Jim Lambers
MAT 419/519
Summer Session 2011-12
Lecture 9 Notes

These notes correspond to Section 3.1 in the text.

Newton's Method

Finding the minimum of the function $f(\mathbf{x})$, where $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$, requires finding its critical points, at which $\nabla f(\mathbf{x}) = \mathbf{0}$. In general, however, solving this system of equations can be quite difficult. Therefore, it is often necessary to use *numerical methods* that compute an *approximate* solution. We now present one such method, known as *Newton's Method* or the *Newton-Rhapson Method*.

Let $\mathbf{g} : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be a function that is differentiable on D. As it is a vector-valued function, it has component functions $g_i(\mathbf{x})$, $i = 1, 2, \ldots, n$, and thus we have

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \cdots \\ g_n(\mathbf{x}) \end{bmatrix}, \quad \mathbf{x} \in D.$$

Newton's Method is an *iterative* method that computes an approximate solution to the system of equations $\mathbf{g}(\mathbf{x}) = \mathbf{0}$. The method requires an initial guess $\mathbf{x}^{(0)}$ as input. It then computes subsequent iterates $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, ... that, hopefully, will converge to a solution $\mathbf{x}^*$ of $\mathbf{g}(\mathbf{x}) = \mathbf{0}$.

The idea behind Newton's Method is to approximate $\mathbf{g}(\mathbf{x})$ near the current iterate $\mathbf{x}^{(k)}$ by a function $\mathbf{g}_k(\mathbf{x})$ for which the system of equations $\mathbf{g}_k(\mathbf{x})$ is easy to solve, and then use the solution as the next iterate $\mathbf{x}^{(k+1)}$, after which this process is repeated. A suitable choice for $\mathbf{g}_k(\mathbf{x})$ is the *linear approximation* of $\mathbf{g}(\mathbf{x})$ at $\mathbf{x}^{(k)}$, whose graph is the *tangent space* to $\mathbf{g}(\mathbf{x})$ at $\mathbf{x}^{(k)}$:

$$\mathbf{g}_k(\mathbf{x}) = \mathbf{g}(\mathbf{x}^{(k)}) + J_{\mathbf{g}}(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)}),$$

where $J_{\mathbf{g}}(\mathbf{x})$ is the *Jacobian matrix* of $\mathbf{g}(\mathbf{x})$, defined by

$$[J_{\mathbf{g}}(\mathbf{x})]_{ij} = \frac{\partial g_i(\mathbf{x})}{\partial x_j}.$$

That is, $J_{\mathbf{g}}(\mathbf{x})$ is the matrix of first partial derivatives of the component functions of $\mathbf{g}(\mathbf{x})$.

Example Let

$$\mathbf{g}(x, y, z) = \begin{bmatrix} g_1(x, y, z) \\ g_2(x, y, z) \\ \cdots \\ g_n(x, y, z) \end{bmatrix} = \begin{bmatrix} x^2 + y^2 + z^2 - 3 \\ x^2 + y^2 - z - 1 \\ x + y + z - 3 \end{bmatrix}.$$

Then

$$J_{\mathbf{g}}(x,y,z) = \begin{bmatrix} \frac{\partial g_1}{\partial x} & \frac{\partial g_1}{\partial y} & \frac{\partial g_1}{\partial z} \\ \frac{\partial g_2}{\partial x} & \frac{\partial g_2}{\partial y} & \frac{\partial g_2}{\partial z} \\ \frac{\partial g_3}{\partial x} & \frac{\partial g_3}{\partial y} & \frac{\partial g_3}{\partial z} \end{bmatrix} = \begin{bmatrix} 2x & 2y & 2z \\ 2x & 2y & -1 \\ 1 & 1 & 1 \end{bmatrix}.$$

□

Now, we solve the equation $\mathbf{g}_k(\mathbf{x}^{(k+1)}) = \mathbf{0}$ for the next iterate $\mathbf{x}^{(k+1)}$. Setting $\mathbf{g}_k(\mathbf{x}) = \mathbf{0}$ in the definition of $\mathbf{g}_k(\mathbf{x})$ yields the system of equations

$$\mathbf{0} = \mathbf{g}(\mathbf{x}^{(k)}) + J_{\mathbf{g}}(\mathbf{x}^{(k)})(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}).$$

Rearranging yields

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [J_{\mathbf{g}}(\mathbf{x}^{(k)})]^{-1} g(\mathbf{x}^{(k)}).$$

This is the process by which each new iterate $\mathbf{x}^{(k+1)}$ is obtained from the previous iterate $\mathbf{x}^{(k)}$, for $k = 0, 1, \ldots$.

