Unconstrained Optimization V: Quasi-Newton methods (continued)

Note Title 4/23/2022

The BFGS and related Quasi-Newton Methods

William C. Davidon

Read the historical remarks at the beginning of CH 6 of NBW, it's interesting.

In UO_1, I motivated quasi-Newton method as an analog of the secant method in 1-D for solving equations, giving rise to the secant equation

$$B_k(x_k - x_{k-1}) = \nabla f_k - \nabla f_{k-1}. \quad - \text{(Sec)}$$

Let's recall the rationale for (sec): If we want to solve the system of nonlinear eqts $F(x) = 0$, $F: \mathbb{R}^n \to \mathbb{R}^n$, and don't want to compute derivatives. Think:

$$F(x) \approx F(x_k) + \overbrace{DF(x_k)}^{B_k}(x - x_k) = L_k(x), \quad \text{and}$$

require B_k to satisfy $L_k(x_{k-1}) = F(x_{k-1})$, which is equivalent to

$$F(x_k) + B_k(x_{k-1} - x_k) = F(x_{k-1}), \quad \text{or} \quad B_k(x_k - x_{k-1}) = F(x_k) - F(x_{k-1}).$$

this is (sec) when $F = \nabla f$.

In the context of solving equation, B_k shouldn't be constrained to be positive definite or even symmetric. But in the optimization context, a line search of the form:

$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k)$$

can guarantee a decrease in the objective if

(i) $B_k \succ 0$, (ii) α_k is chosen to satisfy Armijo sufficient decrease cond.

[Note: $-B_k^{-1}\nabla f_k$ is also the unique minimizer of $m_k(p) := f_k + \nabla f_k^T p + \frac{1}{2} p^T \overbrace{B_k}^{0} p$.
Indeed, when $\nabla^2 f(x_k) \not\succ 0$, it is a little strange to think of modelling the objective near x_k by a convex quadratic.]

Change k to $k+1$ in (sec): $B_{k+1}\underbrace{(x_{k+1}-x_k)}_{=:s_k} = \underbrace{\nabla f(x_{k+1})-\nabla f(x_k)}_{=:y_k}$ — (sec_{k+1})

When f non-convex, it is by no means obvious that $\exists\, B_{k+1}\succ 0$ s.t. (sec_{k+1}) is satisfied.

(If f is such that $\nabla^2 f(x)\succ 0\ \forall x$, then

$\nabla f(x)-\nabla f(y) = \nabla^2 f(\xi)(x-y)$ for some $\xi\in\overline{xy}$.

so $(x-y)^T(\nabla f(x)-\nabla f(y))>0 \quad \forall x\neq y$.

Then

$\exists B\succ 0$ s.t. $B(x-y)=\nabla f(x)-\nabla f(y)$ (your HW#1).)

NOTE: $\boxed{\exists\, B_{k+1}\succ 0 \text{ s.t. } B_{k+1}s_k = y_k \iff s_k^T y_k > 0}$ ($\Rightarrow$) multiply s_k^T to both sides ($\Leftarrow$) your HW#1

N&W call $s_k^T y_k>0$ the curvation condition, which is confusing, considered the eponymous second Wolfe condition $\nabla f_{k+1}^T p_k \geq c_2 \nabla f_k^T p_k$.

Here's the surprise: If the step size α_k in $x_{k+1}=x_k-\alpha_k B_k^{-1}\nabla f(x_k)$ is chosen s.t. it satisfies the second Wolfe condition, then $s_k^T y_k>0$ is satisfied.

