

Local approximation of nonlinear functions

Mark S. Gockenbach

1 Introduction

In computational mathematics, nonlinear problems are usually difficult. For example, it is straightforward to solve a system of thousands of equations in thousands of unknowns, provided the equations are linear. (Of course, one needs a computer to carry out the calculations for such a large system. But the calculations themselves are straightforward, and the computer merely does repetitively what is simple to do by hand for small systems.) On the other hand, it is easy to write down a single *nonlinear* equation in one unknown that cannot be solved via a finite number of calculations. Here is an example:

$$xe^x + \log(x) = 1.$$

For this reason, when trying to solve nonlinear equations, it is usually necessary to use *iterative* algorithms, that is, algorithms that produce a sequence converging to (but never reaching exactly) the true solution. Iterative algorithms for nonlinear equations typically employ, at each step, a linear approximation to the nonlinear function defining the problem.

One of the most important results from calculus is that a function attains its maximum or minimum at a point where the derivative of the function is zero. For this reason, optimization is closely related to the problem of solving systems of equations. An optimization problem can be solved exactly by a finite algorithm if the derivative of the function to be maximized or minimized is linear, which is the case if the function is quadratic. Therefore, when a nonquadratic function is to be optimized, it is usually done by an iterative algorithm employing a sequence of quadratic approximations.

The purpose of this paper is to describe linear and quadratic functions in several variables and to show how to construct linear and quadratic approximations to general nonlinear functions. I will begin with functions of one variable, in which case the algebra is simple and graphs make it possible to visualize the approximations. However, the reader should be forewarned that some important concepts about quadratic functions are not present when there is only one independent variable.

2 Functions of one variable

The reader will recall from calculus that a smooth function can be expanded in a Taylor series:

$$f(x) = f(a) + f'(a)(x - a) + \frac{1}{2!}f''(a)(x - a)^2 + \frac{1}{3!}f'''(a)(x - a)^3 + \dots$$

Polynomial approximations to f near $x = a$ can be obtained by truncating this series. For example, the *best local linear approximation* to f near $x = a$ is

$$f(x) \doteq f(a) + f'(a)(x - a) \quad (x \text{ near } a).$$

This is also called the *tangent line approximation* at $x = a$. An example is shown in Figure 1. The *best local quadratic approximation* to f near $x = a$ is

$$f(x) \doteq f(a) + f'(a)(x - a) + \frac{1}{2}f''(a)(x - a)^2 \quad (x \text{ near } a).$$

Higher-degree approximations can be obtained in a similar manner but are rarely used in optimization.

To show how these approximations are used, I will begin by considering the nonlinear equation mentioned in the introduction:

$$xe^x + \log(x) = 1. \quad (1)$$

I will define $f(x) = xe^x + \log(x) - 1$ and try to solve $f(x) = 0$. A simple graph (see Figure 1) shows that the solution is not far from $x = 0.75$. Figure 1 also shows the best local linear approximation to f near $x = 0.75$:

$$f(x) \doteq \ell(x) = 0.30007 + 5.0381(x - 0.75).$$

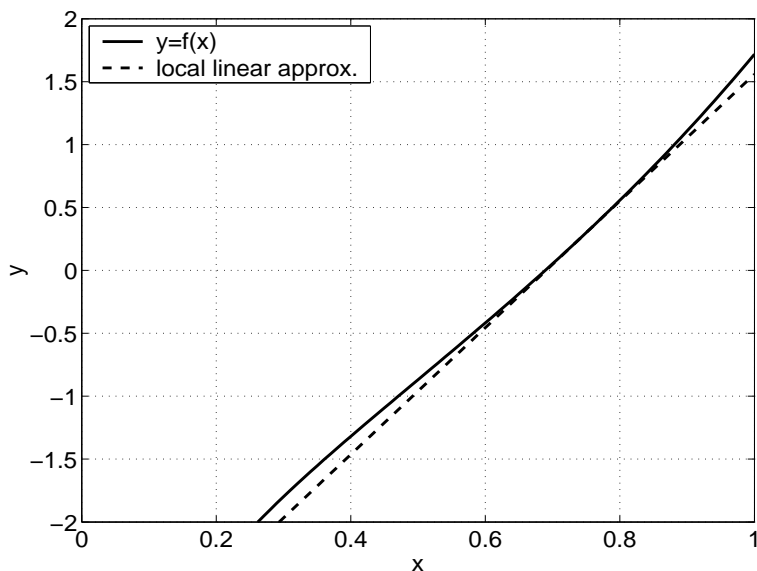


Figure 1: The function $f(x) = xe^x + \log(x) - 1$ and a tangent line approximation.

