Chapter 3

Search directions

Our typical question in this chapter is that of choosing a direction $\mathbf{p}$ at a point $\mathbf{x}$. The direction $\mathbf{p}$ is a *descent direction* (or a *downhill* direction) at a point $\mathbf{x}$ if

$$\boldsymbol{\nabla} f(\mathbf{x})^T \mathbf{p} < 0. \tag{3.1}$$

Since by Taylor's Theorem

$$f(\mathbf{x} + \alpha\mathbf{p}) = f(\mathbf{x}) + \alpha \boldsymbol{\nabla} f(\mathbf{x})^T \mathbf{p} + \mathcal{O}(\alpha^2 \|\mathbf{p}\|^2),$$

we know that if α is chosen small enough, $f(\mathbf{x}+\alpha\mathbf{p}) < f(\mathbf{x})$. Usually, though, the size of the step $\alpha\|\mathbf{p}\|$ is not taken very small, or else we cannot expect a significant decrease in $f(\mathbf{x} + \alpha\mathbf{p})$ as compared to $f(\mathbf{x})$.

Throughout these notes we use the shorthand notation

$$f_k = f(\mathbf{x}_k),\ \boldsymbol{\nabla} f_k = \boldsymbol{\nabla} f(\mathbf{x}_k),\ H_k = \nabla^2 f_k = \nabla^2 f(\mathbf{x}_k). \tag{3.2}$$

3.1 Steepest descent, Newton and quasi-Newton

At a current iterate $\mathbf{x}_k$ let us consider search directions of the form

$$\mathbf{p}_k = -B_k^{-1} \boldsymbol{\nabla} f_k. \tag{3.3}$$

Clearly, $\mathbf{p}_k$ is guaranteed to be a descent direction (i.e. satisfy (3.1)) if B_k is symmetric positive definite.

Steepest descent

The inner product of two vectors $\mathbf{p}$ and $\mathbf{d}$ satisfies,

$$\mathbf{p}^T \mathbf{d} = \|\mathbf{p}\| \|\mathbf{d}\| \cos\theta,$$

where θ is the angle between the vectors. So, fixing the length of these vectors and considering their relative directions, the minimum of this inner product is obtained when

$$\cos\theta = -1, \quad \text{i.e.} \quad \theta = -\pi.$$

For $\mathbf{d} = \boldsymbol{\nabla} f_k$ and $\mathbf{p} = -B_k^{-1}\boldsymbol{\nabla} f_k$ as in (3.3), this minimum corresponds to choosing $B_k = I$. Thus, the *steepest descent* for a very small step size is obtained by choosing

$$\mathbf{p}_k = -\boldsymbol{\nabla} f_k. \tag{3.4}$$

This is certainly a very simple choice of direction.

The steepest descent direction appears as an ingredient in typical *trust region* methods and *Levenberg-Marquardt* methods. If we want to use it in stand-alone mode then we must incorporate it with some *line search* technique. An idealized line search leads to an idealized *greedy algorithm* sometimes called *optimal gradient algorithm*: At the kth iteration, given current iterate $\mathbf{x}_k$,

$$\begin{aligned}
&\mathbf{p}_k = -\boldsymbol{\nabla} f_k;\\
&\text{Find } \alpha = \alpha_k \text{ s.t. } f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) = \min_{\alpha \geq 0} f(\mathbf{x}_k + \alpha \mathbf{p}_k);\\
&\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k.
\end{aligned}$$

The convergence rate of this method is linear.

The steepest descent direction $\mathbf{p}_k$ can be seen to be orthogonal to the *level set* at $\mathbf{x}_k$:

$$\{\mathbf{y} \mid f(\mathbf{y}) = f(\mathbf{x}_k)\}.$$

But a problem with this direction arises when the Hessian $H_k = \nabla^2 f_k$ has widely varying eigenvalues (i.e., a large condition number). For example, consider

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T H \mathbf{x},$$

H positive definite and constant. In $\mathbb{R}^2$ the level sets are ellipses with axes λ_1 and λ_2. If $\lambda_1 \gg \lambda_2$ then the level sets are very elongated. Consecutive steepest descent directions then tend to zig-zag. Indeed it can be shown that for the idealized method with exact line search and a quadratic objective function,

$$f(\mathbf{x}_{k+1}) \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^2 f_k, \quad \kappa = cond(H) = \frac{\lambda_1}{\lambda_2}.$$

This bound is unfortunately often realistic (not overly pessimistic). If $\kappa \gg 1$ then $\left(\frac{\kappa-1}{\kappa+1}\right)^2 \approx 1 - \frac{4}{\kappa}$ and convergence can be very slow indeed.