In practice, one does not compute $\mathbf{x}^{(k+1)}$ by explicitly computing $[J_{\mathbf{g}}(\mathbf{x}^{(k)})]^{-1}$ and then multiplying by $\mathbf{g}(\mathbf{x}^{(k)})$, because this is computationally inefficient. Instead, it is more practical to solve the system of linear equations

$$J_{\mathbf{g}}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} = -\mathbf{g}(\mathbf{x}^{(k)})$$

for the unknown $\mathbf{s}^{(k)}$, using a method such as Gaussian elimination, and then setting $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$.

For a single-variable function $g(x)$, Newton's Method reduces to

$$x^{(k+1)} = x^{(k)} - \frac{g(x^{(k)})}{g'(x^{(k)})}, \quad k = 0, 1, \ldots.$$

Note that it is necessary that $g'(x^{(k)}) \neq 0$; otherwise, the sequence of Newton iterates is undefined. Similarly, in the multi-variable case, when $J_{\mathbf{g}}(\mathbf{x}^{(k)})$ is not an invertible matrix, the solution $\mathbf{s}^{(k)}$ may not exist, in which case the sequence of Newton iterates is also undefined.

We now illustrate the use of Newton's Method in the single-variable case with some examples.

Example We will use of Newton's Method in computing $\sqrt{2}$. This number satisfies the equation $f(x) = 0$ where

$$f(x) = x^2 - 2.$$

Since $f'(x) = 2x$, it follows that in Newton's Method, we can obtain the next iterate $x^{(n+1)}$ from the previous iterate $x^{(n)}$ by

$$x^{(n+1)} = x^{(n)} - \frac{f(x^{(n)})}{f'(x^{(n)})} = x^{(n)} - \frac{[x^{(n)}]^2 - 2}{2x^{(n)}} = x^{(n)} - \frac{[x^{(n)}]^2}{2x^{(n)}} + \frac{2}{2x^{(n)}} = \frac{x^{(n)}}{2} + \frac{1}{x^{(n)}}.$$

We choose our starting iterate $x_0 = 1$, and compute the next several iterates as follows:

$$\begin{aligned}
x_1 &= \frac{1}{2} + \frac{1}{1} = 1.5 \\
x_2 &= \frac{1.5}{2} + \frac{1}{1.5} = 1.41666667 \\
x_3 &= 1.41421569 \\
x_4 &= 1.41421356 \\
x_5 &= 1.41421356.
\end{aligned}$$

Since the fourth and fifth iterates agree in to eight decimal places, we assume that 1.41421356 is a correct solution to $f(x) = 0$, to at least eight decimal places. The first two iterations are illustrated in Figure 1. $\Box$

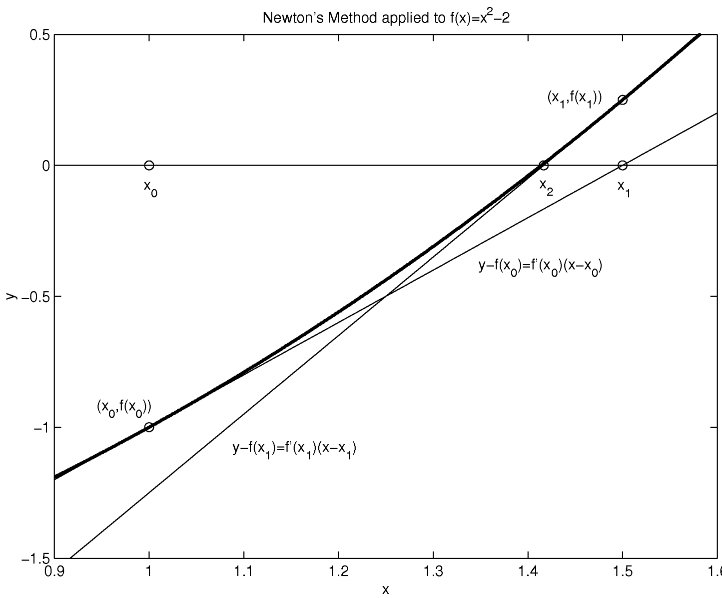

Figure 1: Newton's Method applied to $f(x) = x^2 - 2$. The bold curve is the graph of f. The initial iterate x_0 is chosen to be 1. The tangent line of $f(x)$ at the point $(x_0, f(x_0))$ is used to approximate $f(x)$, and it crosses the x-axis at $x_1 = 1.5$, which is much closer to the exact solution than x_0. Then, the tangent line at $(x_1, f(x_1))$ is used to approximate $f(x)$, and it crosses the x-axis at $x_2 = 1.41\bar{6}$, which is already very close to the exact solution.