The graph suggests how to make use of the linear approximation: the graphs of f and ℓ cross the x -axis at nearly the same point, so solving $\ell(x) = 0$ should give an improved approximation of the solution of $f(x) = 0$. Of course, it is easy to solve the linear equation:

$$0.30007 + 5.0381(x - 0.75) = 0 \Rightarrow x \doteq 0.69044.$$

This estimate is quite accurate; to five digits, the solution is $x_1 \doteq 0.68905$.

The repeated use of the best local linear approximation to improve an estimate of the solution of $f(x) = 0$ is called *Newton's method*. Above I performed one iteration of Newton's method; to demonstrate its power, I will perform the second iteration. The tangent-line approximation at x_1 is

$$f(x) \doteq 0.0067209 + 4.8201(x - 0.69044),$$

and

$$0.0067209 + 4.8201(x - 0.69044) = 0 \Rightarrow x \doteq 0.68905.$$

This estimate, $x_2 \doteq 0.68905$, is correct to five decimal digits. As I will explain later, the next estimate x_3 would be correct to about ten digits.

The typical step of Newton's method takes the form

$$f(x_k) + f'(x_k)(x - x_k) = 0 \Rightarrow x = x_k - \frac{f(x_k)}{f'(x_k)}.$$

The following simple formula can thus be applied repeatedly:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

To demonstrate the usefulness of a quadratic approximation, I will consider the following optimization problem: find the value of x that minimizes

$$f(x) = \sin(x) + x^2 + x.$$

As Figure 2 shows, f has a unique minimizer. I will take as my initial estimate $x = 3$. (I can get a better estimate by looking at the graph, but the visual effect is better if I do not start too near.) Figure 2 also shows the best local quadratic approximation to f near $x = 3$:

$$f(x) \doteq q(x) = 12.1411 + 6.0100(x - 3) + 0.92944(x - 3)^2.$$

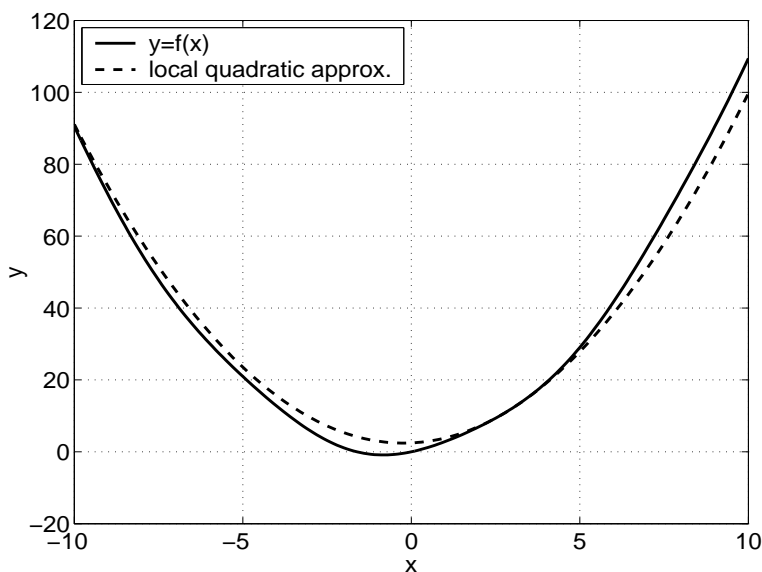


Figure 2: The function $f(x) = \sin(x) + x^2 + x$ and a quadratic approximation.

Minimizing q should provide a good estimate of the minimizer of f . Of course, the minimizer of q is found by setting $q'(x)$ to zero and solving for x :

$$q'(x) = 0 \Rightarrow x \doteq -0.23313.$$

For the sake of illustration, I will give the next step in the iteration, that is, I will use the best local quadratic approximation to f near $x_1 = 0.23313$ to compute a still better approximation. The new quadratic approximation,

$$q(x) \doteq -0.40981 + 1.5067(x + 0.23313) + 1.11551(x + 0.23313)^2,$$

is shown in Figure 3. Its minimizer is easily computed to be

$$x_2 \doteq -0.90846.$$

The minimizer of f , to five digits, is $x \doteq -0.83543$.

