An Introduction to Nonlinear Optimization

We give here a brief introduction to nonlinear optimization and related concepts. The problem is to find an $\mathbf{x}^*$ which minimizes the objective function $f(\mathbf{x})$ so that

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\Omega} f(\mathbf{x}). \tag{1}$$

The Ω gives the set of *admissible solutions* for the problem and can be used to represent constraints. We call a point $\mathbf{x} \in \Omega$ a *feasible point.* In the unconstrained case, we take $\Omega = \mathbb{R}^n$.

The field of optimization is concerned with establishing theory for equation 1 when different assumptions are made about f and Ω, and with developing practical computational methods for approximating solutions. Non-linear optimization has many applications in the sciences, engineering, statistics, economics, machine learning, AI, and other disciplines.

A common strategy is to develop iterative methods with the goal of yielding a sequence $\{\mathbf{x}_n\}_{n=1}^{\infty}$ such that $\mathbf{x}_n \to \mathbf{x}^*$, where $f(\mathbf{x}^*) = \min_{\mathbf{x}\in\Omega} f(\mathbf{x})$. In these notes, as a starting point for discussions, we shall discuss primarily line search methods and handle constraints using penalty terms or projections. Further discussions can be found in the literature and books, such as [1–3].

In line search algorithms, the sequence $\{\mathbf{x}_n\}_{n=1}^{\infty}$ is constructed iteratively at each step choosing a search direction $\mathbf{p}_k$ and attempting to minimize the objective function along a line or ray in this direction. This reduces the problem to a sequence of one dimensional problems with objective $\phi(\alpha) = f(\mathbf{x}_k + \alpha\mathbf{p}_k)$. This yields $\mathbf{x}_k$ given by the recurrence

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k. \tag{2}$$

An important part of these methods is to determine at each stage a choice for the step-length α_k.

The key to having a method that generates a sequence $\mathbf{x}_k$ which converges rapidly is to construct an algorithm which makes effective use of assumptions about $f(x)$ and information from previous iterates. For these iterative methods, this requires good choices for $\mathbf{p}_k$ and the step-length α_k.

A natural condition to try to impose on $\mathbf{p}_k$, to help ensure that progress can be made each iteration, is that the function decrease at least locally in this direction. We call $\mathbf{p}_k$ a *descent direction* if $\nabla f^T\mathbf{p}_k < 0$. Typically, the descent direction has the form $\mathbf{p}_k = -B_k^{-1}\nabla f_k$. In the case that B_k is positive definite this ensures that $\mathbf{p}_k$ is a descent direction

$$\nabla f_k^T\mathbf{p}_k \;\; = \;\; -\nabla f_k^T B_k^{-1}\nabla f_k < 0. \tag{3}$$

A few examples include

- **Gradient Descent** has $B_k = \mathcal{I}$ set to the identity matrix so that $\mathbf{p}_k = -\nabla f_k$.
- **Newton's Method** has $B_k = \nabla^2 f_k$ set to the Hessian so that $\mathbf{p}_k = -\left(\nabla^2 f_k\right)^{-1} \nabla f_k$. It is important to note that $\mathbf{p}_k$ is only ensured to be descent direction if the Hessian is positive definite. The Hessian however is often expensive to compute numerically.
- **Quasi-Newton Methods** construct a B_k which approximates the Hessian by making use of the previous function evaluations $f(x_k)$ and $\nabla f(x_k)$.

In designing a good numerical optimization method some care also must be taken in the choice of step-lengths α_k. This will usually depend on the structure and smoothness of the function $f(\mathbf{x})$ being optimized. We now discuss one widely-used strategy for choosing at each stage the step-length α_k.

Conditions for Steps α_k.

For the search direction $\mathbf{p}_k$, we consider the one dimensional objective function $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k)$. One choice would be to try to take α_k to be the global minimizer of $\phi(\alpha)$. However, in practice, this is typically too expensive and computational resources are usually better spent searching over more directions $\mathbf{p}_k$. In designing algorithms there is often a balance between making progress in minimizing the objective function ϕ for each search direction $\mathbf{p}_k$ vs exploring a larger variety of different directions.