The proof is easy though: $\nabla f_{k+1}^T p_k \geq c_2 \nabla f_k^T p_k$

$$\Rightarrow \nabla f_{k+1}^T (\alpha_k p_k) \geq c_2 \nabla f_k^T (\alpha_k p_k)$$

$$\Rightarrow \underbrace{(\nabla f_{k+1} - \nabla f_k)^T}_{y_k} \underbrace{(\alpha_k p_k)}_{s_k} \geq (c_2 - 1) \nabla f_k^T (\alpha_k p_k) > 0 .$$

So if x_{k+1} is formed by $x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f_k$ with α_k satisfying the second Wolfe condition, the we can always find $B_{k+1} > 0$ that satisfies the secant equation

$$B_{k+1} s_k = y_k \quad \leftarrow n \text{ linear constraints}$$

$\uparrow$ $n(n+1)/2$ degrees of freedom

So there is a $\frac{n(n+1)}{2} - n = \frac{n(n-1)}{2}$ dimensional space of solutions. (The positive-definiteness constraint does not absord these d.o.f.)

To determine B_{k+1} uniquely and cheaply, find B_{k+1} to be a solution close to B_k in some sense, two approaches have been proposed and studied

(i) Find B_{k+1} that solves : $\min_B \| B - B_k \|$ s.t. $B = B^T$, $B s_k = y_k$, $B > 0$.

for some matrix norm $\|\cdot\|$.

(ii) Set $B_{k+1} = B_k +$ (low rank update) st. $B_{k+1} = B_{k+1}^T$, $B_{k+1} s_k = y_k$, $B_{k+1} \succ 0$

To begin, note that any rank 1 update of B_k that satisfies the secant condition must be st.

$$B_{k+1} = B_k + uc^T \quad , \quad (B_k + uc^T) s_k = y_k$$

$$\Downarrow$$

$$uc^T s_k = y_k - B_k s_k \Rightarrow u = \frac{y_k - B_k s_k}{c^T s_k}$$

So $B_{k+1} = B_k + (y_k - B_k s_k) c^T / c^T s_k$.

c has to chosen st. $c^T s_k \neq 0$

Also, c should be chosen so that B_{k+1} is symmetric when B_k is symmetric.

Easy to see that $c = y_k - B_k s_k$, i.e.

$$B_{k+1} = B_k + (y_k - B_k s_k)(y_k - B_k s_k)^T / (y_k - B_k s_k)^T s_k \quad \text{—(SR1)}$$

is the only solution provided $(y_k - B_k s_k)^T s_k \neq 0$.

If $y_k = B_k s_k$, then $B_{k+1} = B_k$ is the only solution.

If $y_k \neq B_k s_k$ but $(y_k - B_k s_k)^T s_k = 0$, then there is no solution.

} See N&W p144

This method, known as symmetric-rank-1, or SR1, has the shortcoming of breaking down in the last case (and related numerical instabilities). Also, there is no way to guarantee $B_{k+1} > 0$ (even when $B_k > 0$.)

But, somewhat surprisingly, with a simple safeguard to adaquately prevent the breakdown and numerical instabilities, the SR-1 method is actually useful in problems with constraints. (more on this later.)

See sec 6.2 for more details on SR-1.

Since rank-1 update cannot fully solves the problem, researchers tried rank 2.

M. Powell had the following idea (1970):
(For notational simplicity, drop the subscripts k. Write B_k as B, B_{k+1} as $\bar{B}$. y_k as y, s_k as s.)

$$\bar{B}^{(1)} = B + (y - Bs)c^T / c^T s \qquad \text{(satisfies secant cond., not symmetric)}$$
$$\bar{B}^{(2)} = \tfrac{1}{2}(\bar{B}^{(1)} + \bar{B}^{(1)T}) \qquad \text{(symmetrized, but doesn't satisfy secant cond.)}$$

Iterate: $\bar{B}^{(2j+1)} = \bar{B}^{(2j)} + (y - \bar{B}^{(2j)}s)c^T / c^T s$
$\bar{B}^{(2j+2)} = \tfrac{1}{2}(\bar{B}^{(2j+1)} + \bar{B}^{(2j+1)T})$

It turns out: $\{\bar{B}^{(j)}\}$ has a limit given by

$$\bar{B} = B + \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T \qquad (P)$$

It looks like a rank-3 update but is actually a rank-2 update. (why?)
Again, c has to be chosen s.t. $c^T s \neq 0$.

