

**Jim Lambers**  
**MAT 461/561**  
**Spring Semester 2009-10**  
**Lecture 24 Notes**

These notes correspond to Section 10.3 in the text.

## Quasi-Newton Methods

One of the drawbacks of using Newton's Method to solve a system of nonlinear equations  $\mathbf{F}(\mathbf{x}) = \mathbf{0}$  is the computational expense that must be incurred during each iteration to evaluate the partial derivatives of  $\mathbf{F}$  at  $\mathbf{x}^{(k)}$ , and then solve a system of linear equations involving the resulting Jacobian matrix. The algorithm does not facilitate the re-use of data from previous iterations, and in some cases evaluation of the partial derivatives can be unnecessarily costly.

An alternative is to modify Newton's Method so that *approximate* partial derivatives are used, as in the Secant Method for a single nonlinear equation, since the slightly slower convergence is offset by the improved efficiency of each iteration. However, simply replacing the analytical Jacobian matrix of  $\mathbf{F}$  with a matrix consisting of finite difference approximations of the partial derivatives does not do much to reduce the cost of each iteration, because the cost of solving the system of linear equations is unchanged.

However, because the Jacobian matrix consists of the partial derivatives evaluated at an element of a convergent sequence, intuitively Jacobian matrices from consecutive iterations are "near" one another in some sense, which suggests that it should be possible to cheaply update an approximate Jacobian matrix from iteration to iteration, in such a way that the *inverse* of the Jacobian matrix can be updated efficiently as well.

This is the case when a matrix has the form

$$B = A + \mathbf{u}\mathbf{v}^T,$$

where  $\mathbf{u}$  and  $\mathbf{v}$  are given vectors. This modification of  $A$  to obtain  $B$  is called a *rank-one update*, since  $\mathbf{u}\mathbf{v}^T$ , an *outer product*, has rank one, since every vector in the range of  $\mathbf{u}\mathbf{v}^T$  is a scalar multiple of  $\mathbf{u}$ . To obtain  $B^{-1}$  from  $A^{-1}$ , we note that if

$$A\mathbf{x} = \mathbf{u},$$

then

$$B\mathbf{x} = (A + \mathbf{u}\mathbf{v}^T)\mathbf{x} = (1 + \mathbf{v}^T\mathbf{x})\mathbf{u},$$

which yields

$$B^{-1}\mathbf{u} = \frac{1}{1 + \mathbf{v}^T A^{-1}\mathbf{u}} A^{-1}\mathbf{u}.$$

On the other hand, if  $\mathbf{x}$  is such that  $\mathbf{v}^T A^{-1} \mathbf{x} = 0$ , then

$$BA^{-1} \mathbf{x} = (A + \mathbf{u}\mathbf{v}^T)A^{-1} \mathbf{x} = \mathbf{x},$$

which yields

$$B^{-1} \mathbf{x} = A^{-1} \mathbf{x}.$$

This takes us to the following more general problem: given a matrix  $C$ , we wish to construct a matrix  $D$  such that the following conditions are satisfied:

- $D\mathbf{w} = \mathbf{z}$ , for given vectors  $\mathbf{w}$  and  $\mathbf{z}$
- $D\mathbf{y} = C\mathbf{y}$ , if  $\mathbf{y}$  is orthogonal to a given vector  $\mathbf{g}$ .

In our application,  $C = A^{-1}$ ,  $D = B^{-1}$ ,  $\mathbf{w} = \mathbf{u}$ ,  $\mathbf{z} = 1/(1 + \mathbf{v}^T A^{-1} \mathbf{u})A^{-1} \mathbf{u}$ , and  $\mathbf{g} = A^{-T} \mathbf{v}$ .

To solve this problem, we set

$$D = C + \frac{(\mathbf{z} - C\mathbf{w})\mathbf{g}^T}{\mathbf{g}^T \mathbf{w}}.$$

Then, if  $\mathbf{g}^T \mathbf{y} = 0$ , the second term in the definition of  $D$  vanishes, and we obtain  $D\mathbf{y} = C\mathbf{y}$ , but in computing  $D\mathbf{w}$ , we obtain factors of  $\mathbf{g}^T \mathbf{w}$  in the numerator and denominator that cancel, which yields

$$D\mathbf{w} = C\mathbf{w} + (\mathbf{z} - C\mathbf{w}) = \mathbf{z}.$$

Applying this definition of  $D$ , we obtain

$$B^{-1} = A^{-1} + \frac{\left(\frac{1}{1 + \mathbf{v}^T A^{-1} \mathbf{u}} A^{-1} \mathbf{u} - A^{-1} \mathbf{u}\right) \mathbf{v}^T A^{-1}}{\mathbf{v}^T A^{-1} \mathbf{u}} = A^{-1} - \frac{A^{-1} \mathbf{u} \mathbf{v}^T A^{-1}}{1 + \mathbf{v}^T A^{-1} \mathbf{u}}.$$

This formula for the inverse of a rank-one update is known as the *Sherman-Morrison Formula*.

