Numerical Optimization

Lecture Notes #19

Quasi-Newton Methods — Symmetric-Rank-1 / Broyden Class

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Fall 2018

San Diego State University

Outline

The DFP and BFGS Methods: Rank-2 Updates

As we saw last time, the updated matrices H_{k+1} / B_{k+1} differ from their predecessors H_k / B_k by rank-2 updates, *e.g.* for BFGS:

$$H_{k+1} = \left(I - \rho_k \bar{\mathbf{s}}_k \bar{\mathbf{y}}_k^T\right) H_k \left(I - \rho_k \bar{\mathbf{y}}_k \bar{\mathbf{s}}_k^T\right) + \rho_k \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T, \quad \rho_k = \frac{1}{\bar{\mathbf{y}}_k^T \bar{\mathbf{s}}_k},$$

or, expressed in terms of B_{k+1} and B_k (with some help from the Sherman-Morrison-Woodbury formula):

$$B_{k+1} = B_k - \frac{B_k \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T B_k}{\bar{\mathbf{s}}_k^T B_k \bar{\mathbf{s}}_k} + \frac{\bar{\mathbf{y}}_k \bar{\mathbf{y}}_k^T}{\bar{\mathbf{y}}_k^T \bar{\mathbf{s}}_k},$$

where

$$\begin{cases} \bar{\mathbf{s}}_k = \bar{\mathbf{x}}_{k+1} - \bar{\mathbf{x}}_k \equiv \alpha_k \bar{\mathbf{p}}_k \\ \bar{\mathbf{y}}_k = \nabla f(\bar{\mathbf{x}}_{k+1}) - \nabla f(\bar{\mathbf{x}}_k). \end{cases}$$

The Symmetric-Rank-1 (SR1) Method: A Simpler Update-Formula

We can find a much simpler update-formula, of rank 1 which **maintains symmetry** and **satisfies the secant equation**

$$B_{k+1}\bar{\mathbf{s}}_k = \bar{\mathbf{y}}_k, \quad \text{or} \quad H_{k+1}\bar{\mathbf{y}}_k = \bar{\mathbf{s}}_k$$

We can, however, **not** guarantee that B_{k+1} is **positive definite.**

Still, stable and effective numerical algorithms based on SR1 can be developed.

The symmetric-rank-1 update has the form

$$B_{k+1} = B_k + \sigma \bar{\mathbf{v}}\bar{\mathbf{v}}^T, \quad \sigma = \pm 1,$$

and σ and $\bar{\mathbf{v}}$ are chosen so that B_{k+1} satisfies the secant equation.

A Note on the Outer Product $\bar{\mathbf{v}}\bar{\mathbf{v}}^T$ [Supplemental]

If $\bar{\mathbf{v}} \in \mathbb{R}^n$, then

$$A = \bar{\mathbf{v}}\bar{\mathbf{v}}^T = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \cdot \begin{bmatrix} - & \bar{\mathbf{v}}^T & - \end{bmatrix} = \begin{bmatrix} v_1 \cdot - \bar{\mathbf{v}}^T - \\ v_2 \cdot - \bar{\mathbf{v}}^T - \\ \vdots \\ v_n \cdot - \bar{\mathbf{v}}^T - \end{bmatrix} = \begin{bmatrix} | & | & & | \\ v_1\bar{\mathbf{v}} & v_2\bar{\mathbf{v}} & \dots & v_n\bar{\mathbf{v}} \\ | & | & & | \end{bmatrix}.$$

I.e all rows of A are multiples of $\bar{\mathbf{v}}^T$, and

$$\text{eig}(A) = \{\lambda_1, \underbrace{0, \dots, 0}_{n-1 \text{ zeros}}\}, \quad \lambda_1 = \bar{\mathbf{v}}^T\bar{\mathbf{v}} = \|\bar{\mathbf{v}}\|_2^2.$$