Another aspect of designing algorithms is to find criteria that ensures they converge, at least to a local minimizer. For this purpose, it is important to be more mathematically precise about what we mean by making adequate progress over each search direction. We might be tempted to only require the function is reduced over a given step, $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) < f(\mathbf{x}_k)$. However, this is not sufficient to ensure convergence to a local minimizer $\mathbf{x}^*$. For example, consider $f(x, y) = x^2 + y^2$, and suppose the algorithm has the output $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ where $\alpha_k = 1/2^k$ and $\mathbf{x}_0 = [3, 0]$. Now if $\mathbf{p}_k = [-1, 0]^T$ we have $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$, but $\mathbf{x}_{k+1} = \mathbf{x}_0 + \sum_{k=0}^{\infty} 2^{-k} \mathbf{p}_k = \mathbf{x}_0 - [2, 0] = [1, 0] \neq \mathbf{x}^* = [0, 0]$. In general, if the first component of $\mathbf{x}_0$ has $x_0^{(1)} \neq 2$ then $x_k^{(1)} \not\to 0$. The issue here is that the step size decayed too rapidly resulting in insufficient progress being made each iteration toward the minimizer. This causes the sequence to prematurely converge to a sub-optimal value. While this example may seem contrived, in practice algorithms encounter similar challenges since they often also need to adjust the step size α_k during iterations.

We now discuss a set of criteria that will ensure that "sufficient progress" is made during each iteration. These are given by the following

Wolfe Conditions for $f(\mathbf{x})$:

(i) (sufficient decrease) $f(\mathbf{x}_k + \alpha \mathbf{p}_k) \leq f(\mathbf{x}_k) + c_1 \alpha \nabla f_k^T \mathbf{p}_k$, where $c_1 \in (0, 1)$.

(ii) (curvature condition) $\nabla f(\mathbf{x}_k + \alpha \mathbf{p}_k)^T \mathbf{p}_k \geq c_2 \nabla f_k^T \mathbf{p}_k$, where $c_2 \in (c_1, 1)$.

We illustrate each of these conditions in Figure 1.

Remark: The curvature condition can be interpreted by using the following equivalent expression $-\nabla f(\mathbf{x}_k + \alpha \mathbf{p}_k)^T \mathbf{p}_k \leq -c_2 \nabla f_k^T \mathbf{p}_k$. We use here that the search direction $\mathbf{p}_k$ is a descent direction so that $\nabla f_k^T \mathbf{p}_k$ is negative. The curvature condition requires that $|\phi'(\alpha)| = |\nabla f(\mathbf{x}_k + \alpha \mathbf{p}_k)^T \mathbf{p}_k|$ decrease sufficiently each iteration, as $|\phi'(\alpha)| \leq c_2 |\phi'(0)|$.

The conditions can be expressed equivalently as

Wolfe Conditions for $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k)$:

(i) (sufficient decrease) $\phi(\alpha) \leq \phi(0) - c_1 \alpha |\phi'(0)|$, where $c_1 \in (0, 1)$.

(ii) (curvature condition) $|\phi'(\alpha)| \leq c_2 |\phi'(0)|$, where $c_2 \in (c_1, 1)$.

We now prove if the function is smooth and bounded below there always exist steps α which satisfy the Wolfe Conditions.

