THE UNIVERSITY OF CHICAGO

Section 6 (short)

Quasi-Newton Methods – Short

(Section 6, Nocedal and Wright)

Mihai Anitescu

6.1 The Secant Method in One-Dimensional Optimization

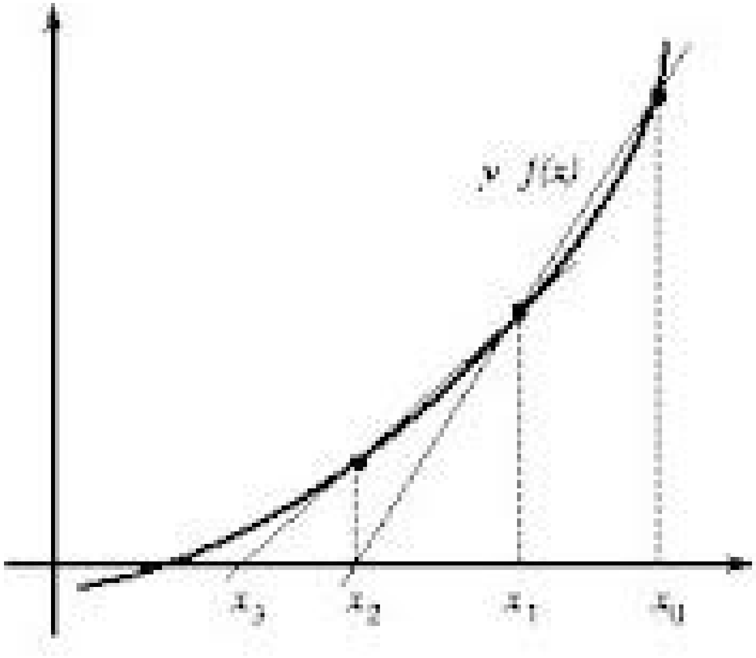


Figure 1 Geometrical illustration of the Secant method.

Newton's Method

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i = 1, 2, \ldots \qquad (1)$$

Approximate the derivative

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \qquad (2)$$

Substituting Equation (2) into Equation (1) gives the Secant method

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

Secant Method

- One can think as approximating second derivative of an optimization problem while computing first order derivative information exactly.
- It is superlinearily convergent. Order of convergence is ~ 1.6
- Question: can we obtain superlinear convergent method for optimization problems while computing gradient information but only approximating hessian information?
- Yes, and that is the subject of quasi-Newton methods.

6.2 Setup and Intuition for Quasi-Newton

- Set up a quadratic model. $m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p.$
- Minimize model (B_k pd). $p_k = -B_k^{-1} \nabla f_k$
- Line search with Wolfe condition $x_{k+1} = x_k + \alpha_k p_k$
- New model:

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p$$

- Match gradient not just at 0, *but also at previous iteration*.
- From p =0, move back the line search and enforce gradient

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k.$$

- Secant condition:

$$\boxed{B_{k+1} \alpha_k p_k = \nabla f_{k+1} - \nabla f_k}$$

The secant equation

- Standard notation $s_k = x_{k+1} - x_k = \alpha_k p_k, \quad y_k = \nabla f_{k+1} - \nabla f_k,$
- Secant equation $B_{k+1} s_k = y_k.$
- If B_{k+1} positive definite, then we need $s_k^T y_k > 0,$
- If Wolfe condition holds:

$$\nabla f(x_{k+1})^T p_k > c_2 \nabla f(x_k)^T p_k \Rightarrow \nabla f(x_{k+1})^T \alpha_k p_k > c_2 \nabla f(x_k)^T \alpha_k p_k$$

$$\Rightarrow \nabla f(x_{k+1})^T s_k > c_2 \nabla f(x_k)^T s_k$$

- Subtract $\nabla f(x_k)^T s_k$ from both sides (remember ...Zoutendijk proof ...)

$$y_k^T s_k = \left(\nabla f(x_{k+1}) - \nabla f(x_k)\right)^T s_k > \left(c_2 - 1\right) \nabla f(x_k)^T s_k > 0$$

- ***If we use Wolfe line search, curvature condition holds!***

The Davidon Fletcher Powell update

- The solution to the “minimum update” problem that is SPD is the DFP update:

$$B_{k+1} = \left(I - \rho_k y_k s_k^T\right) B_k \left(I - \rho_k s_k y_k^T\right) + \rho_k y_k y_k^T \tag{DFP}$$

$$\rho_k = \frac{1}{y_k^T s_k}$$

- If B_k is positive definite, so is B_{k+1}; and it is symmetric and satisfies the secant equation
- It would be nice to have such an update for the inverse.