Further, the normalized eigenvector corresponding to λ_1 is

$$\bar{\mathbf{u}}_1 = \frac{\bar{\mathbf{v}}}{\|\bar{\mathbf{v}}\|}.$$

San Diego State University

The SR1 Update

Using the update, and the secant equation

$$B_{k+1} = B_k + \sigma \bar{\mathbf{v}}\bar{\mathbf{v}}^T, \qquad B_{k+1}\bar{\mathbf{s}}_k = \bar{\mathbf{y}}_k.$$

We see that

$$\bar{\mathbf{y}}_k = \left[B_k + \sigma \bar{\mathbf{v}}\bar{\mathbf{v}}^T\right]\bar{\mathbf{s}}_k = B_k\bar{\mathbf{s}}_k + \sigma \bar{\mathbf{v}}\bar{\mathbf{v}}^T\bar{\mathbf{s}}_k = B_k\bar{\mathbf{s}}_k + \left[\sigma \bar{\mathbf{v}}^T\bar{\mathbf{s}}_k\right]\bar{\mathbf{v}}.$$

Since $\sigma \bar{\mathbf{v}}^T\bar{\mathbf{s}}_k$ is a scalar, we have

$$\bar{\mathbf{v}} = \delta\left[\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k\right], \quad \text{for some } \delta \in \mathbb{R}.$$

We substitute this back, and get

$$(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k) = \underbrace{\sigma\delta^2\left[\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)\right]}_{\text{A ``complicated'' 1}}(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k).$$

San Diego State University

The SR1 Update

In order to satisfy

$$\sigma\delta^2 \left[\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)\right] = 1,$$

we must have

$$\sigma = \text{sign}\left(\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)\right), \quad \delta = \pm\left|\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)\right|^{-1/2},$$

Hence, the unique SR1 update which satisfies the secant equation is given by

$$B_{k+1} = B_k + \frac{(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)^T}{\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)},$$

or equivalently, by Sherman-Morrison-Woodbury

$$H_{k+1} = H_k + \frac{(\bar{\mathbf{s}}_k - H_k\bar{\mathbf{y}}_k)(\bar{\mathbf{s}}_k - H_k\bar{\mathbf{y}}_k)^T}{\bar{\mathbf{y}}_k^T(\bar{\mathbf{s}}_k - H_k\bar{\mathbf{y}}_k)}.$$

Properties of the SR1 Update

Since the terms in the denominators $\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)$ and $\bar{\mathbf{y}}_k^T(\bar{\mathbf{s}}_k - H_k\bar{\mathbf{y}}_k)$ may be negative, it is possible that B_{k+1} (H_{k+1}) is not positive definite, even though B_k (H_k) is.

This makes **SR1 updates useless for linesearch methods.**

> However, for trust-region methods we can allow indefinite Hessian approximations.

The ability of the SR1 method to generate indefinite Hessian approximations is a **strength** (when leveraged right) — away from the optimum there is nothing guaranteeing that the actual Hessian, $\nabla^2 f(\bar{\mathbf{x}}_k)$, is positive definite.

Properties of the SR1 Update

The main problem that the SR1 method can encounter is that the denominators $\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)$ and $\bar{\mathbf{y}}_k^T(\bar{\mathbf{s}}_k - H_k\bar{\mathbf{y}}_k)$ may be **zero** (small).

If we look at the update for B_{k+1} we have three separate cases:

$\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k) \neq 0$:	There is a unique SR1 update (as described above).
$\bar{\mathbf{y}}_k = B_k\bar{\mathbf{s}}_k$:	The only update formula satisfying the secant equation is $B_{k+1} = B_k$.
$\bar{\mathbf{s}}_k \perp (\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)$:	There is no SR1 update formula which satisfies the secant equation.

The last case is bad news! — It suggests that the SR1 method may break down.

Time to Abandon Ship?

Since SR1 may break down, maybe we should just stick with BFGS (where the updates are all SPD and safe) and call it a day?

It turns out that the SR1 method is still useful:

(i) The breakdown can be easily fixed with a simple safeguard.

(ii) The matrices generated by SR1 tend to be very good approximations of the Hessian matrices — in many cases better than the BFGS approximations. (They're easier to compute too!)