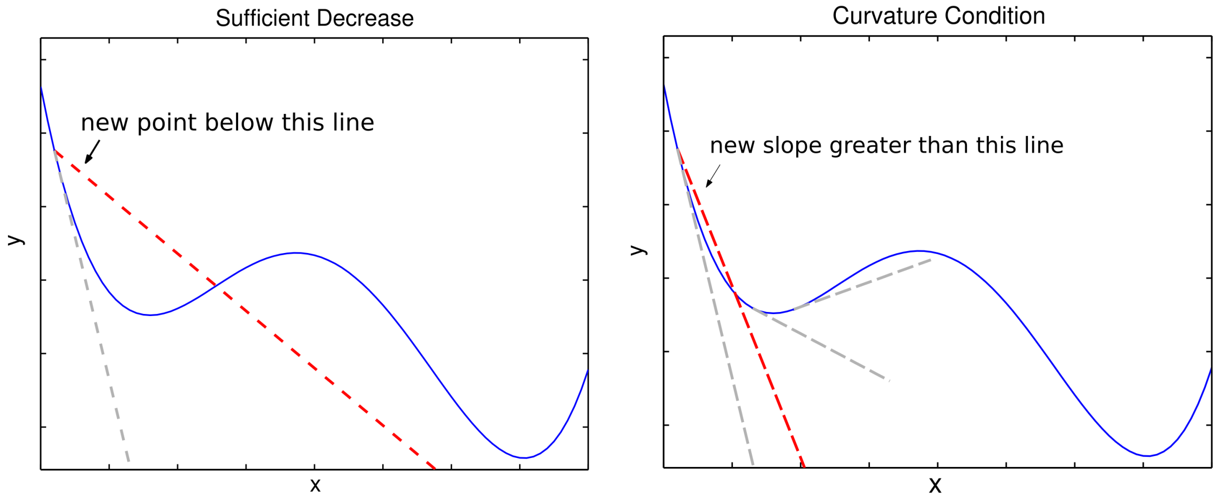

Figure 1: Sufficient Decrease Condition *(left)* and Curvature Condition *(right)*.

Lemma: Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, $\mathbf{p}_k$ be a descent direction, and $\mathcal{R}(\mathbf{p}_k) = \{\mathbf{x}_k + \alpha \mathbf{p}_k \mid \alpha > 0\}$ be the ray in direction $\mathbf{p}_k$. If the function f is bounded below on the ray $\mathcal{R}$ then there exists $\alpha^{(1)}, \alpha^{(2)}$ so that steps $\alpha \in [\alpha^{(1)}, \alpha^{(2)}]$ satisfy the Wolfe Conditions.

Proof: Consider the line $\ell(\alpha) = f(\mathbf{x}_k) + \alpha c_1 \nabla f_k^T \mathbf{p}_k$. Since f is bounded below along the ray we have for all $\alpha > 0$ that $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k) \geq C_0$ for some finite C_0. Since $0 < c_1 < 1$ the line $\ell(\alpha)$ starts out above the graph of $\phi(\alpha)$ for $\alpha > 0$, see Figure 1. Since the line $\ell(\alpha) \to -\infty$ as $\alpha \to \infty$ there is some α^- where $\ell(\alpha^-) < C_0$.

By the intermediate value theorem $\ell(\alpha) - \phi(\alpha)$ must be zero at some location α'. This shows the line $\ell(\alpha)$ must intersect the graph of $\phi(\alpha)$ at least once for $\alpha > 0$. Let $\alpha' > 0$ be the smallest such value, so that $\phi(\alpha') = \ell(\alpha')$ which gives $f(\mathbf{x}_k + \alpha' \mathbf{p}_k) = f(\mathbf{x}_k) + \alpha' c_1 \nabla f_k^T \mathbf{p}_k$. The sufficient decrease condition (i) then holds for $\alpha < \alpha'$.

By the differentiability of the function we have from the mean-value theorem that there exists $\alpha'' \in (0, \alpha')$ such that $f(\mathbf{x}_k+\alpha'\mathbf{p}_k)-f(\mathbf{x}_k) = \alpha'\nabla f(\mathbf{x}_k+\alpha''\mathbf{p}_k)^T\mathbf{p}_k$. Now by substituting for $f(\mathbf{x}_k+\alpha'\mathbf{p}_k)$ from above, we have $\nabla f(\mathbf{x}_k+\alpha''\mathbf{p}_k)^T\mathbf{p}_k = c_1\nabla f_k^T\mathbf{p}_k > c_2\nabla f_k^T\mathbf{p}_k$, since $c_1 < c_2$ and $\nabla f_k^T\mathbf{p}_k < 0$. The α'' then satisfies the curvature condition (ii).

Since $c_1 < c_2$ and $\alpha'' < \alpha'$ we further have there exists a $\delta > 0$ so that both the curvature condition and sufficient decrease condition holds in a neighborhood of α'' given by $\alpha \in [\alpha''-\delta, \alpha''+\delta] = [\alpha^{(1)}, \alpha^{(2)}]$ with $\alpha^{(1)} = \alpha''-\delta > 0$ and $\alpha^{(2)} = \alpha''+\delta < \alpha'$. ■

Convergence of Line Search Optimization Methods

We now consider what additional conditions are required for line search methods to converge. The following is a useful result concerning the role of the search directions $\mathbf{p}_k$.

Theorem (Zoutendijk's Condition): Assume $\{\mathbf{x}_k\}$ is generated by a line search algorithm satisfying the Wolfe Conditions (i),(ii). Suppose that $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is bounded below and has a Lipschitz continuous gradient ∇f,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x}-\mathbf{y}\|. \tag{4}$$

We then have

$$\sum_{k=0}^{\infty} \cos^2(\theta_k)\|\nabla f_k\|^2 < \infty, \tag{5}$$

where

$$\cos(\theta_k) = \frac{-\nabla f_k^T\mathbf{p}_k}{\|\nabla f_k\|\|\mathbf{p}_k\|}. \tag{6}$$

Remark: The θ_k gives the angle between the steepest descent direction $-\nabla f_k$ and the search direction $\mathbf{p}_k$.