Inverse of the DFP

- The inverse of matrices using low rank updates can be obtained with the Sherman-Morrison formula:

$$\boxed{\hat{A} = A + UV^T} \rightarrow \boxed{\hat{A}^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1}}$$

- Defining $H_k = B_k^{-1}$ we obtain

$$\text{(DFP)} \qquad H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}.$$

- THIS is what we actually compute.

The dual view

- This generated a *huge* revolution in optimization. Various other updates on the same principle were discovered.
- The most successful is considered to be BFGS: Broyden-Fletcher-Goldfarb-Shanno.
- ***This is obtained by taking the dual view: what if the inverse of the Hessian is what I want to approximate?***
- The inverse of the Hessian satisfies the following secant equation:

$$H_{k+1} y_k = s_k$$

- We look for minimal updates with respect to the metric:

$$\min_H \|H - H_k\|$$

$$\text{subject to} \quad H = H^T, \qquad H y_k = s_k$$

- We now choose $\bar{G}_k$, the average Hessian, to be the scaling matrix

The BFGS formula update

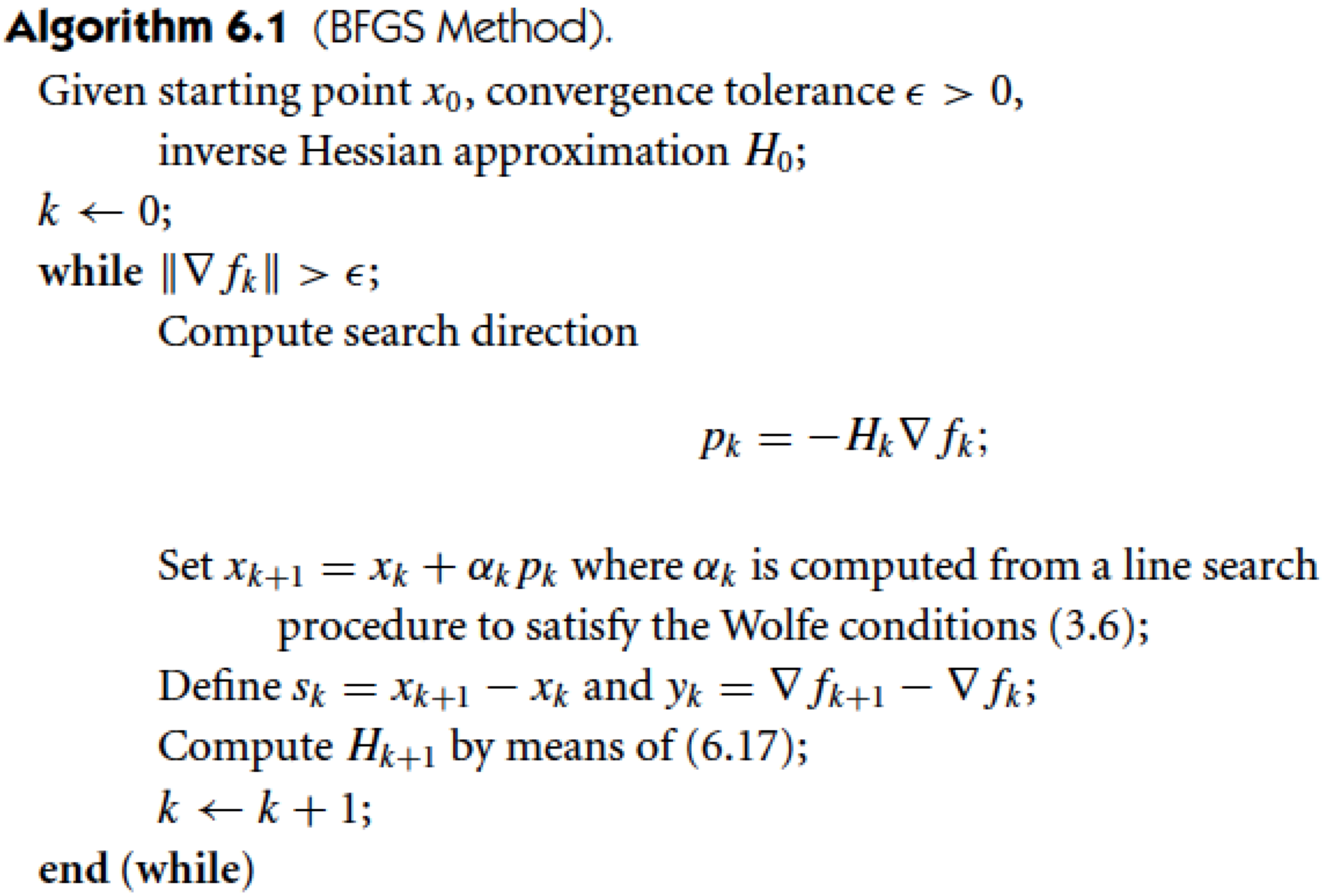
- We then obtain the BFGS update (looks like DFP!!)