(iii) In some cases (constrained problems, or partially separable functions) it may be impossible to satisfy the curvature condition $\bar{\mathbf{y}}_k^T \bar{\mathbf{s}}_k > 0$, which is required for BFGS updating.

Fixing the SR1 Breakdown

The fix is almost embarrassingly simple: *"If the denominator is too small, do not update!"*

In more detail, the update is only applied when

$$\left| \bar{\mathbf{s}}_k^T (\bar{\mathbf{y}}_k - B_k \bar{\mathbf{s}}_k) \right| \geq r \, \|\bar{\mathbf{s}}_k\| \| \, \bar{\mathbf{y}}_k - B_k \bar{\mathbf{s}}_k \|, \quad \text{for some } r \in (0, 1)$$

A good choice is $r \geq \sqrt{\epsilon_{\text{mach}}} = 10^{-8}$, in our typical floating-point environments.

Not applying the update means $B_{k+1} = B_k$.

Algorithm: The SR1 Trust-Region Method

Algorithm: The SR1 Trust-Region Method

Given starting point $\bar{\mathbf{x}}_0$, convergence tolerance $\epsilon > 0$, initial Hessian approximation B_0, trust-region radius Δ_0, $\eta \in (0, 10^{-3})$ and $r \in (0,1)$:

k = 0
while($\|\nabla f(\bar{\mathbf{x}}_k)\| > \epsilon$)
  $\bar{\mathbf{s}}_k = \arg\min_{\bar{\mathbf{s}}} \nabla f(\bar{\mathbf{x}}_k)^T \bar{\mathbf{s}} + \frac{1}{2}\bar{\mathbf{s}}^T B_k \bar{\mathbf{s}}: \quad \|\bar{\mathbf{s}}\| \leq \Delta_k$
  $\bar{\mathbf{y}}_k = \nabla f(\bar{\mathbf{x}}_{k+1}) - \nabla f(\bar{\mathbf{x}}_k)$
  ared = $f(\bar{\mathbf{x}}_k) - f(\bar{\mathbf{x}}_k + \bar{\mathbf{s}}_k)$
  pred = $-\left[\nabla f(\bar{\mathbf{x}}_k)^T \bar{\mathbf{s}}_k + \frac{1}{2}\bar{\mathbf{s}}_k^T B_k \bar{\mathbf{s}}_k\right]$
  if(ared/pred > η) $\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + \bar{\mathbf{s}}_k$ else $\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k$
  if(ared/pred > 0.75)
    if($\|\bar{\mathbf{s}}_k\| \leq 0.8\Delta_k$) $\Delta_{k+1} = \Delta_k$ else $\Delta_{k+1} = 2 \cdot \Delta_k$
  elseif($0.1 \leq$ ared/pred ≤ 0.75) $\Delta_{k+1} = \Delta_k$
  else $\Delta_{k+1} = \Delta_k / 2$
  if($|\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)| \geq r\,\|\bar{\mathbf{s}}_k\|\|\,\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k\|$)
    $B_{k+1} = B_k + \frac{(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)^T}{\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)}$
  else $B_{k+1} = B_k$
end-while(k = k + 1)

Further Properties of SR1 Updating

The SR1 method generates good Hessian approximations. We take a closer look at applying the SR1 update to a quadratic objective function, with fixed step-length 1, *i.e.*

$$\bar{\mathbf{p}}_k = -H_k \nabla f(\bar{\mathbf{x}}_k), \quad \bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + \bar{\mathbf{p}}_k$$

The following can be shown:

Theorem

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is the strongly convex function $f(\bar{\mathbf{x}}) = \bar{\mathbf{b}}^T \bar{\mathbf{x}} + \frac{1}{2}\bar{\mathbf{x}}^T A \bar{\mathbf{x}}$, where A is a symmetric positive definite matrix. Then for any starting point $\bar{\mathbf{x}}_0$ and symmetric starting matrix H_0, the iterates $\{\bar{\mathbf{x}}_k\}$ generated by the SR1 updates and $\bar{\mathbf{p}}_k = -H_k \nabla f(\bar{\mathbf{x}}_k)$, $\bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + \bar{\mathbf{p}}_k$ converges to the minimizer in at most n steps, provided that $(\bar{\mathbf{s}}_k - H_k \bar{\mathbf{y}}_k)^T \bar{\mathbf{y}}_k \neq 0$ for all k. Moreover, if n steps are performed, and if the search directions p_i are linearly independent, then $H_n = A^{-1}$.