Proof: From the curvature condition (ii), we have

$$\left(\nabla f_{k+1} - \nabla f_k\right)^T \mathbf{p}_k \geq (c_2 - 1)\,\nabla f_k^T\mathbf{p}_k. \tag{7}$$

The Lipschitz continuity gives

$$\left(\nabla f_{k+1} - \nabla f_k\right)^T \mathbf{p}_k \leq \alpha_k L\|\mathbf{p}_k\|^2. \tag{8}$$

Combining both relations yields

$$\alpha_k \geq \frac{c_2-1}{L}\frac{\nabla f_k^T\mathbf{p}_k}{\|\mathbf{p}_k\|^2}. \tag{9}$$

By substituting this into (i),

$$f_{k+1} \leq f_k - c_1\frac{1-c_2}{L}\frac{\left(\nabla f_k^T\mathbf{p}_k\right)^2}{\|\mathbf{p}_k\|^2}. \tag{10}$$

From the definition of $\cos(\theta_k)$,

$$f_{k+1} \leq f_k - c\cos^2(\theta_k)\|\nabla f_k\|^2, \tag{11}$$

where $c = \frac{c_1(1-c_2)}{L}$. By summing all indices less than k, we have $f_{k+1} \leq f_0 - c\sum_{j=0}^{k} \cos^2(\theta_j)\|\nabla f_j\|^2$. Since $f(x)$ is bounded below, $f(x) \geq C_0$, we must have $f_0 - c\sum_{j=0}^{k} \cos^2(\theta_j)\|\nabla f_j\|^2 \geq C_0$ which requires

$$\sum_{j=0}^{\infty} \cos^2(\theta_j)\|\nabla f_j\|^2 < \infty. \tag{12}$$

■

Remark: A useful consequence of this theorem follows from the properties of convergent series. In particular, a series $\sum_{j=0}^{\infty} a_j$ converges only if the summands decay to zero, $a_j \to 0$. The theorem requires $\cos^2(\theta_k)\|\nabla f_k\|^2 \to 0$, as $k \to \infty$. As a result, if we construct a line search method with the search directions $\mathbf{p}_k$ having the uniform bound $\cos(\theta_k) > \delta > 0$ for some δ, the theorem requires that

$$\lim_{k\to\infty} \|\nabla f_k\| = 0. \tag{13}$$

This corresponds with choosing search directions $\mathbf{p}_k$ so they are uniformly bounded away from being orthogonal to the descent direction $-\nabla f_k$.

For such line search methods, if the sequence $\{\mathbf{x}_k\}$ remains bounded then there is a limit point $\mathbf{x}^*$ that is also a critical point $\nabla f(\mathbf{x}^*) = 0$. Given the sufficient decrease condition (i), this point would be a good candidate for a local minimizer. However, we caution that without further assumptions about f and analysis we can not yet rule out a saddle point or other structures at $\mathbf{x}^*$.

To further characterize the convergence, we say that a sequence $\{b_k\}$ *converges to b^* at rate p* if

$$\lim_{k\to\infty} \frac{|b_{k+1} - b^*|}{|b_k - b^*|^p} \leq c. \tag{14}$$

In the case $p = 1$, we further require $c < 1$. In the other cases with $p > 1$, the c can be any finite value. We say the method has *linear convergence* if $p = 1$ and *quadratic convergence* if $p = 2$. We now show how these results can be used to establish convergence of two widely-used optimization methods (i) Gradient Descent and (ii) Newton's Method.

Convergence of Gradient Descent

In this case, we use $\mathbf{p}_k = -\mathbf{g}_k = -\nabla f_k$ for each step. This gives $\cos(\theta_k) = 1$ for all iterations, so by the theorem $\lim_{k\to\infty} \|\nabla f_k\| = 0$. Provided the iterations $\{\mathbf{x}_k\}$ remain within a bounded set, the methods will have a limit point that is a critical point of f.