Example Newton's Method can be used to compute the reciprocal of a number a *without perform-*

ing any divisions. The solution, $1/a$, satisfies the equation $f(x) = 0$, where

$$f(x) = a - \frac{1}{x}.$$

Since

$$f'(x) = \frac{1}{x^2},$$

it follows that in Newton's Method, we can obtain the next iterate $x^{(n+1)}$ from the previous iterate $x^{(n)}$ by

$$x^{(n+1)} = x^{(n)} - \frac{a - 1/x^{(n)}}{1/[x^{(n)}]^2} = x^{(n)} - \frac{a}{1/x^{(n)}}^{2} + \frac{1/x^{(n)}}{1/[x^{(n)}]^2} = 2x^{(n)} - a[x^{(n)}]^2.$$

Note that no divisions are necessary to obtain $x^{(n+1)}$ from $x^{(n)}$. This iteration was actually used on older IBM computers to implement division in hardware.

We use this iteration to compute the reciprocal of $a = 12$. Choosing our starting iterate to be 0.1, we compute the next several iterates as follows:

$$\begin{aligned}
x_1 &= 2(0.1) - 12(0.1)^2 = 0.08 \\
x_2 &= 2(0.08) - 12(0.08)^2 = 0.0832 \\
x_3 &= 0.0833312 \\
x_4 &= 0.08333333333279 \\
x_5 &= 0.08333333333333.
\end{aligned}$$

We conclude that 0.08333333333333 is an accurate approximation to the correct solution.

Now, suppose we repeat this process, but with an initial iterate of $x_0 = 1$. Then, we have

$$\begin{aligned}
x_1 &= 2(1) - 12(1)^2 = -10 \\
x_2 &= 2(-6) - 12(-6)^2 = -1220 \\
x_3 &= 2(-300) - 12(-300)^2 = -17863240
\end{aligned}$$

It is clear that this sequence of iterates is not going to converge to the correct solution. In general, for this iteration to converge to the reciprocal of a, the initial iterate x_0 must be chosen so that $0 < x_0 < 2/a$. This condition guarantees that the next iterate x_1 will at least be positive. The contrast between the two choices of x_0 are illustrated in Figure 2. □

These examples demonstrate that on the one hand, Newton's Method can converge to a solution very rapidly. On the other hand, it may not converge at all, if the initial guess $\mathbf{x}^{(0)}$ is not chosen sufficiently close to the solution $\mathbf{x}^*$.

Example We consider the system of equations $\mathbf{g}(\mathbf{x}) = \mathbf{0}$, where

$$\mathbf{g}(x, y, z) = \begin{bmatrix} x^2 + y^2 + z^2 - 3 \\ x^2 + y^2 - z - 1 \\ x + y + z - 3 \end{bmatrix}.$$