Example 3.1 (Aoki, pp. 106-107.)
Consider the function

$$f(\mathbf{x}) = \frac{1}{2}(x_1^2 - x_2)^2 + \frac{1}{2}(x_1 - 1)^2$$

for $\mathbf{x} = (x_1, x_2)^T$.

The gradient and Hessian are given by

$$\boldsymbol{\nabla} f(\mathbf{x}) = \begin{pmatrix} 2x_1(x_1^2 - x_2) + x_1 - 1 \\ x_2 - x_1^2 \end{pmatrix}, \quad \nabla^2 f(\mathbf{x}) = \begin{pmatrix} 6x_1^2 - 2x_2 + 1 & -2x_1 \\ -2x_1 & 1 \end{pmatrix}.$$

By inspection, there is a minimum at $\mathbf{x}^* = (1, 1)^T$, where $f(\mathbf{x}^*) = 0$.

Where is $\nabla^2 f(\mathbf{x})$ positive definite? Any real symmetric 2×2 matrix $\begin{pmatrix} a & b \\ b & d \end{pmatrix}$ is positive definite iff $a > 0$, $d > 0$ and $ad > b^2$. Here this translates into the condition

$$2(x_1^2 - x_2) + 1 > 0.$$

At $\mathbf{x}^*$, in particular, the Hessian is positive definite. In fact, its eigenvalues equal $3 \pm \frac{1}{2}\sqrt{32}$. Thus, $\frac{\lambda_1}{\lambda_2} \approx 34$. This causes a relatively slow convergence for the optimal gradient (steepest descent) method.

Starting at $\mathbf{x}_0 = (0, 0)^T$ we get

k	$2f_k$
0	1
1	.289
2	.176
3	.112
4	.087

It goes on, slowly. (Think of wanting $FTOL = 1.e - 8$, for instance!) See Figure 3.1

♦

Newton

Let us use Taylor's expansion to model f locally by a quadratic:

$$f(\mathbf{x}_k + \mathbf{p}) \approx f_k + \mathbf{p}^T \boldsymbol{\nabla} f_k + \frac{1}{2}\mathbf{p}^T H_k \mathbf{p} =: m_k(\mathbf{p}). \tag{3.5a}$$