There is more we can say about the method of Gradient Descent if we make additional assumptions about f. To help demonstrate the concepts, we consider quadratic objective functions for which we can analyze readily the iterations,

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} - \mathbf{b}^T\mathbf{x}. \tag{15}$$

For these objectives, we take $Q = Q^T$ to be positive definite. From $\nabla f(\mathbf{x}^*) = Q\mathbf{x}^* - \mathbf{b} = 0$, we obtain that the minimizer satisfies the linear equation $Q\mathbf{x}^* = \mathbf{b}$.

Without loss of generality, we will consider the case with $\mathbf{b} = 0$. This form can always be obtained by using the change of variable $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}^* = \mathbf{x} - Q^{-1}\mathbf{b}$ and objective $\tilde{f}(\tilde{\mathbf{x}}) = \frac{1}{2}\tilde{\mathbf{x}}^T Q\tilde{\mathbf{x}} = f(\mathbf{x}) - \frac{1}{2}\mathbf{x}^{*,T}Q\mathbf{x}^*$. We can compute explicitly the step-length α which minimizes $\phi(\alpha)$. This can be expressed as

$$\phi(\alpha) = f(\mathbf{x}_k - \alpha\mathbf{g}_k) = \frac{1}{2}\left(\mathbf{x}_k - \alpha\mathbf{g}_k\right)^T Q\left(\mathbf{x}_k - \alpha\mathbf{g}_k\right). \tag{16}$$

The minimizer $\alpha > 0$ satisfies

$$\phi'(\alpha) = -\nabla f(\mathbf{x}_k - \alpha\mathbf{g}_k)^T\mathbf{g}_k = -\left(\mathbf{x}_k - \alpha\mathbf{g}_k\right)^T Q\mathbf{g}_k = 0. \tag{17}$$

By setting $\phi'(\alpha) = 0$, we obtain

$$\alpha_k = \frac{\mathbf{x}_k^T Q\mathbf{g}_k}{\mathbf{g}_k^T Q\mathbf{g}_k} = \frac{\mathbf{g}_k^T\mathbf{g}_k}{\mathbf{g}_k^T Q\mathbf{g}_k}, \tag{18}$$

where we use that $\mathbf{g}_k = Q\mathbf{x}_k$. By using $\mathbf{g}_k = \nabla f_k$, we can also express this as

$$\alpha_k = \frac{\nabla f_k^T\nabla f_k}{\nabla f_k^T Q\nabla f_k}. \tag{19}$$

This gives the line search iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\frac{\nabla f_k^T\nabla f_k}{\nabla f_k^T Q\nabla f_k}\right)\nabla f_k. \tag{20}$$

From the expressions above it can be shown that the function is reduced toward the minimizer each iteration by the factor $(1-\gamma) < 1$, where

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) = (1-\gamma)\left(f(\mathbf{x}_k) - f(\mathbf{x}^*)\right), \tag{21}$$

and

$$\gamma = \frac{\left(\nabla f_k^T\nabla f_k\right)^2}{\left(\nabla f_k^T Q\nabla f_k\right)\left(\nabla f_k^T Q^{-1}\nabla f_k\right)}. \tag{22}$$

This indicates the rate of convergence in f to the minimizer f^* is first order, $p = 1$.

We can also express this in terms of the eigenvalues of Q as

$$\left(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)\right) \leq \left[\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right]^2\left(f(\mathbf{x}_k) - f(\mathbf{x}^*)\right), \tag{23}$$

where $0 \leq \lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$ are eigenvalues of Q. This is proven in [3]. This can be expressed in terms of the condition number $\kappa = \kappa(Q) = \lambda_n/\lambda_1$ as

$$(f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*)) \leq \left[\frac{1-\kappa^{-1}}{1+\kappa^{-1}}\right]^2 (f(\mathbf{x}_k) - f(\mathbf{x}^*)) \,. \tag{24}$$

As the condition number $\kappa \to \infty$, we have the factor $(1-\gamma) \to 1$. This shows the performance of Gradient Descent can become poor when the condition number of Q becomes large.

While our analysis above was for the quadratic case, many of the results are also indicative of the behaviors of more general non-linear objective functions. For smooth objective functions, we expect the behaviors close to a minimizer $\mathbf{x}^*$ would behave similar to the quadratic case since we can perform second-order Taylor expansions which yield quadratics where $Q = \nabla^2 f$ is the Hessian. The results indicate the performance of the Gradient Descent method will depend on the condition number of the Hessian. When the Hessian is non-zero, the analysis also indicates we would expect a convergence rate of first-order $p = 1$. These results indicate that we may be able to improve the convergence by making further use of the second-order Hessian of f and surrogate models for the objective function.