The process of minimizing successive quadratic approximations in order to locate a minimizer of a nonquadratic function is also referred to as Newton's method. In fact, the alert reader will notice that it is equivalent to Newton's method for solving $f'(x) = 0$. However, there is an advantage to thinking of a quadratic approximation to f rather than a linear approximation to f' : the quadratic approximation has either a minimum or a maximum (depending on whether the coefficient of x^2 is

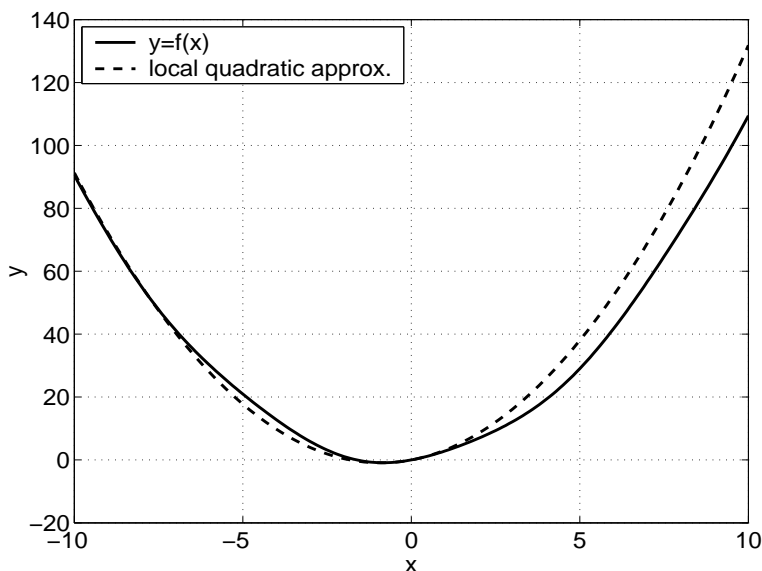


Figure 3: The function $f(x) = \sin(x) + x^2 + x$ and another quadratic approximation.

positive or negative), and so it shows whether the iteration is going in the right direction or not. After all, it is not enough to converge to just any solution of $f'(x) = 0$. The quadratic approximation shows whether the iteration is tending to a stationary point that corresponds to a minimum. If it is not, then the next step should be modified in some fashion. This will be even more important for minimizing functions of several variables.

The best local linear approximation ℓ to f near $x = a$ has the following properties:

$$\ell(a) = f(a), \quad \ell'(a) = f'(a).$$

In fact, I can derive ℓ by imposing these conditions. I assume that ℓ has the form

$$\ell(x) = c_0 + c_1(x - a)$$

and use the conditions $\ell(a) = f(a)$ and $\ell'(a) = f'(a)$ to determine c_0 and c_1 . The equations become simply

$$\begin{aligned} c_0 &= f(a), \\ c_1 &= f'(a). \end{aligned}$$

The reason that the results turned out to be so simple is that I chose to write the linear polynomial ℓ in the form $c_0 + c_1(x - a)$. Had I chosen the equally general form $c_0 + c_1x$, I would have found the same function ℓ but with a more complicated expression for c_0 .

The best local quadratic approximation can be derived in a similar manner. I assume that q has the form

$$q(x) = c_0 + c_1(x - a) + c_2(x - a)^2$$

and use the conditions $q(a) = f(a)$, $q'(a) = f'(a)$, and $q''(a) = f''(a)$ to find c_0 , c_1 , and c_2 . The resulting equations are

$$\begin{aligned} c_0 &= f(a), \\ c_1 &= f'(a), \\ 2c_2 &= f''(a). \end{aligned}$$

Therefore,

$$q(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2}(x - a)^2.$$

Before turning to the case in which f depends on several variables, I wish to discuss some terminology. It is common to refer to a function of the form

$$\ell(x) = c_0 + c_1x$$

as a linear function, although, strictly speaking, it is the second term in the expression that defines a linear function. That is, a *linear* function of one variable has the form

$$\ell(x) = cx,$$

where c is a constant. Such a function satisfies

$$\ell(\alpha_1x_1 + \alpha_2x_2) = \alpha_1\ell(x_1) + \alpha_2\ell(x_2) \text{ for all } \alpha_1, \alpha_2 \in \mathbb{R}, x_1, x_2 \in \mathbb{R}. \quad (2)$$

Condition (2) is the defining property of a linear function. It is rather trivial for functions of one variable, but it will be much more interesting for functions of several variables. A function of the form

$$\ell(x) = c(x - a)$$

can be described as “linear in $x - a$.”

