}\boldsymbol{J}(\boldsymbol{x})^T\boldsymbol{r}(\boldsymbol{x})$$

- This is referred to in the literature as the ***Levenberg-Marquardt algorithm*** because the original modification was developed specifically for the nonlinear least-squares problem.
- An alternative interpretation of the Levenberg-Marquardt algorithm is to view the term $\mu_k\boldsymbol{I}$ as an approximation to $\boldsymbol{S}(\boldsymbol{x})$ in the Newton's algorithm.