Michael Powell
FRS FAA

I'll omit the proof of (P). It is easy to check that $\bar{B}$ in (P) satisfies both (i) the secant condition $\bar{B}s = y$ and (ii) B symmetric $\Rightarrow \bar{B}$ symmetric.

Ex: check (i). ((ii) is trivial.)

The low-rank update approach is connected to the "closest" update approach by the following result:

Thm: Let $B \in \mathbb{R}^{n\times n}$, $B = B^T$, $c, s, y \in \mathbb{R}^n$ with $c^T s \neq 0$.

$\overline{B}$ defined by (P) is the unique solution to:

$$\min \|M(\hat{B} - B)M\|_F \quad \text{s.t.} \quad \hat{B} = \hat{B}^T, \ \hat{B}s = y,$$

where M is any non-singular symmetric matrix s.t. $Mc = M^{-1}s$.

Proof: (P) $\Rightarrow$ $$\overline{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T$$

Take any symmetric $\hat{B}$ with $\hat{B}s = y$, pre- and post-multiply the above by M, and write $z := Mc = M^{-1}s$. Then

$$M(\overline{B} - B)M = \frac{M(\hat{B} - B)sc^T M + Mc((\hat{B} - B)s)^T M}{z^T z} - \frac{((\hat{B} - B)s)^T s}{(z^T z)^2} MCC^T M$$

$$= \frac{M(\hat{B} - B)MM^{-1}sc^T M + Mcs^T M^{-1}M(\hat{B} - B)M}{z^T z} - \frac{s^T M^{-1}M(\hat{B} - B)MM^{-1}s}{(z^T z)^2} MCC^T M$$

$$\overbrace{M(\bar{B}-B)M}^{\bar{E}} = \overbrace{M(\hat{B}-B)M}^{E}\frac{zz^T}{z^Tz} + \frac{zz^T}{z^Tz}\overbrace{M(\hat{B}-B)M}^{E} - \frac{z^T\overbrace{M(\hat{B}-B)M}^{E}z}{(z^Tz)^2}zz^T$$

$$\bar{E} = \frac{Ezz^T + zz^TE}{z^Tz} - \frac{z^TEz}{(z^Tz)^2}zz^T$$

We have

$$\bar{E}z = Ez$$

$$v \perp z \Rightarrow \bar{E}v = \frac{zz^T}{z^Tz}Ev \quad \Rightarrow \|\bar{E}v\|_2 \le \|Ev\|_2$$

So $\|\bar{E}\|_F \le \|E\|_F$, as desired. (Ex: Explain this.)

The uniqueness follows from the strict convexity of $f : \mathbb{R}^{n\times n} \to \mathbb{R}$, $f(A) = \|M(A-B)M\|_F$ restricted to the affine subspace (hence convex) $\{\hat{B} : \hat{B}^T = \hat{B}, \hat{B}s = y\}$.

Q.E.D.

Remark:

One may directly solve the minimization problem in the theorem to retrieve the update formula (P).

An obvious choice of c is $c = s$ (so $M = I$ above), this gives the so-called Powell symmetric Broyden update

$$\overline{B}_{PSB} = B + \frac{(y - Bs)s^T + s(y - Bs)^T}{s^T s} - \frac{(y - Bs)^T s}{(s^T s)^2} ss^T.$$

Problem with this formula: (i) $B \succ 0 \not\Rightarrow \overline{B} \succ 0$, (ii) lacks affine/scale invariance.

Surprisingly, if we choose $c = y$, the resulted update formula — credited to Davidon, Fletcher and Powell — enjoys both properties.

A simple twist of DFP resulted in the BFGS formula, which enjoys the same two properties, on top of being more effective than DFP.

See below. The development is a little long, but it's good to go through it.