Similarly, it is common to call

$$c_0 + c_1x + c_2x^2$$

a quadratic function, though it might be more precise to restrict the word quadratic for functions of the form

$$cx^2.$$

Thus, the approximation

$$f(x) \doteq q(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2}(x - a)^2 \quad (x \text{ near } a)$$

contains a constant term ($f(a)$), a term linear in $x - a$ ($f'(a)(x - a)$), and a term quadratic in $x - a$ ($(1/2)f''(a)(x - a)^2$).

3 Functions of several variables

A function of the type considered in the last section is denoted $f : \mathbb{R} \rightarrow \mathbb{R}$, meaning that the name of the function is f , its domain (the set of input values) is \mathbb{R} , the set of real numbers, and the output values also lie in \mathbb{R} . In other words, f takes a real number x and maps it to a real number $f(x)$. In some cases, the domain of f is not all of \mathbb{R} , but rather some interval of real numbers. For example, the function $f(x) = xe^x + \log(x) - 1$ is defined only for positive numbers, so I can write $f : (0, \infty) \rightarrow \mathbb{R}$.

I now discuss functions that take several input variables. I will always use vector notation; instead of writing $f(x_1, x_2, \dots, x_n)$, I will write $f(x)$, where $x \in \mathbb{R}^n$ (\mathbb{R}^n is the set of vectors with n components, each component a real number). The reader should notice that I use the same notation for $x \in \mathbb{R}$ and $x \in \mathbb{R}^n$; context will show which is intended. For reasons that will soon be obvious, it is necessary to consider both real-valued functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and *vector-valued* functions $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$. In studying optimization problems, one normally deals with the case in which $m = n$, but in certain special problems (for example, nonlinear least-squares problems), $m \neq n$ is encountered.

3.1 Systems of nonlinear equations

As I mentioned in the introduction, systems of nonlinear equations can be impossible to solve exactly by algebraic methods. The system

$$\begin{aligned}x_2 e^{-x_1^2} &= \frac{1}{10}, \\x_2 + \sin(x_1) &= 0,\end{aligned}$$

which can be written $F(x) = 0$, where

$$F(x) = \begin{bmatrix} x_2 e^{-x_1^2} - \frac{1}{10} \\ x_2 + \sin(x_1) \end{bmatrix}.$$

I can graph the curves $x_2 e^{-x_1^2} - 1/10 = 0$ and $x_2 + \sin(x_1) = 0$ in the plane; any points of intersection correspond to solution of the system.¹ The graphs are shown in Figure 4, which shows that the system has two solutions. This example illustrates one of the reasons that systems of nonlinear equations are difficult to solve: it is usually impossible to predict in advance how many solutions a given system has. By contrast, a system of n linear equations in n unknowns has exactly one solution except in the case that the coefficient matrix defining the system is singular, in which case such a system must have no solutions or infinitely many solutions.

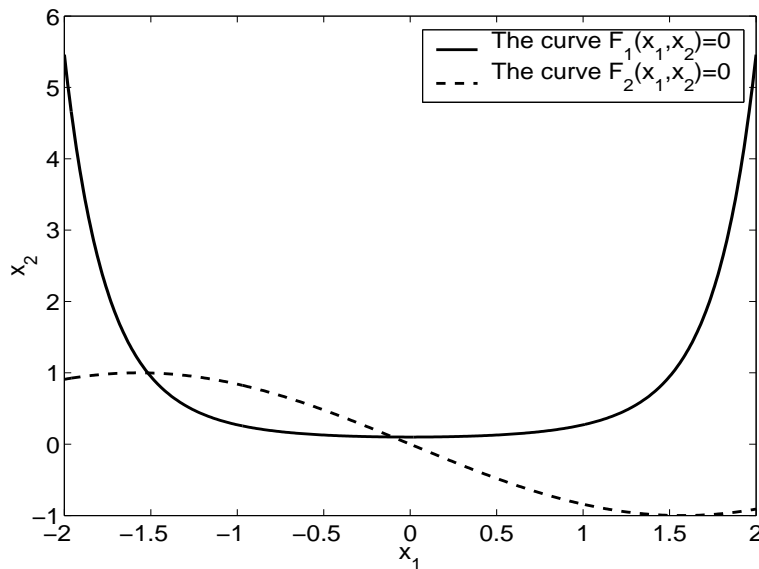


Figure 4: The two curves defined by a 2×2 system of nonlinear equations.