Convergence of Newton's Method

We now consider some other choices for the search directions based on second-order information of f. We use search directions based on the Hessian $\nabla^2 f_k$ and gradient of the form

$$\mathbf{p}_k = -\nabla^2 f_k^{-1} \mathbf{g}_k, \tag{25}$$

where $\mathbf{g}_k = \nabla f_k$. This is motivated by a local quadratic model approximating the objective function $q_k(\mathbf{x} - \mathbf{x}_k) = f(\mathbf{x}_k) + \mathbf{g}_k^T(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \nabla^2 f_k (\mathbf{x} - \mathbf{x}_k)$. If we minimize $q_k(\tilde{\mathbf{x}})$ by taking $\nabla_{\tilde{\mathbf{x}}} q_k = 0$ we obtain $\tilde{\mathbf{x}}' = -\nabla^2 f_k^{-1} \mathbf{g}_k$. Since $\tilde{\mathbf{x}}' = \mathbf{x}' - \mathbf{x}_k$ this suggests the iteration $\mathbf{x}' = \mathbf{x}_k + \tilde{\mathbf{x}}' = \mathbf{x}_k - \nabla^2 f_k^{-1} \mathbf{g}_k = \mathbf{x}_k + \alpha \mathbf{p}_k$. This corresponds to the search direction $\mathbf{p}_k$ above with step size $\alpha = 1$. When the Hessian positive definite we see this will yield a descent direction, since $-\mathbf{g}_k^T \mathbf{p}_k = \mathbf{g}_k^T \nabla^2 f_k^{-1} \mathbf{g}_k$.

In the case of quadratic objective functions with $Q = Q^T$ positive semi-definite, we see that Newton's Method would converge in one iteration. The key idea to obtain Newton's Method was to use q_k as a surrogate model for the objective function and to perform minimization of q_k. Other fitting and modeling approaches also could be used to obtain iterations.

When using Newton Methods in practice, one needs to be careful since the Hessian $\nabla^2 f_k$ may not always be positive definite. Whether $\mathbf{p}_k$ is a descent direction $-\mathbf{g}_k^T \mathbf{p}_k = \mathbf{g}_k^T \nabla^2 f_k^{-1} \mathbf{g}_k$ will depend in this case on the direction $\mathbf{g}_k$. This can be problematic and iterations can exhibit erratic iterations in this setting. Fortunately, provided Newton's Method starts sufficiently close to a minimizer it can be shown to converge. This is often accomplished by using a combination with other methods, such as Gradient Descent, to perform initial optimizations to get close to a minimizer and then using Newton's Method to rapidly obtain more accurate approximations.

We now show the circumstances under which Newton's Method converges with a quadratic rate, $p = 2$. Consider

$$\begin{aligned} \mathbf{x}_k + \mathbf{p}_k - \mathbf{x}^* &= \mathbf{x}_k - \mathbf{x}^* - \nabla^2 f_k^{-1} \mathbf{g}_k \\ &= \nabla^2 f_k^{-1} \left[\nabla^2 f_k \cdot (\mathbf{x}_k - \mathbf{x}^*) - (\mathbf{g}_k - \mathbf{g}_*) \right] \end{aligned} \tag{26}$$

where $\mathbf{g}_* = \nabla f(\mathbf{x}^*) = 0$. We can express the gradient as

$$\mathbf{g}_k - \mathbf{g}_* = \int_0^1 \nabla^2 f(\mathbf{x}_k + t(\mathbf{x}^* - \mathbf{x}_k)) \cdot (\mathbf{x}_k - \mathbf{x}^*) dt, \tag{27}$$

we have

$$\begin{aligned} \left\| \nabla^2 f(\mathbf{x}_k) \cdot (\mathbf{x}_k - \mathbf{x}^*) - (\mathbf{g}_k - \mathbf{g}_*)) \right\| &= \left\| \int_0^1 \left[\nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k + t(\mathbf{x}^* - \mathbf{x}_k)) \right] \cdot (\mathbf{x}_k - \mathbf{x}^*) dt \right\| \\ &\leq \int_0^1 \left\| \nabla^2 f(\mathbf{x}_k) - \nabla^2 f(\mathbf{x}_k + t(\mathbf{x}^* - \mathbf{x}_k)) \right\| \cdot \left\| \mathbf{x}_k - \mathbf{x}^* \right\| dt \\ &\leq \| \mathbf{x}_k - \mathbf{x}^* \|^2 \int_0^1 L t dt. \end{aligned} \tag{28}$$