Analyzing the $B \succ 0 \Rightarrow \bar{B} \succ 0$ property

Recall $\bar{B} = B + \underbrace{\frac{(y-Bs)c^T + c(y-Bs)^T}{c^Ts} - \frac{(y-Bs)^Ts}{(c^Ts)^2} cc^T}$

$= \underbrace{pc^T}_{1st+3rd} + \underbrace{cq^T}_{2nd}$, or $\underbrace{\tilde{p}c^T}_{1st} + \underbrace{c\tilde{q}}_{2nd+3rd}$, or $\overbrace{v}^{1st+\frac{1}{2}3rd}c^T + c\overbrace{v^T}^{2nd+\frac{1}{2}3rd}$

$v = \frac{y-Bs}{c^Ts} - \frac{1}{2}\frac{(y-Bs)^Ts}{(c^Ts)^2}$

Thm: If $B \succ 0$, then $\bar{B} \succ 0 \iff \det \bar{B} > 0$.

Proof: Of course ($\Rightarrow$) is always true. The converse has to do with rank 2.

($\Leftarrow$): $\bar{B} = B + vc^T + cv^T = B + \frac{1}{2}[(v+c)(v+c)^T - (v-c)(v-c)^T]$

Recall the eigenvalue interlacing thm at the end of uO_3:

Lemma: If A is symmetric with eigenvalues $\lambda_1 \le \cdots \le \lambda_n$, and $A^* = A + \overset{0}{\sigma} uu^T$, then A^* has eigenvalues λ_i^* s.t. $\lambda_1 \le \lambda_1^* \le \lambda_2 \le \cdots \le \lambda_n \le \lambda_n^*$.

See, e.g., Wilkinson "The Algebraic Eigenvalue Problem" (1965)

Applying this result twice to $\bar{B}$ shows that $\bar{B}$ can have at most one non-positive eigenvalue.

But $0 < \det \bar{B}$ = product of eigenvalues of $\bar{B}$,
so all the eigenvalues of $\bar{B}$ must be positive, so $\bar{B} > 0$. Q.E.D.

To see what choices of c yield $\det \bar{B} > 0$, we need an expression for $\det \bar{B}$.

Lemma: For $v, w \in \mathbb{R}^n$, $\det(I + vw^T) = 1 + v^T w$.

Lemma (Sherman-Morrison) Let $u, v \in \mathbb{R}^n$, A non-singular. Then
$A + uv^T$ is non-singular $\iff \sigma := 1 + v^T A^{-1} u \neq 0$.
If $\sigma \neq 0$, then
$$(A + uv^T)^{-1} = A^{-1} - \frac{1}{\sigma} A^{-1} u v^T A^{-1}.$$

(See Dennis-Moré 77 SIAM Rev paper for proofs, they are not hard.)

Lemma: Let $u_i \in \mathbb{R}^n$, $i = 1, 2, 3, 4$. Then
$$\det(I + u_1 u_2^T + u_3 u_4^T) = (1 + u_1^T u_2)(1 + u_3^T u_4) - u_1^T u_4 \, u_2^T u_3.$$

Proof: Assume $u_1^T u_2 \neq -1$, so $I + u_1 u_2^T$ is nonsingular (by lemma above), and

$$I + u_1 u_2^T + u_3 u_4^T = (I + u_1 u_2^T)(I + (I + u_1 u_2^T)^{-1} u_3 u_4^T)$$

Then

$$\det(I + u_1 u_2^T + u_3 u_4^T) = \det(I + u_1 u_2^T) \det(I + \underbrace{(I + u_1 u_2^T)^{-1}}_{I - \frac{u_1 u_2^T}{1 + u_1^T u_2}} u_3 u_4^T)$$

$$= (1 + u_1^T u_2) \det(I + u_3 u_4^T - \underbrace{u_1 u_2^T u_3 u_4^T / (1 + u_1^T u_2)}_{= \frac{u_2^T u_3}{1 + u_1^T u_2} u_1 u_4^T})$$

$$I + (u_3 - \tfrac{u_2^T u_3}{1 + u_1^T u_2} u_1) u_4^T$$

$$= (1 + u_1^T u_2)(1 + (u_3 - \tfrac{u_2^T u_3}{1 + u_1^T u_2} u_1)^T u_4)$$

$$= (1 + u_1^T u_2)(1 + u_3^T u_4 - \tfrac{u_2^T u_3 \, u_1^T u_4}{1 + u_1^T u_2})$$

$$= (1 + u_1^T u_2)(1 + u_3^T u_4) - (u_2^T u_3)(u_1^T u_4).$$

Since the result holds for $u_1^T u_2 \neq -1$, a continuity argument shows that it holds in general.