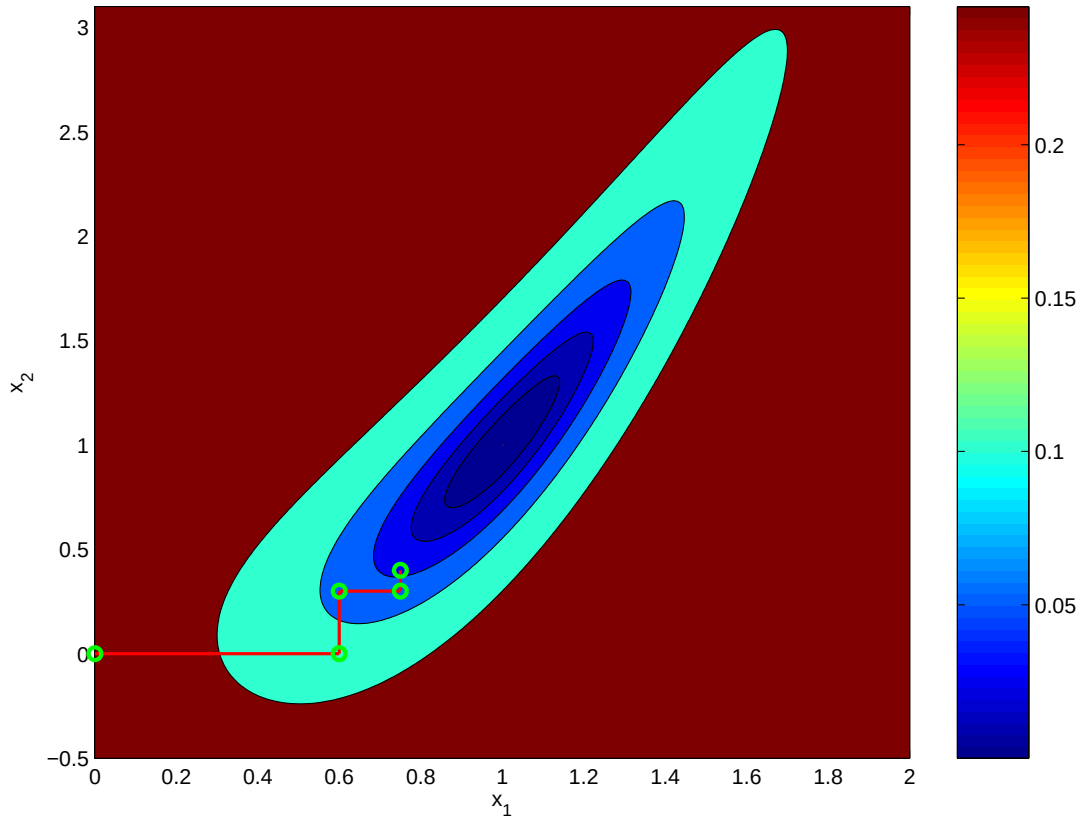

Figure 3.1: Level sets and steepest descent for Example 3.1. With exact line search the steepest descent directions are orthogonal to each other and zig-zagging.

If H_k is positive definite then the minimum of $m_k(\mathbf{p})$ is at its critical point, $\boldsymbol{\nabla} f_k + H_k \mathbf{p}_k = \mathbf{0}$. Thus,

$$\mathbf{p}_k = -H_k^{-1} \boldsymbol{\nabla} f_k. \tag{3.5b}$$

This defines Newton's direction. Note that it is guaranteed to be a descent direction *if* the Hessian H_k is positive definite at the current iterate $\mathbf{x}_k$.

Unlike for the steepest descent direction there is a natural step size here, $\alpha_k = 1$. The pure Newton's method reads:

$$\begin{aligned} &\text{Solve } H_k \mathbf{p}_k = -\boldsymbol{\nabla} f_k; \\ &\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k. \end{aligned}$$

But with Newton's method, too, line search is possible and may be useful. We return to this later.

Note that the objective function f itself is not involved in the pure Newton method: this is just the usual Newton's method for solving a nonlinear system of equations

$$\boldsymbol{\nabla} f(\mathbf{x}) = \mathbf{0}.$$

Quasi-Newton

The Newton direction (3.5b) has two drawbacks. One is that the computed direction $\mathbf{p}_k$ (if it exists) is not necessarily a descent direction unless $\nabla^2 f(\mathbf{x}_k)$ is positive definite. Another is that an explicit use of second derivative information is made: this may be demanding of a user (who is asked to supply the Hessian function), hard to evaluate, expensive to evaluate, expensive to invert... all depending on the application, of course.

Thus, we look for B_k which is

1. symmetric positive definite,
2. easily computable and invertible,
3. approximates the action of $H_k = \nabla^2 f(\mathbf{x}_k)$ somehow.

To address the quick invertibility issue we could instead consider

$$G_k = B_k^{-1}, \quad \text{i.e. } \mathbf{p}_k = -G_k \boldsymbol{\nabla} f_k,$$

and update G_k for an iteration of the sort

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k G_k \boldsymbol{\nabla} f_k.$$

Note that B_k is symmetric positive definite iff G_k is (the eigenvalues each reciprocate, thus they stay positive). So, we start with $B_0 = I$ or $G_0 = I$, and in the kth iteration obtain B_{k+1} or G_{k+1} from B_k or G_k, respectively, as a quick update.

Derivation requirements

1. By Taylor's expansion, see (2.4b),

$$H_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k) \approx \boldsymbol{\nabla} f_{k+1} - \boldsymbol{\nabla} f_k.$$

So, letting

$$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k, \quad \mathbf{y}_k = \boldsymbol{\nabla} f_{k+1} - \boldsymbol{\nabla} f_k, \tag{3.6}$$

we require

$$G_{k+1}\mathbf{y}_k = \mathbf{s}_k, \quad (\text{or } B_{k+1}\mathbf{s}_k = \mathbf{y}_k). \tag{3.7}$$

2. Symmetry: require that if G_k is symmetric then so is G_{k+1}.

3. Positive definiteness: require that if G_k is positive definite then so is G_{k+1}.

   To get the above requirement to hold it is clear that we must have

   $$\mathbf{s}_k^T \mathbf{y}_k > 0. \tag{3.8}$$

   This is called the *curvature condition*. Note that this condition imposes a restriction on how we choose $\mathbf{x}_{k+1}$.