$$\text{(BFGS)} \qquad H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$$

- By Sherman-Morrison, it has the inverse (as I sometimes would like to have an approximation of the Hessian itself)

$$\text{(BFGS)} \qquad B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$

A BFGS-based algorithm



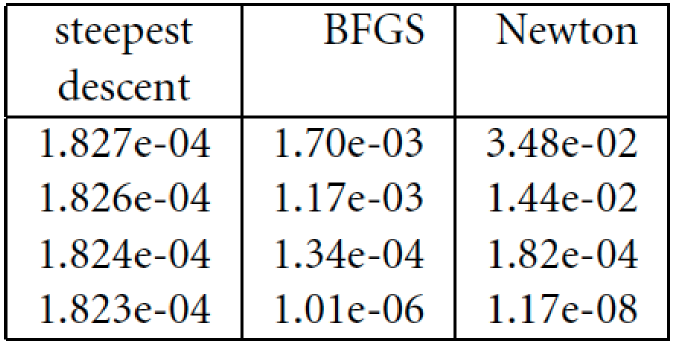
Algorithm 6.1 (BFGS Method).

```
Given starting point $x_0$, convergence tolerance $\epsilon > 0$,
        inverse Hessian approximation $H_0$;
$k \leftarrow 0$;
while $\|\nabla f_k\| > \epsilon$;
        Compute search direction
                $$p_k = -H_k \nabla f_k;$$
        Set $x_{k+1} = x_k + \alpha_k p_k$ where $\alpha_k$ is computed from a line search
                procedure to satisfy the Wolfe conditions (3.6);
        Define $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$;
        Compute $H_{k+1}$ by means of (6.17);
        $k \leftarrow k + 1$;
end (while)
```

Properties of the BFGS algorithm

- It is superlinearily convergent.
- One needs to divide by $y_k^T s_k$ this can be very small. Isn’t this a problem? It turns out, that if we do Wolfe’s line search it tends to self-correct from occasionally small $y_k^T s_k$
- In the Wolfe type search, we tend to choose c_1=1e-4 and c_2=0.9
- If not better idea, it is started with the identity matrix.

- Example superlinear convergence.
- Rosenbrock’s function (from point (-1.2, 1) to 1e-5 gradient norm.
- SD: 5K+ iterations, BFGS 34, Newton 21.