We now return to the problem of approximating the Jacobian of  $\mathbf{F}$ , and efficiently obtaining its inverse, at each iterate  $\mathbf{x}^{(k)}$ . We begin with an exact Jacobian,  $A_0 = J_{\mathbf{F}}(\mathbf{x}^{(0)})$ , and use  $A_0$  to compute the first iterate,  $\mathbf{x}^{(1)}$ , using Newton's Method. Then, we recall that for the Secant Method, we use the approximation

$$f'(x_1) \approx \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Generalizing this approach to a system of equations, we seek an approximation  $A_1$  to  $J_{\mathbf{F}}(\mathbf{x}^{(1)})$  that has these properties:

- $A_1(\mathbf{x}^{(1)} - \mathbf{x}^{(0)}) = \mathbf{F}(\mathbf{x}^{(1)}) - \mathbf{F}(\mathbf{x}^{(0)})$
- If  $\mathbf{z}^T(\mathbf{x}^{(1)} - \mathbf{x}^{(0)}) = 0$ , then  $A_1 \mathbf{z} = J_{\mathbf{F}}(\mathbf{x}^{(0)}) \mathbf{z} = A_0 \mathbf{z}$ .

It follows from previous discussion that

$$A_1 = A_0 + \frac{\mathbf{y}_1 - A_0 \mathbf{s}_1}{\mathbf{s}_1^T \mathbf{s}_1} \mathbf{s}_1^T,$$

where

$$\mathbf{s}_1 = \mathbf{x}^{(1)} - \mathbf{x}^{(0)}, \quad \mathbf{y}_1 = F(\mathbf{x}^{(1)}) - \mathbf{F}(\mathbf{x}^{(0)}).$$

Furthermore, once we have computed  $A_0^{-1}$ , we have

$$A_1^{-1} = A_0^{-1} - \frac{A_0^{-1} \left( \frac{\mathbf{y}_1 - A_0 \mathbf{s}_1}{\mathbf{s}_1^T \mathbf{s}_1} \mathbf{s}_1^T \right) A_0^{-1}}{1 + \mathbf{s}_1^T A_0^{-1} \left( \frac{\mathbf{y}_1 - A_0 \mathbf{s}_1}{\mathbf{s}_1^T \mathbf{s}_1} \right)} = A_0^{-1} + \frac{(\mathbf{s}_1 - A_0^{-1} \mathbf{y}_1) \mathbf{s}_1^T A_0^{-1}}{\mathbf{s}_1^T A_0^{-1} \mathbf{y}_1}.$$

Then, as  $A_1$  is an approximation to  $J_{\mathbf{F}}(\mathbf{x}^{(1)})$ , we can obtain our next iterate  $\mathbf{x}^{(2)}$  as follows:

$$A_1 \mathbf{s}_2 = -\mathbf{F}(\mathbf{x}^{(1)}), \quad \mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \mathbf{s}_2.$$

Repeating this process, we obtain the following algorithm, which is known as *Broyden's Method*:

```

Choose  $\mathbf{x}^{(0)}$ 
 $A_0 = J_{\mathbf{F}}(\mathbf{x}^{(0)})$ 
 $\mathbf{s}_1 = -A_0^{-1} \mathbf{F}(\mathbf{x}^{(0)})$ 
 $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \mathbf{s}_1$ 
 $k = 1$ 
while not converged do
     $\mathbf{y}_k = \mathbf{F}(\mathbf{x}^{(k)}) - \mathbf{F}(\mathbf{x}^{(k-1)})$ 
     $\mathbf{w}_k = A_{k-1}^{-1} \mathbf{y}_k$ 
     $c = 1 / \mathbf{s}_k^T \mathbf{w}_k$ 
     $A_k^{-1} = A_{k-1}^{-1} + c(\mathbf{s}_k - \mathbf{w}_k) \mathbf{s}_k^T A_{k-1}^{-1}$ 
     $\mathbf{s}_{k+1} = -A_k^{-1} \mathbf{F}(\mathbf{x}^{(k)})$ 
     $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}_{k+1}$ 
     $k = k + 1$ 
end

```

Note that it is not necessary to compute  $A_k$  for  $k \geq 1$ ; only  $A_k^{-1}$  is needed. It follows that no systems of linear equations need to be solved during an iteration; only matrix-vector multiplications are required, thus saving an order of magnitude of computational effort during each iteration compared to Newton's Method.