Further Properties of SR1 Updating

For general non-linear objectives, the SR1 updates generate good Hessian approximations as well, but the provable results is a little weaker:

Theorem

Suppose that f is twice continuously differentiable, and the Hessian is bounded and Lipschitz continuous in a neighborhood of a point $\bar{\mathbf{x}}^*$. Let $\{\bar{\mathbf{x}}_k\}$ be any sequence of iterates such that $\bar{\mathbf{x}}_k \to \bar{\mathbf{x}}^*$ for some $\bar{\mathbf{x}}^* \in \mathbb{R}^n$. Suppose in addition that

$$|\bar{\mathbf{s}}_k^T(\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k)| \geq r\,\|\bar{\mathbf{s}}_k\|\|\,\bar{\mathbf{y}}_k - B_k\bar{\mathbf{s}}_k\|$$

holds for all k, for some $r \in (0,1)$, and that the steps $\bar{\mathbf{s}}_k$ are uniformly linearly independent. Then the matrices B_k generated by the SR1 updating formula satisfy*

$$\lim_{k\to\infty} \|B_k - \nabla^2 f(\bar{\mathbf{x}}^*)\| = 0.$$

* See next slide.

Note: "Uniformly Linearly Independent"

Here, "Uniformly Linearly Independent" means, roughly, that the steps do not tend to fall in a subspace of dimension less than n.

This assumption is usually, but not always, satisfied in practice.

The Broyden Class

So far we have looked at three quasi-Newton methods: DFP, BFGS, and SR1.

There are infinitely many more quasi-Newton methods, of which the **Broyden class** is of interest.

The updates of the Broyden class takes the form

$$B_{k+1} = B_k - \frac{B_k \bar{\mathbf{s}}_k \bar{\mathbf{s}}_k^T B_k}{\bar{\mathbf{s}}_k^T B_k \bar{\mathbf{s}}_k} + \frac{\bar{\mathbf{y}}_k \bar{\mathbf{y}}_k^T}{\bar{\mathbf{y}}_k^T \bar{\mathbf{s}}_k} + \Phi_k (\bar{\mathbf{s}}_k^T B_k \bar{\mathbf{s}}_k) \bar{\mathbf{v}}_k \bar{\mathbf{v}}_k^T,$$

where

$$\bar{\mathbf{v}} = \left[\frac{\bar{\mathbf{y}}_k}{\bar{\mathbf{y}}_k^T \bar{\mathbf{s}}_k} - \frac{B_k \bar{\mathbf{s}}_k}{\bar{\mathbf{s}}_k^T B_k \bar{\mathbf{s}}_k} \right],$$

and Φ_k is a scalar.

The Broyden Class

The BFGS and DFP methods are part of the Broyden class:

Broyden Class	Corresponding Method
$\Phi_k = 0$	**BFGS**
$\Phi_k = 1$	**DFP**

We can view the Broyden class as a linear combination of BFGS and DFP

$$B_{k+1}^{\text{Broyden}} = (1 - \Phi_k) B_{k+1}^{\text{BFGS}} + \Phi_k B_{k+1}^{\text{DFP}}$$

Since BFGS and DFP satisfy the secant equation, so does the Broyden class.

Since BFGS and DFP preserve positive definiteness when $\bar{\mathbf{s}}_k^T \bar{\mathbf{y}}_k > 0$, so does the Broyden class, for $0 \leq \Phi_k \leq 1$.