Now suppose I wish to apply Newton's method to solve $F(x) = 0$. I can see from Figure 4 that there is a solution near the point $(-1/10, 1/10)$. Indeed, with $x^{(0)} = (-1/10, 1/10)$,

$$F(x^{(0)}) \doteq \begin{bmatrix} 0.26788 \\ 0.15853 \end{bmatrix},$$

which is not far from the zero vector. The question I must answer is: how do I construct a local linear approximation to F near $x^{(0)}$?

Part of the answer is provided by a standard result from linear algebra: Each linear transformation from \mathbb{R}^2 to \mathbb{R}^2 can be represented by matrix-vector multiplication with a certain 2×2 matrix.

¹The reader should notice that such graphical analysis is strictly impossible if there are more than three unknowns, and is usually not useful for the case of three variables.

This implies that the local linear approximation to F near $x^{(0)}$ will have the form

$$F(x) \doteq L(x) = F\left(x^{(0)}\right) + J\left(x - x^{(0)}\right) \quad \left(x \text{ near } x^{(0)}\right),$$

where J is a 2×2 matrix. If I impose the condition that F and L must have the same first partial derivatives at $x^{(0)}$, that is,

$$\frac{\partial F_i}{\partial x_j}\left(x^{(0)}\right), \quad i, j = 1, 2,$$

then I obtain the so-called *Jacobian matrix*:

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial x_1}\left(x^{(0)}\right) & \frac{\partial F_1}{\partial x_2}\left(x^{(0)}\right) \\ \frac{\partial F_2}{\partial x_1}\left(x^{(0)}\right) & \frac{\partial F_2}{\partial x_2}\left(x^{(0)}\right) \end{bmatrix}.$$

To be precise, the derivative of F at $x^{(0)}$ is a linear transformation $DF\left(x^{(0)}\right)$ that satisfies

$$F(x) \doteq F\left(x^{(0)}\right) + DF\left(x^{(0)}\right)\left(x - x^{(0)}\right) \quad \left(x \text{ near } x^{(0)}\right)$$

in a certain sense to be described later. As in the case of one variable, this formula involves a constant term and a term linear in $x - x^{(0)}$. The linear transformation $DF\left(x^{(0)}\right)$ is represented by the matrix J in the sense that

$$DF\left(x^{(0)}\right)y = Jy \quad \text{for all } y \in \mathbb{R}^2.$$

The matrix J depends directly on $x^{(0)}$, so it would be more correct to write $J\left(x^{(0)}\right)$; however, I will usually suppress this dependence and just write J . For a more complete explanation and derivation of the Jacobian matrix (as well as the other results in this paper), the reader can consult my paper [1].

Now I can show how to improve on the estimate $x^{(0)}$ of the solution of $F(x) = 0$. Since

$$F(x) \doteq F\left(x^{(0)}\right) + J\left(x - x^{(0)}\right) \quad \left(x \text{ near } x^{(0)}\right),$$

I solve

$$F\left(x^{(0)}\right) + J\left(x - x^{(0)}\right) = 0;$$

the solution will be $x^{(1)}$, the improved estimate of the true solution. It is easy to show that

$$x^{(1)} = x^{(0)} - J^{-1}F\left(x^{(0)}\right).$$

In this example, the reader can verify that

$$J\left(x^{(0)}\right) \doteq \begin{bmatrix} 0.019801 & 0.99005 \\ 0.99500 & 1.00000 \end{bmatrix}$$

and

$$x^{(1)} \doteq \begin{bmatrix} -1.0120 \\ 1.0103 \end{bmatrix}.$$

Moreover,

$$F\left(x^{(1)}\right) \doteq \begin{bmatrix} -3.8635 \cdot 10^{-7} \\ 7.2364 \cdot 10^{-8} \end{bmatrix},$$

which shows that Newton's method really did succeed in producing an improved estimate of the true solution.

In general, the Newton iteration takes the form

$$x^{(k+1)} = x^{(k)} - J^{-1}F\left(x^{(k)}\right) \quad \left(J = J\left(x^{(k)}\right)\right).$$

This formula is the same whether $F: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, as in the above example, or $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ for $n > 2$. (This is the power of matrix-vector notation.) The only difference in moving from $n = 2$ to $n > 2$ is that the vectors $x^{(k)}$, $x^{(k+1)}$, and $F\left(x^{(k)}\right)$ become n -vectors, and the matrix J becomes an $n \times n$ matrix.