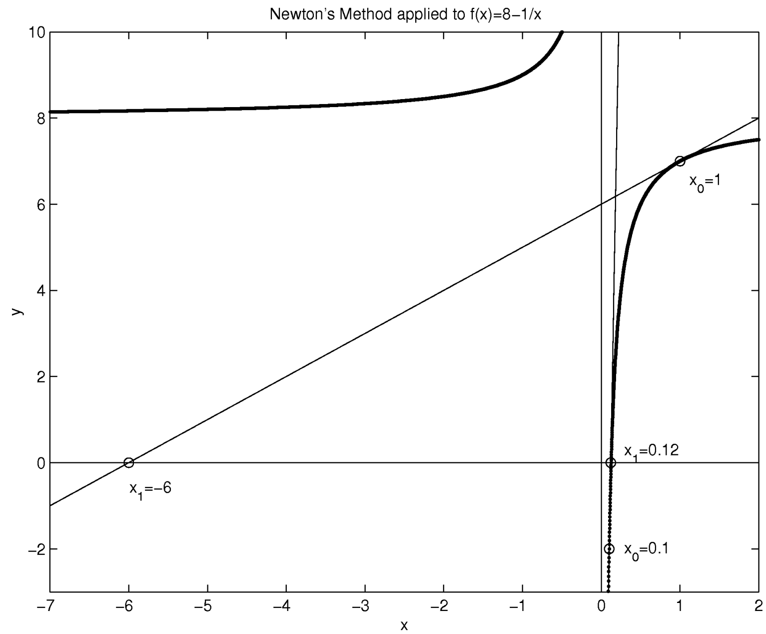

Figure 2: Newton's Method used to compute the reciprocal of 8 by solving the equation $f(x) = 8 - 1/x = 0$. When $x_0 = 0.1$, the tangent line of $f(x)$ at $(x_0, f(x_0))$ crosses the x-axis at $x_1 = 0.12$, which is close to the exact solution. When $x_0 = 1$, the tangent line crosses the x-axis at $x_1 = -6$, which causes searcing to continue on the wrong portion of the graph, so the sequence of iterates does not converge to the correct solution.

We will begin to use Newton's Method to solve this system of equations, with initial guess $\mathbf{x}^{(0)} = (x^{(0)}, y^{(0)}, z^{(0)}) = (1, 0, 1)$.

As computed in a previous example,

$$J_{\mathbf{g}}(x, y, z) = \begin{bmatrix} 2x & 2y & 2z \\ 2x & 2y & -1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Therefore, each Newton iterate $\mathbf{x}^{(k+1)}$ is obtained by solving the system of equations

$$J_{\mathbf{g}}(x^{(k)}, y^{(k)}, z^{(k)})(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = -\mathbf{g}(x^{(k)}, y^{(k)}, z^{(k)}),$$

or

$$\begin{bmatrix} 2x^{(k)} & 2y^{(k)} & 2z^{(k)} \\ 2x^{(k)} & 2y^{(k)} & -1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x^{(k+1)} - x^{(k)} \\ y^{(k+1)} - y^{(k)} \\ z^{(k+1)} - z^{(k)} \end{bmatrix} = - \begin{bmatrix} (x^{(k)})^2 + (y^{(k)})^2 + (z^{(k)})^2 - 3 \\ (x^{(k)})^2 + (y^{(k)})^2 - z^{(k)} - 1 \\ x^{(k)} + y^{(k)} + z^{(k)} - 3 \end{bmatrix}.$$

Setting $k = 0$ and substituting $(x^{(0)}, y^{(0)}, z^{(0)}) = (1, 0, 1)$ yields the system

$$\begin{bmatrix} 2 & 0 & 2 \\ 2 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x^{(1)} - 1 \\ y^{(1)} \\ z^{(1)} - 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},$$

which has the solution $\mathbf{x}^{(1)} = (\frac{3}{2}, \frac{1}{2}, 1)$. Repeating this process with $k = 1$ yields the system

$$\begin{bmatrix} 3 & 1 & 2 \\ 3 & 1 & -1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x^{(2)} - \frac{3}{2} \\ y^{(2)} - \frac{1}{2} \\ z^{(2)} - 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \\ -\frac{1}{2} \\ 0 \end{bmatrix},$$

which has the solution $\mathbf{x}^{(2)} = (\frac{5}{4}, \frac{3}{4}, 1)$. A similar process yields the next iterate $\mathbf{x}^{(3)} = \left(\frac{9}{8}, \frac{7}{8}, 1\right)$.

It can be seen that these iterates are converging to $(1, 1, 1)$, which is the exact solution. However, if we instead use the initial guess $\mathbf{x}^{(0)} = (0, 0, 0)$, then we obtain

$$J_{\mathbf{g}}(0, 0, 0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 1 & 1 & 1 \end{bmatrix},$$

which is not invertible. The system $J_{\mathbf{g}}(0, 0, 0)\mathbf{s}^{(0)} = -\mathbf{g}(0, 0, 0)$ does not have a solution, and therefore Newton's Method fails. □

Now, suppose that we wish to minimize a function $f(\mathbf{x})$, and therefore need to solve the system of equations $\nabla f(\mathbf{x}) = \mathbf{0}$. Then, $g(\mathbf{x}) = \nabla f(\mathbf{x})$, and $J_{\mathbf{g}}(\mathbf{x}) = Hf(\mathbf{x})$. Therefore, the Newton's Method step takes the form

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [Hf(\mathbf{x}^{(k)})]^{-1} \nabla f(\mathbf{x}^{(k)}).$$