The Restricted Broyden Class, $\Phi_k \in [0, 1]$

Theorem

Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is the strongly convex quadratic function $f(\bar{\mathbf{x}}) = \bar{\mathbf{b}}^T \bar{\mathbf{x}} + \frac{1}{2}\bar{\mathbf{x}}^T A \bar{\mathbf{x}}$, where A is SPD. Let $\bar{\mathbf{x}}_0$ be any starting point for the iteration

$$\bar{\mathbf{p}}_k = -B_k^{-1} \nabla f(\bar{\mathbf{x}}_k), \quad \bar{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_k + \bar{\mathbf{p}}_k$$

and B_0 be any SPD starting matrix, and suppose that the matrices B_k are updated by the Broyden formula with $\Phi_k \in [0, 1]$. Define $\lambda_1^{(k)} \leq \lambda_2^{(k)} \leq \cdots \leq \lambda_n^{(k)}$ to be the eigenvalues of the matrix

$$A^{1/2} B_k^{-1} A^{1/2}$$

Then, for all k we have

$$\min\{\lambda_i^{(k)}, 1\} \leq \lambda_i^{(k+1)} \leq \max\{\lambda_i^{(k)}, 1\}, \quad i = 1, 2, \ldots, n$$

Moreover, this is not true for $\Phi_k \notin [0, 1]$.

The Restricted Broyden Class: What the Theorem Means

If the eigenvalues of the matrix

$$A^{1/2} B_k^{-1} A^{1/2}$$

are all 1, then the quasi-Newton approximation B_k is identical to the Hessian A of the quadratic objective. (This is what we want in a perfect world)

The relation

$$\min\{\lambda_i^{(k)}, 1\} \leq \lambda_i^{(k+1)} \leq \max\{\lambda_i^{(k)}, 1\}, \quad i = 1, 2, \dots, n$$

tells us that the eigenvalues $\{\lambda_i^{(k)}\}$ converge monotonically to 1.

If some $\lambda_i^{(k)} = 0.8$, then we know that $\lambda_i^{(k+1)} \in [0.8, 1]$ — convergence is not strict, but at least we are not moving away from the desired result.

SAN DIEGO STATE UNIVERSITY

Square Roots of SPD Matrices [SUPPLEMENTAL]

- A positive semi-definite matrix, M has a unique positive semi-definite square root, $R = M^{1/2}$.
- When $M = X\Lambda X^{-1} \overset{\text{SPD}}{=} Q\Lambda Q^T$, let $R = QSQ^T$, and

  $$R^2 = (QSQ^T)^2 = QSQ^TQSQ^T = QSSQ^T = QS^2Q^T = M,$$

  showing that

  $$S = \Lambda^{1/2}, \quad \text{and therefore} \quad R = Q\Lambda^{1/2}Q^T$$

- $\exists$ other approaches.

The Broyden Class

The theorem seems to suggest that the best update formulas belong to the restricted Broyden class. However, this has not been established.

On the contrary, computational testing and some analysis suggest that algorithms that allow $\Phi_k < 0$ may outperform BFGS.

We have already seen one example — the SR1 method

Broyden Class	Corresponding Method
$\Phi_k = 0$	BFGS
$\Phi_k = 1$	DFP
$\Phi_k = \frac{\bar{\mathbf{s}}_k^T \bar{\mathbf{y}}_k}{\bar{\mathbf{s}}_k^T \bar{\mathbf{y}}_k - \bar{\mathbf{s}}_k^T B_k \bar{\mathbf{s}}_k}$	SR1

The Broyden Class: Properties

We know that if B_k is SPD, $\bar{\mathbf{y}}_k^T \bar{\mathbf{s}}_k > 0$, and $\Phi_k \geq 0$, then B_{k+1} is also SPD if a restricted Broyden class update is used.

It can be shown that B_{k+1} is SPD for a wider range of Φ_k, including some negative values; B_{k+1} becomes singular (has at least one zero eigenvalue) when Φ_k takes the critical value Φ_k^c

$$\Phi_k^c = \frac{1}{1-\mu_k}, \qquad \mu_k = \frac{(\bar{\mathbf{y}}_k^T B_k^{-1} \bar{\mathbf{y}}_k)(\bar{\mathbf{s}}_k^T B_k \bar{\mathbf{s}}_k)}{(\bar{\mathbf{y}}_k^T \bar{\mathbf{s}}_k)^2}$$

$\mu_k \geq 1$, so $\Phi_k^c \leq 0$.