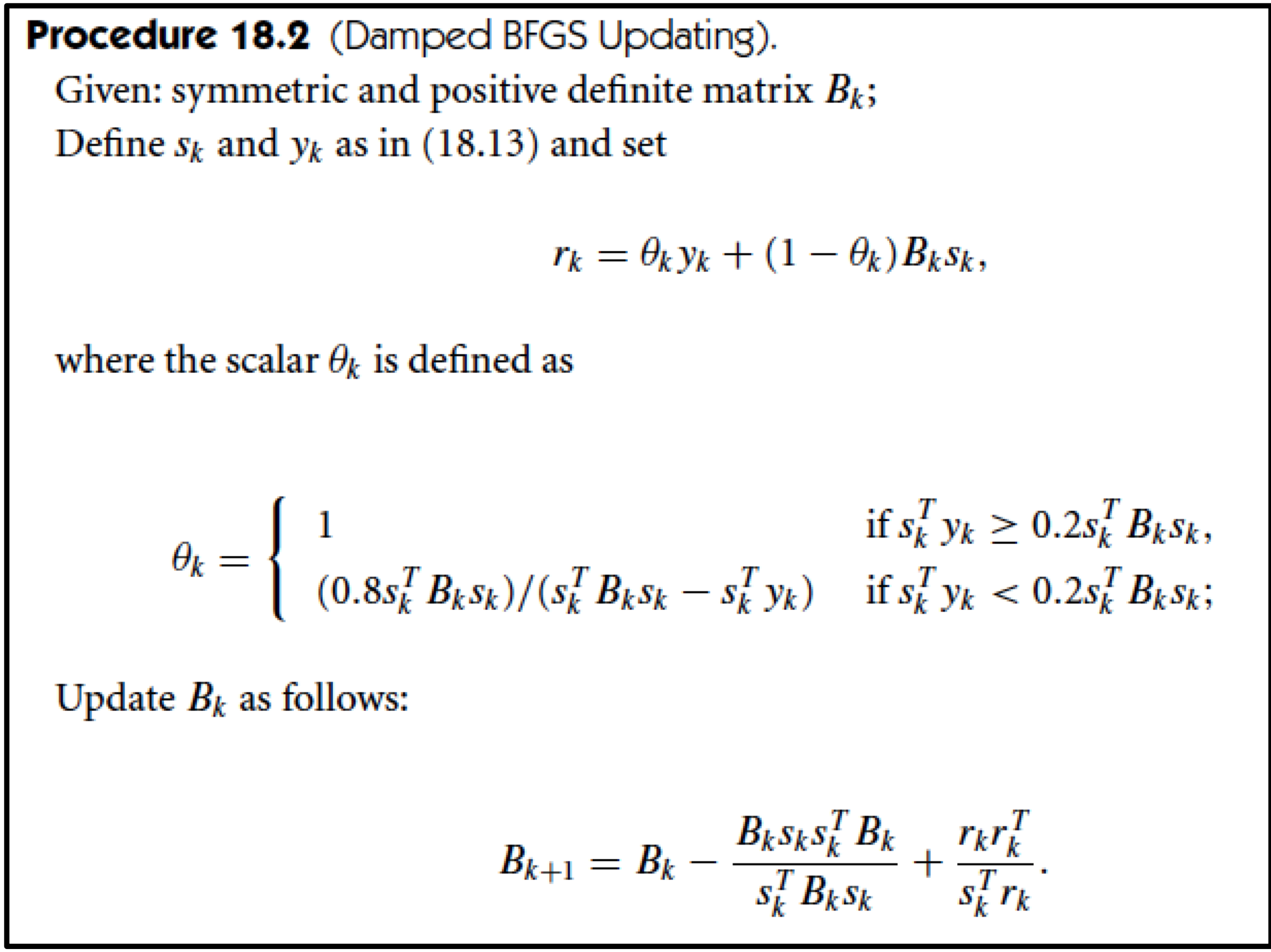
steepest descent	BFGS	Newton
1.827e-04	1.70e-03	3.48e-02
1.826e-04	1.17e-03	1.44e-02
1.824e-04	1.34e-04	1.82e-04
1.823e-04	1.01e-06	1.17e-08

Modification for back-tracking

- Back-tracking *cannot ensure that at the end of the search,* $y_k^T s_k > 0$
- Modify the update by damping:
- Note that

$$s_k^T r_k = 0.2 s_k^T B_k s_k > 0$$

So update is defined and stable



Procedure 18.2 (Damped BFGS Updating).

Given: symmetric and positive definite matrix B_k;