3.2 Minimization of a function of several variables

I now consider the problem of finding a minimizer of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The reader who has understood my development to this point will recognize that the first order of business is to find the best local linear and quadratic approximations to f near a given point $a \in \mathbb{R}^n$.

Since \mathbb{R} is isomorphic to \mathbb{R}^1 , the results of the last section show how to construct a local linear approximation:

$$f(x) \doteq f(a) + J(x - a) \quad (x \text{ near } a),$$

where J is the $1 \times n$ matrix

$$J = J(a) = \left[\frac{\partial f}{\partial x_1}(a) \quad \frac{\partial f}{\partial x_2}(a) \quad \cdots \quad \frac{\partial f}{\partial x_n}(a) \right].$$

It turns out convenient to write this approximation in an equivalent fashion using the $(n \times 1)$ gradient vector

$$\nabla f(a) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(a) \\ \frac{\partial f}{\partial x_2}(a) \\ \vdots \\ \frac{\partial f}{\partial x_n}(a) \end{bmatrix}$$

in place of the $1 \times n$ Jacobian matrix. The best local linear approximation then takes the form

$$f(x) \doteq f(a) + \nabla f(a) \cdot (x - a) \quad (x \text{ near } a).$$

The reader should notice the dot product in this formula.

Constructing the best local quadratic approximation to f near a is a bit more involved, because I have to explain the form of a quadratic function q in n variables. Such a function must contain all terms of the form

$$x_i x_j, \quad i, j = 1, 2, \dots, n,$$

and hence can be written as

$$q(x) = \sum_{i=1}^n \sum_{j=1}^n h_{ij} x_i x_j.$$

However, this representation is not unique; since $x_i x_j = x_j x_i$, h_{ij} and h_{ji} can be changed arbitrarily as long as $h_{ij} + h_{ji}$ is not changed. For example, the quadratic in two variables

$$x_1^2 + 4x_1 x_2 + 3x_2^2 \quad (h_{11} = 1, h_{12} = 4, h_{21} = 0, h_{22} = 3) \tag{3}$$

can also be written as

$$x_1^2 + 3x_1 x_2 + x_2 x_1 + 3x_2^2 \quad (h_{11} = 1, h_{12} = 3, h_{21} = 1, h_{22} = 3),$$

and in infinitely many other ways. For reasons that will become clear below, I will assume that $h_{ij} = h_{ji}$. For example, (3) would be written as

$$x_1^2 + 2x_1 x_2 + 2x_2 x_1 + 3x_2^2 \quad (h_{11} = 1, h_{12} = 2, h_{21} = 2, h_{22} = 3).$$

A familiarity with matrix-vector manipulations suggests the following calculation:

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^n h_{ij} x_i x_j &= \sum_{i=1}^n x_i \left(\sum_{j=1}^n h_{ij} x_j \right) \\ &= \sum_{i=1}^n x_i (Hx)_i \\ &= x \cdot Hx, \end{aligned}$$

where H is the $n \times n$ matrix whose i, j -entry is h_{ij} . According to my assumption about the coefficients h_{ij} , H is a *symmetric* matrix, which means that $H^T = H$. Symmetric matrices have special properties with regard to their eigenvalues and eigenvectors; these properties will be important later.

Thus a quadratic function in n variables is determined by an $n \times n$ symmetric matrix H . Moreover, if

$$q(x) = x \cdot Hx = \sum_{i=1}^n \sum_{j=1}^n h_{ij} x_i x_j,$$

then

$$\frac{\partial^2 q}{\partial x_j \partial x_i} = h_{ij} + h_{ji} = 2h_{ij}.$$

To avoid the annoying factor of 2, it is usual to write a general quadratic in the form

$$q(x) = \frac{1}{2} x \cdot Hx.$$

(The reader might think that the factor of $1/2$ would be more annoying than the factor of 2. However, in numerical optimization we are constantly manipulating the derivatives and much less often the function itself.)