4. Use a simple update procedure.

How can we satisfy all these derivation requirements? By some update procedure which is "simple but not too simple". As it turns out, a rank-one update,

$$G_{k+1} = G_k + \mathbf{u}_k \mathbf{u}_k^T$$

is too simple in this sense. So, let's try a *rank-two update*

$$G_{k+1} = G_k + \beta \mathbf{u}_k \mathbf{u}_k^T + \gamma \mathbf{v}_k \mathbf{v}_k^T.$$

Then by the secant equation (3.7) we get

$$G_k \mathbf{y}_k + \beta(\mathbf{u}_k^T \mathbf{y}_k)\mathbf{u}_k + \gamma(\mathbf{v}_k^T \mathbf{y}_k)\mathbf{v}_k = \mathbf{s}_k.$$

Choosing $\mathbf{u}_k = \mathbf{s}_k, \mathbf{v}_k = G_k \mathbf{y}_k$ yields the famous Davidon-Fletcher-Powell (DFP) formula

$$G_{k+1} = G_k - \frac{G_k \mathbf{y}_k \mathbf{y}_k^T G_k}{\mathbf{y}_k^T G_k \mathbf{y}_k} + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}. \tag{3.9}$$

Note, incidentally, that instead of updating G_k we could update B_k – the inversion of a low-rank update is simple by the *Sherman-Morrison-Woodbury formula* (2.6).

The most popular formula in practice today is due to Broyden-Fletcher-Goldfarb-Shanno (BFGS),

$$G_{k+1} = (I - \frac{\mathbf{s}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}) G_k (I - \frac{\mathbf{y}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}) + \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{y}_k^T \mathbf{s}_k}. \tag{3.10}$$

An iteration now looks like

$\mathbf{p}_k = -G_k \boldsymbol{\nabla} f_k$;
Find α_k somehow, e.g. $\alpha_k = 1$;
$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$;
$\mathbf{s}_k = \alpha_k \mathbf{p}_k$;
$\mathbf{y}_k = \boldsymbol{\nabla} f_{k+1} - \boldsymbol{\nabla} f_k$;
G_{k+1} by (3.10)

The BFGS formula can also be given in terms of an update to B_k.

Of course, one may wonder further how people came up with these formulae in the first place, what properties can be proved and how to go about doing this, and so on. However, we cut the discussion short at this point (see Chapter 8 of [1]), and end our exposition with the following notes:

1. The BFGS and DFP updates enjoy all the properties mentioned earlier as requirements (including rapid evaluation of the search direction and a guaranteed positive definite Hessian approximation, hence descent). In addition, convergence near a solution (with step sizes $\alpha_k = 1$) is *superlinear* under similar assumptions to those which yield quadratic convergence for Newton's method.

2. Still, Newton's method is not dead: There are many applications where $\nabla^2 f$ is not only easy to evaluate but is also large and sparse. Whereas sparsity of the Hessian is automatically retained by Newton's method it is not retained by the usual quasi-Newton updates. Special updates (e.g. limited memory techniques) exist, but are in general less effective.

Example 3.2

Let us return to the problem of Example 3.1, and apply a step of the three methods we have seen.

1. Starting at $\mathbf{x}_0 = (0,0)^T$ we have $\boldsymbol{\nabla} f_0 = (-1,0)^T$, $\nabla^2 f_0 = I$, and we also use $G_0 = I$ for the BFGS iteration. Thus, the same first direction

$$\mathbf{p}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

   is obtained for the steepest descent, Newton and BFGS methods.

   The next iterate is obtained as

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{p}_0 = \begin{pmatrix} \alpha_0 \\ 0 \end{pmatrix}$$

   where α_0 is chosen such that $f(\mathbf{x}_1) = \frac{1}{2}(\alpha_0^4 + (\alpha_0 - 1)^2)$ is sufficiently smaller than $f(\mathbf{x}_0) = \frac{1}{2}$.

   Clearly, $\alpha_0 = 1$ does not yield a sufficient decrease in f.[1] On the other hand, for $\alpha_0 = 0.5$ we obtain

$$\mathbf{x}_1 = (0.5,\ 0)^T, \quad f(\mathbf{x}_1) = \frac{5}{32} \approx .156,$$

[1] Even though it turns out that if we let $\alpha_0 = 1$ then the next full Newton step ends in $\mathbf{x}^*$ – but that's luck.

and this does represent sufficient decrease in the value of f. So $\mathbf{x}_1$ becomes the current iterate, common to all three methods.