Q.E.D.

Now the above formula to $\overline{B} = B + \underbrace{\frac{(y-Bs)c^T + c(y-Bs)^T}{c^Ts} - \frac{(y-Bs)^Ts}{(c^Ts)^2} cc^T}_{\text{rank 2}}$.

$$= B\left[I + B^{-1}[\ \text{rank 2}\ ]\right]$$

With some algebra, using the last lemma (tricky):

$$\det\overline{B} = \det B\left[(c^TB^{-1}y)^2 - c^TB^{-1}c\, y^TB^{-1}y + c^TB^{-1}c\, y^Ts\right]/(c^Ts)^2.$$

Since we assume $B \succ 0$, $\sqrt{B}$ exists. Write $v := B^{-\frac{1}{2}}y$, $w = B^{-\frac{1}{2}}c$, then

$$\det\overline{B} = \det B\left[(v^Tw)^2 - \|v\|^2\|w\|^2 + \|w\|^2 y^Ts\right]/\|c^Ts\|^2,$$

So $\overline{B} \succ 0 \iff \det\overline{B} > 0 \iff \|w\|^2 y^Ts > \underbrace{\|v\|^2\|w\|^2 - (v^Tw)^2}$.

≥ 0 by Cauchy-Schwartz

$= 0$ if w is a multiple of v

Recall that the (two different, subtlely related) curvature conditions guarantee $y^Ts > 0$, so

$\overline{B} \succ 0$ is guaranteed if w is a multiple of v

$\iff c$ is a multiple of y

This leads to the DFP update

$$\bar{B}_{DFP} = B + \frac{(y - Bs)y^T + y(y - Bs)^T}{y^T s} - \frac{(y - Bs)^T s}{(y^T s)^2} yy^T$$

$$= B + \frac{y^T s\,[2yy^T - Bsy^T - ys^T B] - y^T s\, yy^T + (s^T Bs)yy^T}{(y^T s)^2}$$

$$= B - \frac{Bsy^T}{y^T s} - \frac{ys^T B}{y^T s} + \frac{y s^T Bs y^T}{(y^T s)^2} + \frac{yy^T}{y^T s}$$

$$= \left(I - \frac{ys^T}{y^T s}\right) B \left(I - \frac{sy^T}{y^T s}\right) + \frac{yy^T}{y^T s}$$

Applying Sherman-Morrison twice to get:

$$\bar{B}_{DFP}^{-1} = B^{-1} + \underbrace{\frac{ss^T}{s^T y} - \frac{B^{-1} y y^T B^{-1}}{y^T B^{-1} y}}_{\text{also rank-2!}}$$

Ex: Prove this formula.
(Do not use the last expression of $\bar{B}_{DFP}$. $I - \frac{ys^T}{y^T s}$ and $I - \frac{sy^T}{y^T s}$ are singular!)

Again: $B > 0$ and $y^T s > 0 \Rightarrow \bar{B}_{DFP}$, $\bar{B}_{DFP}^{-1} > 0$.