Effectively, when used for minimization, Newton's Method approximates $f(\mathbf{x})$ by its *quadratic approximation* near $\mathbf{x}^{(k)}$,

$$f_k(\mathbf{x}) = f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)}) \cdot (\mathbf{x} - \mathbf{x}^{(k)}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^{(k)}) \cdot Hf(\mathbf{x}^{(k)})(\mathbf{x} - \mathbf{x}^{(k)})$$

and then computing the unique critical point of $f_k(\mathbf{x})$, which is the unique solution of $\nabla f_k(\mathbf{x}) = \mathbf{0}$. Note that $\nabla f_k(\mathbf{x}) = \nabla f(\mathbf{x}^{(k)})$, and $Hf_k(\mathbf{x}) = Hf(\mathbf{x}^{(k)})$.

If $Hf(\mathbf{x}^{(k)}$ is positive definite, then this critical point is also guaranteed to be the unique strict global minimizer of $f_k(\mathbf{x})$. For quadratic functions in general, we have this useful result.

Theorem Let A be an $n \times n$ symmetric positive definite matrix, let $\mathbf{b} \in \mathbb{R}^n$, and let $a \in \mathbb{R}$. Then the quadratic function

$$f(\mathbf{x}) = a + \mathbf{b}\dot{\mathbf{x}} + \frac{1}{2}\mathbf{x} \cdot A\mathbf{x}$$

is strictly convex and has a unique strict global minimizer $\mathbf{x}^*$, where

$$A\mathbf{x}^* = -\mathbf{b}.$$

For any initial guess $\mathbf{x}^{(0)}$, Newton's Method applied to $f(\mathbf{x})$ converges to $\mathbf{x}^*$ in one step; that is, $\mathbf{x}^{(1)} = \mathbf{x}^*$.

If $f(\mathbf{x})$ is not a quadratic function, then Newton's Method will generally not compute a minimizer of $f(\mathbf{x})$ in one step, even if its Hessian $Hf(\mathbf{x})$ is positive definite. However, in this case, Newton's Method is guaranteed to make progress, as the following theorem indicates.

Theorem Let $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ be the sequence of Newton iterates for the function $f(\mathbf{x})$. If $Hf(\mathbf{x}^{(k)}$ is positive definite and if $\nabla f(\mathbf{x}^{(k)}) \neq \mathbf{0}$, then the vector

$$\mathbf{s}^{(k)} = -[Hf(\mathbf{x}^{(k)}]^{-1}\nabla f(\mathbf{x}^{(k)})$$

from $\mathbf{x}^{(k)}$ to $\mathbf{x}^{(k+1)}$ is a descent direction for $f(\mathbf{x})$; that is,

$$f(\mathbf{x}^{(k)} + t\mathbf{s}^{(k)}) < f(\mathbf{x}^{(k)})$$

for t sufficiently small.

Example Let $f(x, y) = x^4 + 2x^2y^2 + y^4$. Then we have

$$\nabla f(x, y) = (4x^3 + 4xy^2, 4x^2y + 4y^3),$$

and

$$Hf(x, y) = \begin{bmatrix} 12x^2 + 4y^2 & 8xy \\ 8xy & 4x^2 + 12y^2 \end{bmatrix}.$$

Let $\mathbf{x}^{(0)} = (a, a)$ for some $a \neq 0$. Then

$$\nabla f(\mathbf{x}^{(0)} = (8a^3, 8a^3) = 8a^3(1, 1),$$

and

$$Hf(\mathbf{x}^{(0)}) = \begin{bmatrix} 16a^2 & 8a^2 \\ 8a^2 & 16a^2 \end{bmatrix} = 8a^2 \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Then, using the formula for the inverse of a 2×2 matrix,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix},$$

we obtain

$$\mathbf{s}^{(0)} = -[Hf(\mathbf{x}^{(0)}]^{-1}\nabla f(\mathbf{x}^{(0)} = -\frac{8a^3}{24a^2} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = -\frac{a}{3} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

It follows that

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \mathbf{s}^{(0)} = (a, a) - \frac{a}{3}(1, 1) = \left(\frac{2a}{3}, \frac{2a}{3}\right).$$

It can be seen that in general,

$$\mathbf{x}^{(k)} = \left(\frac{2}{3}\right)^k (a, a),$$

and therefore the Newton iterates converge to $(0, 0)$, which is the unique strict global minimizer of $f(x, y)$. $\square$

Exercises

1. Chapter 3, Exercise 2
2. Chapter 3, Exercise 3
3. Chapter 3, Exercise 4