2. At $\mathbf{x}_1$ we have

$$\boldsymbol{\nabla} f_1 = -\frac{1}{4}\begin{pmatrix}1\\1\end{pmatrix},\ \nabla^2 f_1 = \begin{pmatrix}2.5 & -1\\ -1 & 1\end{pmatrix},\ [\nabla^2 f_1]^{-1} = \frac{1}{3}\begin{pmatrix}2 & 2\\ 2 & 5\end{pmatrix}$$

- The steepest descent direction is

$$\mathbf{p}_1^{sd} = \frac{1}{4}\begin{pmatrix}1\\1\end{pmatrix},$$

and an exact line search along $\mathbf{x}_1 + \alpha\mathbf{p}_1$ gives $\alpha_1^{sd} \approx 1.1921$, $\mathbf{x}_2^{sd} = (.5 + .25\alpha_1^{sd},\ .25\alpha_1^{sd})^T$, and $f(\mathbf{x}_2^{sd}) \approx .0778$.

- The Newton direction is

$$\mathbf{p}_1^{newt} = \begin{pmatrix}1/3\\7/12\end{pmatrix}.$$

Trying the full Newton step $\alpha_1 = 1$ we obtain $\mathbf{x}_2^{newt} = (5/6,\ 7/12)^T$ and $f(\mathbf{x}_2^{newt}) = .0201$, so the Newton direction seems better indeed than steepest descent.

- For the BFGS quasi-Newton update we first calculate

$$\mathbf{s}_0 = \mathbf{x}_1 - \mathbf{x}_0 = \alpha_0\mathbf{p}_0 = (1/2,\ 0)^T$$
$$\mathbf{y}_0 = \boldsymbol{\nabla} f_1 - \boldsymbol{\nabla} f_0 = (3/4,\ -1/4)^T.$$

Thus,

$$\mathbf{s}_0^T\mathbf{y}_0 = \mathbf{y}_0^T\mathbf{s}_0 = 3/8 > 0, \quad \mathbf{s}_0\mathbf{y}_0^T = \begin{pmatrix}3/8 & -1/8\\ 0 & 0\end{pmatrix},$$

$$\mathbf{y}_0\mathbf{s}_0^T = [\mathbf{s}_0\mathbf{y}_0^T]^T = \begin{pmatrix}3/8 & 0\\ -1/8 & 0\end{pmatrix}, \quad \mathbf{s}_0\mathbf{s}_0^T = \begin{pmatrix}1/4 & 0\\ 0 & 0\end{pmatrix}.$$

The BFGS update now yields

$$\begin{aligned} G_1 &= \left(I - \frac{8}{3}\begin{pmatrix}3/8 & -1/8\\ 0 & 0\end{pmatrix}\right) I \left(I - \frac{8}{3}\begin{pmatrix}3/8 & 0\\ -1/8 & 0\end{pmatrix}\right) + \frac{8}{3}\begin{pmatrix}1/4 & 0\\ 0 & 0\end{pmatrix} \\ &= \begin{pmatrix}.7778 & 1/3\\ 1/3 & 1\end{pmatrix}. \end{aligned}$$

Note that G_1 is not particularly close to $[\nabla^2 f_1]^{-1}$.
The obtained direction is

$$\mathbf{p}_1^{bfgs} = -G_1 \boldsymbol{\nabla} f_1 = (.27778,\ 1/3)^T.$$

For the choice $\alpha_1^{bfgs} = 1$ we get $\mathbf{x}_2^{bfgs} = (.7778,\ 1/3)^T$ and $f(\mathbf{x}_2^{bfgs}) = .0616$. This is better than the best we can get with the steepest descent direction, and worse than Newton's.

♦

3.2 Newton's method and nonlinear equations

Newton's method can be defined more generally than just for (2.2). Consider a system of nonlinear equations,

$$\mathbf{r}(\mathbf{x}) = \mathbf{0} \tag{3.11}$$

where $\mathbf{r} : \mathbb{R}^n \to \mathbb{R}^n$, and denote the $n \times n$ Jacobian matrix by J. In the unconstrained optimization case, $J = H$ is symmetric and often positive definite. Here, however, it is generally neither.

By Taylor's series,

$$\mathbf{0} \approx \mathbf{r}(\mathbf{x}_k + \mathbf{p}_k) = \mathbf{r}(\mathbf{x}_k) + J(\mathbf{x}_k)\mathbf{p}_k + \mathcal{O}(\|\mathbf{p}_k\|^2).$$

Newton's basic iteration is, therefore,

$$\begin{aligned} &\text{Solve } J_k \mathbf{p}_k = -\mathbf{r}_k; \\ &\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k. \end{aligned}$$

The material below is contained in Chapter 11 of [1].