Recap:

Assume $B_k \succ 0$ is an approximate Hessian at x_k.
With B_k, $x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f_k$ can be computed.
With α_k chosen to satisfy Wolfe's condition, $s_k^T y_k > 0$
($s_k = x_{k+1} - x_k$, $y_k = \nabla f_{k+1} - \nabla f_k$)

Then the following problem:
Find $B_{k+1} = B_k + (\text{rank-2 update})$ (?) s.t. $B_{k+1} \succ 0$ and $B_{k+1} s_k = y_k$

has the following solution by DFP:

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

What if we aim to approximate/update the inverse Hessian instead of the Hessian? i.e. we ask the following question:

Assume $H_k \succ 0$ is an approximate inverse Hessian at x_k.
Define $x_{k+1} = x_k - \alpha_k H_k \nabla f_k$, α_k satisfies Wolfe's conds, so $s_k^T y_k > 0$.

Find $H_{k+1} = H_k + (\text{low rank update})$ (??) s.t. $H_{k+1} \succ 0$, $H_{k+1} y_k = s_k$.

We already have an answer:

$$H_{k+1}^{DFP} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}$$

Nothing is new.

But wait... Note that the last question has the exact same algebraic structure as the original question but with
y and s interchanged.

What does it mean?

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

$$\downarrow \quad y \leftrightarrow s, \; B \leftrightarrow H$$

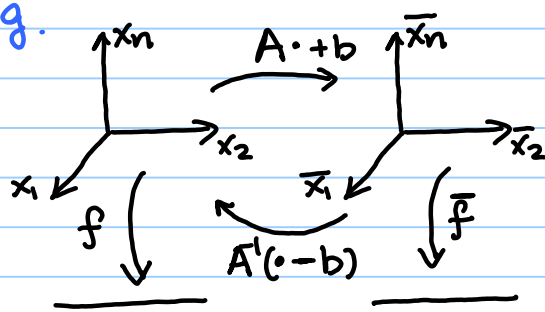
$$H_{k+1}^{BFGS} = \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) H_k \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k}.$$

NOT only is it new, but it happens to outperform DFP!

Yet another surprise, DFP and BFGS methods, similar to Newton's method, have an affine invariant property.

(As mentioned, PSB does not share this property.)

AI guarantees that the ROC is not affected by ill-conditioning.



In the original coordinate system:

$$x_{k+1} = x_k - s_k H_k \nabla f(x_k)$$

In the transformed coordinate system:

$$\bar{x}_{k+1} = \bar{x}_k - \bar{s}_k \bar{H}_k \nabla \bar{f}(\bar{x}_k)$$

$$\downarrow A^{-1}(\cdot - b)$$

$$x_{k+1} = x_k - \bar{s}_k A^{-1} \bar{H}_k A^{-T} \nabla f(x_k)$$

Chain rule:

$$\bar{f}(\bar{x}) = f(A^{-1}(\bar{x}-b))$$
$$d\bar{f}(\bar{x}) = df(A^{-1}(\bar{x}-b))\cdot A^{-1}$$
$$\nabla\bar{f}(\bar{x}) = A^{-T}\nabla f(A^{-1}(\bar{x}-b))$$
$$d\nabla\bar{f}(\bar{x}) = A^{-T} d\nabla f(A^{-1}(\bar{x}-b)) A^{-1}$$

($d\nabla\bar{f}(\bar{x}) = \nabla^2\bar{f}(\bar{x})$, $d\nabla f(A^{-1}(\bar{x}-b)) = \nabla^2 f(A^{-1}(\bar{x}-b))$)

$$[\nabla^2\bar{f}(\bar{x})]^{-1} = A[\nabla^2 f(A^{-1}(\bar{x}-b))]^{-1}A^T$$

? ↑ the answer is affirmative if we can show:

(i) $H_k = A^{-1}\bar{H}_k A^{-T}$ (i.e. H_k satisfies the same transformation law as the true inverse Hessian.)

and

(ii) $\bar{s}_k = s_k \quad \forall k.$

Ex: (i) Assume for instance we have an affine invariant way to pick the initial approx. inverse Hessian, so that $\bar{H}_0 = A H_0 A^T$. (E.g. by choosing H_0 to be the honest Hessian.)

Prove by induction that $\bar{H}_k = A H_k A^T \quad \forall k \geq 0$, for both DFP and BFGS updates.