I have now shown that a local quadratic approximation to f near a should have the form

$$f(x) \doteq f(a) + \nabla f(a) \cdot (x - a) + \frac{1}{2} (x - a) \cdot H(x - a).$$

The principle that the best local quadratic approximation should have the same second partial derivatives as does f itself yields

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(a) & \frac{\partial^2 f}{\partial x_2 \partial x_1}(a) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(a) \\ \frac{\partial^2 f}{\partial x_1 \partial x_2}(a) & \frac{\partial^2 f}{\partial x_2^2}(a) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_2}(a) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(a) & \frac{\partial^2 f}{\partial x_2 \partial x_n}(a) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(a) \end{bmatrix}.$$

(For a more complete justification, the reader can consult [1].) This matrix containing the second partial derivatives of f is called the *Hessian* matrix of f (at a) and is denoted $\nabla^2 f(a)$. As long as f is smooth,

$$\frac{\partial f}{\partial x_i \partial x_j}(a) = \frac{\partial f}{\partial x_j \partial x_i}(a),$$

and so $\nabla^2 f(a)$ is guaranteed to be a symmetric matrix.

I have now derived the best local quadratic approximation to f near $x = a$:

$$f(x) \doteq f(a) + \nabla f(a) \cdot (x - a) + \frac{1}{2} (x - a) \cdot \nabla^2 f(a)(x - a) \quad (x \text{ near } a).$$

As before, this formula involves a constant term, a term linear in $x - a$, and a term quadratic in $x - a$. Newton's method then takes the form

$$x^{(k+1)} = x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}).$$

(Since the Hessian is the Jacobian of the gradient, this really is just Newton's method applied to the system $\nabla f(x) = 0$.)

Here is a specific example; the reader can verify the individual steps in the computation. I define $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ by

$$f(x) = x_1 e^{-x_2^2} + x_1^2 + x_1 + x_2^2 - x_2.$$

Then

$$\begin{aligned}\nabla f(x) &= \begin{bmatrix} e^{-x_2^2} + 2x_1 + 1 \\ -2x_1x_2e^{-x_2^2} + 2x_2 - 1 \end{bmatrix}, \\ \nabla^2 f(x) &= \begin{bmatrix} 2 & -2x_2e^{-x_2^2} \\ -2x_2e^{-x_2^2} & 4x_1x_2^2e^{-x_2^2} - 2x_1e^{-x_2^2} + 2 \end{bmatrix}.\end{aligned}$$

I will take $x^{(0)} = (0, 0)$ as my initial estimate of the minimizer. The initial function value is $f(x^{(0)}) = 0$. Then

$$\begin{aligned}\nabla f(x^{(0)}) &= \begin{bmatrix} 2 \\ -1 \end{bmatrix}, \\ \nabla^2 f(x^{(0)}) &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},\end{aligned}$$

and $x^{(1)} = (-1, 1/2)$. Since $f(x^{(1)}) \doteq -1.0288$, Newton's method succeeded in reducing the function value and presumably moved toward a minimizer. The next iteration proceeds as follows:

$$\begin{aligned}\nabla f(x^{(1)}) &\doteq \begin{bmatrix} -0.22120 \\ 0.77880 \end{bmatrix}, \\ \nabla^2 f(x^{(1)}) &\doteq \begin{bmatrix} 2.0000 & -0.77880 \\ -0.77880 & 2.7788 \end{bmatrix}, \\ x^{(2)} &\doteq \begin{bmatrix} -0.99836 \\ 0.22020 \end{bmatrix}.\end{aligned}$$

The new function value is $f(x^{(2)}) \doteq -1.1245$, which again shows a reduction. Also,

$$\nabla f(x^{(2)}) \doteq \begin{bmatrix} -0.044042 \\ -0.14075 \end{bmatrix},$$

and the fact that the gradient is getting smaller also suggests that Newton's method is converging to a minimizer. In fact, the minimizer is

$$x^* \doteq \begin{bmatrix} -0.96662 \\ 0.26287 \end{bmatrix},$$

and $f(x^*) \doteq -1.1281$.

Newton's method is ideal for *local convergence*, that is, improving a reasonably good estimate of the desired solution to a high degree of accuracy. However, if a good initial estimate of the solution is not known, then Newton's method must be combined with other techniques to create a robust algorithm.

4 More about local approximations

For any function f (scalar- or vector-valued), the best local linear approximation near $x = a$ takes the form

$$f(x) \doteq f(a) + Df(a)(x - a) \quad (x \text{ near } a), \quad (4)$$

where $Df(a)$ is a linear transformation. If $f : \mathbb{R} \rightarrow \mathbb{R}$, then the derivative $Df(a)$ is represented the ordinary derivative $f'(a)$, which is a number.² If $f : \mathbb{R}^n \rightarrow \mathbb{R}$, then $Df(a)$ is represented by the gradient vector $\nabla f(a)$. Finally, if $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, then $Df(a)$ is represented by the $m \times n$ Jacobian matrix $J(a)$.