Hence, if B_0 is SPD, $\bar{\mathbf{y}}_k^T \bar{\mathbf{s}}_k > 0$, and $\Phi_k > \Phi_k^c$, then B_k are all SPD.

When the line search is **exact**, all Broyden class methods with $\Phi_k > \Phi_k^c$ generate the same sequence of iterates $\{\bar{\mathbf{x}}_k\}$. The directions $\bar{\mathbf{p}}_k$ differ only in length, so the exact line searches identify the same $\bar{\mathbf{s}}_k = \alpha_k^* \bar{\mathbf{p}}_k$.

San Diego State University

The Broyden Class: Properties

Theorem

Suppose that a method of the Broyden class is applied to a strongly convex function $f : \mathbb{R}^n \to \mathbb{R}$, where $\bar{\mathbf{x}}_0$ is the starting point and B_0 is any SPD matrix. Assume that α_k is the exact step length and that $\Phi_k \geq \Phi_k^c$ for all k. Then the following statements are true:

(i) *The iterates converge to the solution in at most n iterations.*

(ii) *The secant equation is satisfied for all previous search directions, i.e.*

$$B_k \bar{\mathbf{s}}_j = \bar{\mathbf{y}}_j, \quad j = 1, 2, \ldots, k-1$$

(iii) *If the starting matrix is $B_0 = I$, then the iterates are identical to those generated by the* **conjugate gradient method.** *In particular, the search directions are conjugate, i.e.*

$$\bar{\mathbf{s}}_i^T A \bar{\mathbf{s}}_j = 0, \quad \text{for } i \neq j$$

where A is the Hessian of the quadratic function.

(iv) *If n iterations are performed, then $B_n = A$.*

The Broyden Class: Properties

The results **(i)**, **(ii)**, and **(iv)** echo the result for the SR1 method applied to a strongly convex objective function.

Result **(iii)** may seem a little surprising?!? With an exact line search the Broyden class methods compute the conjugate gradient directions while constructing the Hessian (and/or inverse Hessian)!

If $B_0 \neq I$, then the Broyden class methods generate the iterates of the preconditioned conjugate gradient method $\texttt{PCG}(B_0)$.

These results are mainly theoretical curiosities, since any practical implementation would use inexact line-searches. This causes the performance to differ, sometimes dramatically.

This sort of analysis was, however, the key to development of quasi-Newton methods.

SAN DIEGO STATE UNIVERSITY

Homework #5 — Due 11/16/2018

Implement BFGS (and if you have too much time on your hands `DFP` and `SR1`).

Grab `rosenbrock_2Nd.m` from the class webpage, and use the 18-dimensional initial condition returned by `rosenbrock_2Nd(x,-1)`.

Compare against full Newton optimization — count number of "outer iterations" $x_k \to x_{k+1}$, as well as "inner iterations" (linesearches / trust-region model rebuilds).

Check the Quasi-Newton convergence criteria (from lecture #5)

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(\bar{\mathbf{x}}_k))\bar{\mathbf{p}}_k\|}{\|\bar{\mathbf{p}}_k\|} = 0$$

and things that may be revealed in the future (lecture #20).

Notes on `rosenbrock_2Nd.m`

- $\vec{x}_0$ = `rosenbrock_2Nd(x,-1)` ::
  returns $\vec{x}_0 \in \mathbb{R}^{18}$, to be used as the initial point.
  The argument `x` is ignored.

- `rosenbrock_2Nd(x,0)` ::
  returns $f(\mathtt{x}) \in \mathbb{R}$, for $\mathtt{x} \in \mathbb{R}^{2m}$.

- `rosenbrock_2Nd(x,1)` ::
  returns $\nabla f(\mathtt{x}) \in \mathbb{R}^{2m}$, for $\mathtt{x} \in \mathbb{R}^{2m}$.

- `rosenbrock_2Nd(x,2)` ::
  returns $\nabla^2 f(\mathtt{x}) \in \mathbb{R}^{2m \times 2m}$, for $\mathtt{x} \in \mathbb{R}^{2m}$.

San Diego State University

Index