(ii) seems very difficult as s_k is determined by some complicated algorithm for satisfying the Wolfe conditions.

But the line search function, $\bar{\phi}(\alpha)$, in the transformed coordinates is:

$$\begin{aligned}
\bar{\phi}(\alpha) &= \bar{f}(\bar{x}_k - \alpha \bar{H}_k \nabla \bar{f}(\bar{x}_k)) \\
&= \bar{f}(\bar{x}_k - \alpha A H_k A^T A^{-T} \nabla f(x_k)) \quad \text{by (i) and the transformation law of } \nabla f. \\
&= f(A^{-1}(\bar{x}_k - b - \alpha A H_k \nabla f(x_k)) \\
&= f(x_k - \alpha H_k \nabla f(x_k)) \quad = \phi(\alpha).
\end{aligned}$$

Since the line search function does not change, any algorithm applied to ϕ and

$\bar{\phi}$ would yield the same step size, i.e. $s_k = \bar{s}_k$.

Technical Remarks:

The derivation of DFP and BFGS above follows the 1977 SIAM Review paper by Dennis-Moré. There is no mention of affine invariance in the paper.

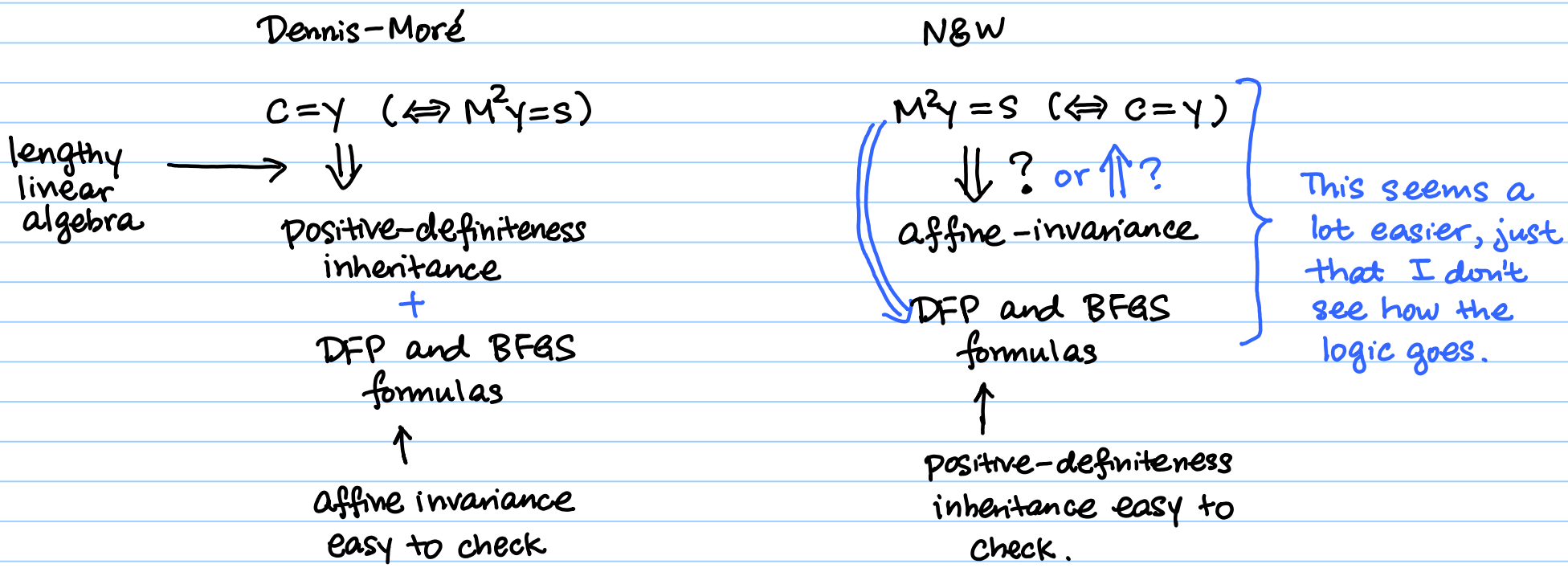
N&W has no mention of Powell's iteration leading to the general formula (P). It only mentions that DFP and BFGS can be obtained by solving the minimization problem $\min_{\hat{B}} \|M(\hat{B}-B)M\|_F$ s.t. $\hat{B}=\hat{B}^T$, $\hat{B}s=y$. — (V)