Theorem

Suppose that $\mathbf{r}$ is Lipschitz continuously differentiable in an open convex set $\mathcal{D} \subset \mathbb{R}^n$, i.e., there is a constant γ such that

$$\|J(\mathbf{x}) - J(\mathbf{y})\| \le \gamma \|\mathbf{x} - \mathbf{y}\|, \quad \forall\ \mathbf{x}, \mathbf{y} \in \mathcal{D}.$$

Suppose further that there is a nondegenerate root $\mathbf{x}^* \in \mathcal{D}$, i.e.,

$$\mathbf{r}(\mathbf{x}^*) = \mathbf{0}, \quad J(\mathbf{x}^*) \text{ nonsingular}.$$

Then, for $\mathbf{x}_0$ sufficiently close to $\mathbf{x}^*$, Newton's method converges quadratically.

Proof [Given because it is typical]

By Taylor yet again,

$$\mathbf{r}(\mathbf{x}+\mathbf{p}) = \mathbf{r}(\mathbf{x}) + \int_0^1 J(\mathbf{x}+t\mathbf{p})\ \mathbf{p}\ dt.$$

Here,

$$\mathbf{0} = \mathbf{r}(\mathbf{x}^*) = \mathbf{r}_k + \int_0^1 J(\mathbf{x}_k + t(\mathbf{x}^* - \mathbf{x}_k))\ (\mathbf{x}^* - \mathbf{x}_k)\ dt,$$

hence,

$$\mathbf{r}_k + J_k(\mathbf{x}^* - \mathbf{x}_k) = \int_0^1 [J_k - J(\mathbf{x}_k + t(\mathbf{x}^* - \mathbf{x}_k))]\ (\mathbf{x}^* - \mathbf{x}_k)\ dt,$$

yielding by Lipschitz continuity

$$\|\mathbf{r}_k + J_k(\mathbf{x}^* - \mathbf{x}_k)\| \le \frac{\gamma}{2}\|\mathbf{x}^* - \mathbf{x}_k\|^2.$$

Now, by our continuity assumptions there is a ball $\mathcal{B} \subset \mathcal{D}$ around $\mathbf{x}^*$ and a constant β such that

$$\|J^{-1}(\mathbf{x})\| \le \beta, \quad \forall \mathbf{x} \in \mathcal{B}.$$

For $\mathbf{x}_{k+1} = \mathbf{x}_k - J_k^{-1}\mathbf{r}_k$ we have

$$\mathbf{x}^* - \mathbf{x}_{k+1} = \mathbf{x}^* - \mathbf{x}_k + J_k^{-1}\mathbf{r}_k = J_k^{-1}[\mathbf{r}_k + J_k(\mathbf{x}^* - \mathbf{x}_k)],$$

so

$$\|\mathbf{x}^* - \mathbf{x}_{k+1}\| \le \|J_k^{-1}\|\|\mathbf{r}_k + J_k(\mathbf{x}^* - \mathbf{x}_k)\| \le \frac{\beta\gamma}{2}\|\mathbf{x}^* - \mathbf{x}_k\|^2.$$

This establishes quadratic convergence, because we can make the ball $\mathcal{B}$ small enough that, starting with $\mathbf{x}_0 \in \mathcal{B}$, the iterates all stay in that ball. ♦

In practice, of course there is no reason to assume that we'll always know how to choose $\mathbf{x}_0$ close to $\mathbf{x}^*$. Thus, line search or other techniques for global convergence may be called for. If

$$\mathbf{r}(\mathbf{x}) = \boldsymbol{\nabla} f(\mathbf{x})$$

for a function $f(\mathbf{x})$ to be minimized then we would require sufficient decrease in $f(\mathbf{x}_k)$ at each iteration. If there is no $f(\mathbf{x})$ to begin with then we can supply one, e.g.,

$$\min_{\mathbf{x}} \phi(\mathbf{x}) = \frac{1}{2}\|\mathbf{r}(\mathbf{x})\|^2, \tag{3.12}$$

so $\boldsymbol{\nabla}\phi(\mathbf{x}) = J^T\mathbf{r}$, and require sufficient decrease in the *merit function* $\phi(\mathbf{x})$ at each iteration.