Define s_k and y_k as in (18.13) and set

$$r_k = \theta_k y_k + (1 - \theta_k) B_k s_k,$$

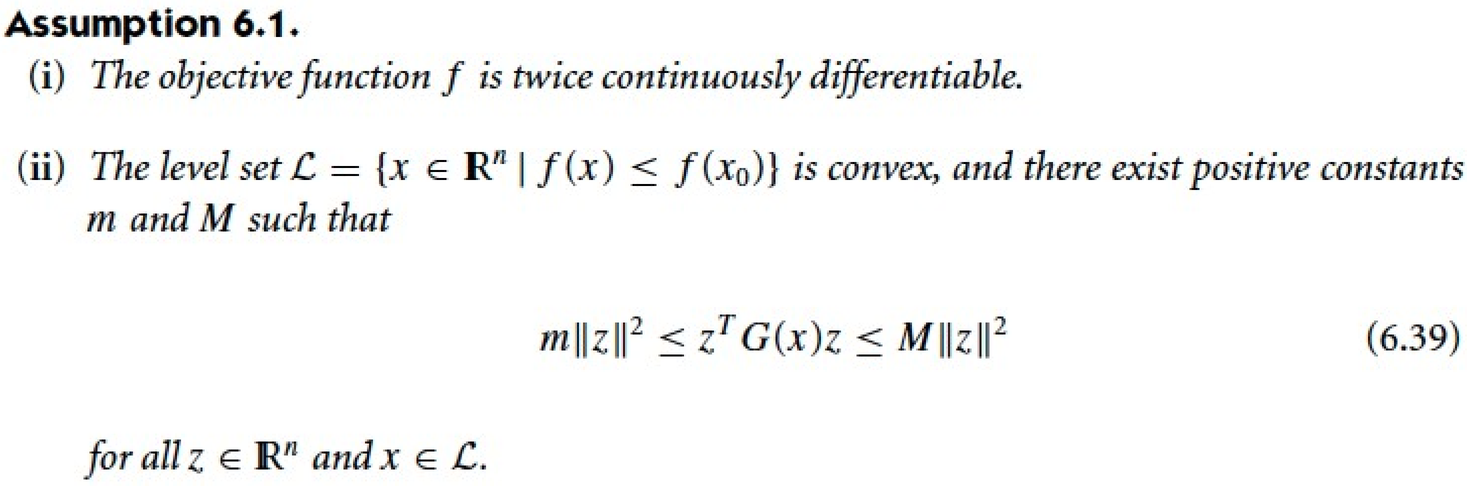
where the scalar θ_k is defined as

$$\theta_k = \begin{cases} 1 & \text{if } s_k^T y_k \geq 0.2 s_k^T B_k s_k, \\ (0.8 s_k^T B_k s_k)/(s_k^T B_k s_k - s_k^T y_k) & \text{if } s_k^T y_k < 0.2 s_k^T B_k s_k; \end{cases}$$

Update B_k as follows:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{r_k r_k^T}{s_k^T r_k}.$$

6.5 Results for BFGS



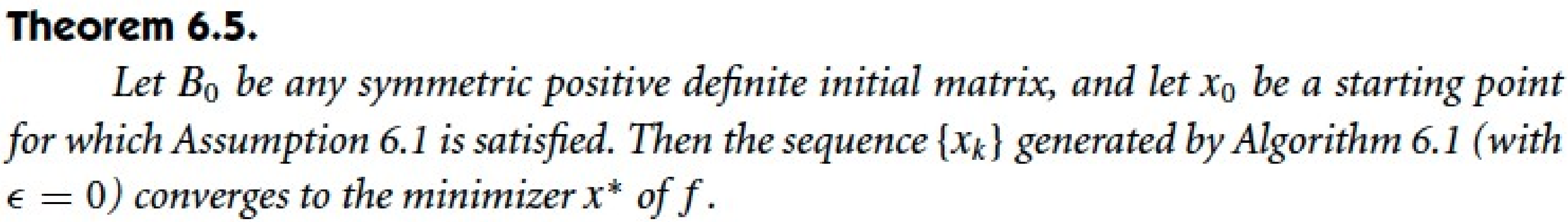
Assumption 6.1.

(i) *The objective function f is twice continuously differentiable.*

(ii) *The level set $\mathcal{L} = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ is convex, and there exist positive constants m and M such that*

$$m\|z\|^2 \leq z^T G(x) z \leq M\|z\|^2 \qquad (6.39)$$

for all $z \in \mathbb{R}^n$ and $x \in \mathcal{L}$.



Theorem 6.5.

Let B_0 be any symmetric positive definite initial matrix, and let x_0 be a starting point for which Assumption 6.1 is satisfied. Then the sequence $\{x_k\}$ generated by Algorithm 6.1 (with $\epsilon = 0$) converges to the minimizer x^* of f.

Advantage of quasi-Newton

- For BFGS ALWAYS positive definite, so line search works fine.
- ***It is hard to prove, but for quadratics it is exactly CG.*** **So CG and BFGS are very related (Theorem 6.4)**
- It needs ONLY gradient information.
- It behaves *almost* like Newton in the limit (convergence is superlinear).
- Optimality is nice, but what you really need is that (1) it needs only derivatives and (2) it satisfies the secant property and (3) if the original matrix is PSD so is the update. These things you should be able to prove.
- In its L-BFGS variant it is the workhorse of weather forecast and operational data assimilation in general (a max likelihood procedure, really).