The L is the Lipschitz constant for $\nabla^2 f(\mathbf{x})$ for $\mathbf{x}$ near $\mathbf{x}^*$. This gives

$$\|\mathbf{x}_k + \mathbf{p}_k - \mathbf{x}^*\| \leq L \|\nabla^2 f(\mathbf{x}^*)^{-1}\| \|\mathbf{x}_k - \mathbf{x}^*\|^2 \tag{29}$$

$$\Rightarrow \|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq L \|\nabla^2 f(\mathbf{x}^*)^{-1}\| \|\mathbf{x}_k - \mathbf{x}^*\|^2 \tag{30}$$

$$\Rightarrow \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \leq L \|\nabla^2 f(\mathbf{x}^*)^{-1}\|. \tag{31}$$

This shows that $\mathbf{x}_k \to \mathbf{x}^*$ has a quadratic rate of convergence ($p = 2$). As mentioned above, Newton's Method is typically used in conjunction with other optimization methods. A widely employed strategy is to perform initial iterations using a more robust alternative method, such as Gradient Descent, to obtain a rough approximate solution sufficiently close to a minimizer. The Newton's Methods are then employed to rapidly improve the result to obtain a more refined accurate solution that benefits from the quadratic convergence.

Line Search Algorithms

We now discuss a few algorithms for finding step-lengths α satisfying the Wolfe Conditions. The basic strategy is to use interpolation and a bisection search by determining which half of an interval contains points satisfying the Wolfe conditions. We state some pseudo-code for a few algorithms. Additional discussions and more details can be found in [1].

Algorithm: (LineSearch)

```
input: $\mathbf{x}_k$, $\mathbf{p}_k$, $\alpha_{\mathrm{max}}$.
output: $\alpha_*$.

$\alpha_0 \leftarrow 0$, and specify $\alpha_1 > 0$ and $\alpha_{\mathrm{max}}$;
$i \leftarrow 1$
repeat:
    evaluate $\phi(\alpha_i)$;
    if $\phi(\alpha_i) > \phi(0) + c_1\alpha_i\phi'(0)$ or $\phi(\alpha_i) \geq \phi(\alpha_{i-1})$ and $i > 1$.
    $\alpha_* \leftarrow$ zoom $(\alpha_{i-1}, \alpha_i)$ and stop.
    evaluate $\phi'(\alpha_i)$;
    if $|\phi'(\alpha_i)| \leq -c_2\phi'(0)$
        set $\alpha_* \leftarrow \alpha_i$ and stop;
    if $\phi'(\alpha_i) \geq 0$
        set $\alpha_* \leftarrow$ zoom$(\alpha_i, \alpha_{i-1})$ and stop;
    choose $\alpha_{i+1} \in (\alpha_i, \alpha_{\mathrm{max}})$
    $i \leftarrow i + 1$;
end (repeat)
```

Algorithm: (Zoom)

```
input: $\alpha_{\mathrm{lo}}$,$\alpha_{\mathrm{hi}}$.
output: $\alpha_*$.

repeat
    interpolate (using quadratic, cubic, or bisection) to find a trial step length $\alpha_j$
    between $\alpha_{\mathrm{lo}}$ and $\alpha_{\mathrm{hi}}$
    evaluate $\phi(\alpha_j)$;
    if $\phi(\alpha_j) > \phi(0) + c_1\alpha_j\phi'(0)$ or $\phi(\alpha_j) \geq \phi(\alpha_{\mathrm{lo}})$
        $\alpha_{\mathrm{hi}} \leftarrow \alpha_j$;
    else
        evaluate $\phi'(\alpha_j)$;
            if $|\phi(\alpha_j)| \leq -c_2\phi'(0)$
                set $\alpha_* \leftarrow \alpha_j$ and stop;
            if $\phi'(\alpha_j)(\alpha_{\mathrm{hi}} - \alpha_{\mathrm{lo}}) \geq 0$
                $\alpha_{\mathrm{hi}} \leftarrow \alpha_{\mathrm{lo}}$;
            $\alpha_{\mathrm{lo}} \leftarrow \alpha_j$;
end (repeat)
```

Solving Nonlinear Unconstrained Optimization Problems

Consider unconstrained optimization problems of the form

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}). \tag{32}$$

One line search algorithm to approximate solutions of this problem is the following.