To analyze the convergence of Newton's method, it is necessary to know the sense in which the approximation (4) is valid. The definition is as follows:

²The reader will recall from page 5 that a linear function from \mathbb{R} to \mathbb{R} is represented by a single number.

Definition 4.1 Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $a \in \mathbb{R}^n$. Then f is differentiable at a if there exists a linear transformation $Df(a) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ satisfying

$$\frac{\|f(x) - f(a) - Df(a)\delta x\|}{\|\delta x\|} \rightarrow 0 \text{ as } \|\delta x\| \rightarrow 0. \quad (5)$$

(If either x or f is scalar-valued, then corresponding norm can be replaced by absolute value signs.) In order that the ratio

$$\frac{\|f(x) - f(a) - Df(a)\delta x\|}{\|\delta x\|}$$

tend to zero as $\|\delta x\|$ goes to zero, it must be the case that

$$\|f(x) - f(a) - Df(a)\delta x\|$$

goes to zero faster than $\|\delta x\|$ does. It is customary to write

$$f(x) - f(a) - Df(a)\delta x = o(\|\delta x\|) \text{ as } \|\delta x\| \rightarrow 0$$

to indicate that $f(x) - f(a) - Df(a)\delta x$ is small compared to δx in this sense. (Often the phrase “as $\|\delta x\| \rightarrow 0$ ” is omitted.) The symbol $o(\|\delta x\|)$ is read “little-oh of $\|\delta x\|$.”

The upshot of all this is that the error in the approximation

$$f(x) \doteq f(a) + Df(a)(x - a)$$

is a quantity that is small compared to $\|x - a\|$ as $x \rightarrow a$.

The situation with the quadratic approximation is more difficult to describe in general because I do not want to take the time describe second derivatives of vector-valued functions $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Therefore, I will just give the results for scalar-valued functions. If $f : \mathbb{R} \rightarrow \mathbb{R}$ is twice differentiable, then

$$f(x) - f(a) - f'(a)(x - a) - \frac{f''(a)}{2}(x - a)^2 = o(|x - a|^2),$$

the precise meaning of which is

$$\frac{\left| f(x) - f(a) - f'(a)(x - a) - \frac{f''(a)}{2}(x - a)^2 \right|}{|x - a|^2} \rightarrow 0 \text{ as } |x - a| \rightarrow 0.$$

If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice differentiable, then

$$f(x) - f(a) - \nabla f(a) \cdot (x - a) - \frac{1}{2}(x - a) \cdot \nabla^2 f(a)(x - a) = o(\|x - a\|^2).$$

Similarly, this means that

$$\frac{\left\| f(x) - f(a) - \nabla f(a) \cdot (x - a) - \frac{1}{2}(x - a) \cdot \nabla^2 f(a)(x - a) \right\|}{\|x - a\|^2} \rightarrow 0 \text{ as } \|x - a\| \rightarrow 0.$$

Finally, if $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice-differentiable, then

$$f(x) - f(a) - \nabla f(a) \cdot (x - a) = \frac{1}{2}(x - a) \cdot \nabla^2 f(a)(x - a) + o(\|x - a\|^2).$$

The quantity

$$\left\| \frac{1}{2}(x - a) \cdot \nabla^2 f(a)(x - a) \right\|^2$$

is bounded by a constant times $\|x - a\|^2$. This fact is expressed as

$$\frac{1}{2}(x - a) \cdot \nabla^2 f(a)(x - a) = O(\|x - a\|^2).$$

The symbol $O(\|x - a\|^2)$ is read “big-oh of $\|x - a\|^2$ ” and is interpreted informally as a quantity that is approximately proportional to $\|x - a\|^2$ as $\|x - a\| \rightarrow 0$. It follows that, in the case that f is twice-differentiable,

$$f(x) - f(a) - \nabla f(a) \cdot (x - a) = O(\|x - a\|^2).$$

In other words, when f is not just once but twice differentiable, the error in the linear approximation is not just $o(\|x - a\|)$, but in fact $O(\|x - a\|^2)$. This is a more precise estimate of the error. This result is valid in general: if $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is twice differentiable, then

$$f(x) - f(a) - Df(a)(x - a) = O(\|x - a\|^2).$$

References

- [1] Mark S. Gockenbach. A primer on differentiation. *Optimization and Engineering*, 2:75–129, 2001.