Algorithm: (An Unconstrained Line Search Optimization)

```
input: $\mathbf{x}_0$, $\alpha_{\max}, \epsilon$.
output: $\mathbf{x}^*$.

$k \leftarrow 0$
$\mathbf{x}_k \leftarrow \mathbf{x}_0$
evaluate $\nabla f_k \leftarrow \nabla f(\mathbf{x}_k)$;
repeat
    $\mathbf{p}_k = -B_k^{-1}\nabla f_k$;
    $\alpha_k \leftarrow \mathrm{LineSearch}(\mathbf{x}_k, \mathbf{p}_k, \alpha_{\max})$;
    $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k$;
    evaluate $\nabla f_k \leftarrow \nabla f(\mathbf{x}_k)$;
    if $\|\nabla f_k\| < \epsilon$ then
        $\mathbf{x}^* \leftarrow \mathbf{x}_k$ and stop;
end (repeat)
```

Solving Nonlinear Constrained Optimization Problems
Consider constrained optimization problems of the form

$$\begin{aligned} \mathbf{x}^* = \arg\min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject:} \quad & c_i(\mathbf{x}) = 0, \ i \in \mathcal{E} \\ & c_i(\mathbf{x}) \geq 0, \ i \in \mathcal{I}. \end{aligned} \tag{33}$$

The $\mathcal{E}$ gives the indices for the equality constraints and $\mathcal{I}$ the indices for the inequality constraints.

A line search algorithm can be constructed to handle these constraints. This is formulated by building on the unconstrained optimization problem and incorporating the constraints through penalty terms. The strategy is to then solve each of the unconstrained problems with a penalty parameter μ_j up to a tolerance of ϵ_j. By repeatedly solving a family of these problems by successively refining ϵ_j, we can obtain an approximate solution to the constrained problem. For each problem, we use as the starting point the solution $\mathbf{x}_j^*$ of the previously solved problem. We successively reduce the values of μ_j and ϵ_j, to obtain a sequence of solutions aiming for $\mathbf{x}_j \to \mathbf{x}^*$. In practice, one must take some care on how μ_j and ϵ_j are reduced each iteration to ensure convergence and efficient use of computational resources. Additional discuss on this and other details can be found in the references. One approach to approximating solutions of the constrained optimization problem is the following.

Algorithm: (A Constrained Line Search Optimization)

```
input: $\mathbf{x}_0$, $\alpha_{\max}, \epsilon_*, \delta_*$.
output: $\mathbf{x}^*$.

$j \leftarrow 0$
$\mathbf{x}_j^* \leftarrow \mathbf{x}_0$
$\mu_j \leftarrow \mu_0$
repeat 1
```

```
    let $F(\mathbf{x}, \mu_j) = f(\mathbf{x}) + \frac{1}{2\mu_j} \sum_{i \in \mathcal{E}} c_i^2(\mathbf{x}) - \mu_j \sum_{i \in \mathcal{I}} \log\left(c_i(\mathbf{x})\right)$
    $\mathbf{x}_j^* \leftarrow \text{UnconstrainedOptimization}(\mathbf{x}_j^*, \alpha_{\max}, \epsilon_j)$
    evaluate $\nabla F_j \leftarrow \nabla_{\mathbf{x}} F(\mathbf{x}_j^*, \mu_j)$;
    if $\|\nabla F_j\| < \epsilon_*$ and $\mu_j < \delta_*$ then $\mathbf{x}^* \leftarrow \mathbf{x}_j^*$ and stop;
    $j \leftarrow j + 1$
    reduce the value of $\mu_j$
    reduce the value of $\epsilon_j$
end (repeat 1)
```

Conclusion
These notes are meant to serve as a brief introduction. Non-linear optimization is a broad field with many applications in the sciences, engineering, statistics, economics, machine learning, AI, and other disciplines. Additional discussions and details on these algorithms also can be found in the references.

For comments or errors concerning these notes, please email: `atzberg@gmail.com`.

References

[1] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[2] Edwin KP Chong, Wu-Sheng Lu, and Stanislaw H Zak. *An Introduction to Optimization: With Applications to Machine Learning*. John Wiley & Sons, 2023.

[3] David G Luenberger, Yinyu Ye, et al. *Linear and nonlinear programming*. Vol. 2. Springer, 1984.