N&W repeated 3 times (!) in sec 6.1, but with no clear justification, that a choice of a (symmetric nonsingular) M s.t. $M^2y = s$ is necessary for the resulted formula to satisfy affine invariance.

And any such M leads to the DFP formula.

And it is explained in sec 6.1 that the positive-definiteness inheritance property is easy to check (pg 141) once we have the DFP or BFGS formulas.

The derivation of Dennis-Moré is very different. We went through a lengthy linear algebra development to see that the choice of $C=Y$ in (P) — which corresponds to a M with $M^2Y=S$ in the variational formulation (V), according to the theorem on Page 7 above — would imply positive-definiteness inheritance.



Computational Experiments

Let's try different versions of BFGS, GD, Newton on the generalized Rosenbrock function

$$f(x) = \sum_{i=1}^{n-1} \underset{//}{100}\,(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$$

It is non-convex degree 4 polynomial, with a global minimizer at $[1, \cdots, 1]^T$, and at least a local minimizer near $[-1, 1, \cdots, 1]^T$.

See class demo :

- a naive implementation of BFGS with Wolfe line search and $H_0 = I$.

  vs

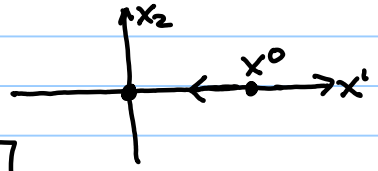
  Matlab's fminunc( )

- empirical global convergence of BFGS (but there is no theorem for it.)

Note: If $H_0 = \beta I$, BFGS and DFP are invariant under rigid transformations (like GD). So the same examples we use to show that GD can get stuck at a saddle point applies to BFGS or DFP.

$$f(x) = x_1^2 - x_2^2$$

$H_0 = \beta I$, $x^0 = \begin{bmatrix} * \\ 0 \end{bmatrix} \Rightarrow$ the BFGS/DFP iterates x^k converge to the saddle pt. $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$.



BFGS and SR-1 methods are known to be remarkable robust in practice, but unfortunately there is no known theoretical result that the iterates of these quasi-Newton methods approach a stationary point from any starting point and any (suitable) initial Hessian approximation.

But a global convergence result can be proved in the convex case.

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*

$$m\|z\|^2 \leq z^T G(x) z \leq M\|z\|^2 \tag{6.39}$$

$G(x) = \nabla^2 f(x)$

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.

↑ BFGS with Wolfe line search, any $H_0 \succ 0$

The proof is a beautiful application of Zoutendijk's result. See NW, sec 6.4.

The setting of the above theorem can also guarantee that the iterates converge fast enough so that

$$\sum_k \|x_k - x^*\| < \infty.$$

(Proof found in Dennis-Moré)

This isn't quite superlinear convergence, but this condition can somehow be "bootstrapped" to show superlinear convergence:

Assumption 6.2.

The Hessian matrix G is Lipschitz continuous at x^*, that is,

$$\|G(x) - G(x^*)\| \leq L\|x - x^*\|,$$

for all x near x^*, where L is a positive constant.

Theorem 6.6.

Suppose that f is twice continuously differentiable and that the iterates generated by the BFGS algorithm converge to a minimizer x^* at which Assumption 6.2 holds. Suppose also that (6.52) holds. Then x_k converges to x^* at a superlinear rate.

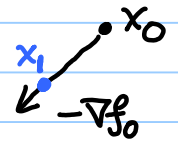
As you may expect, its proof is based on the fundamental result of superlinear convergence by Dennis-Moré presented in UO_4 (last week).

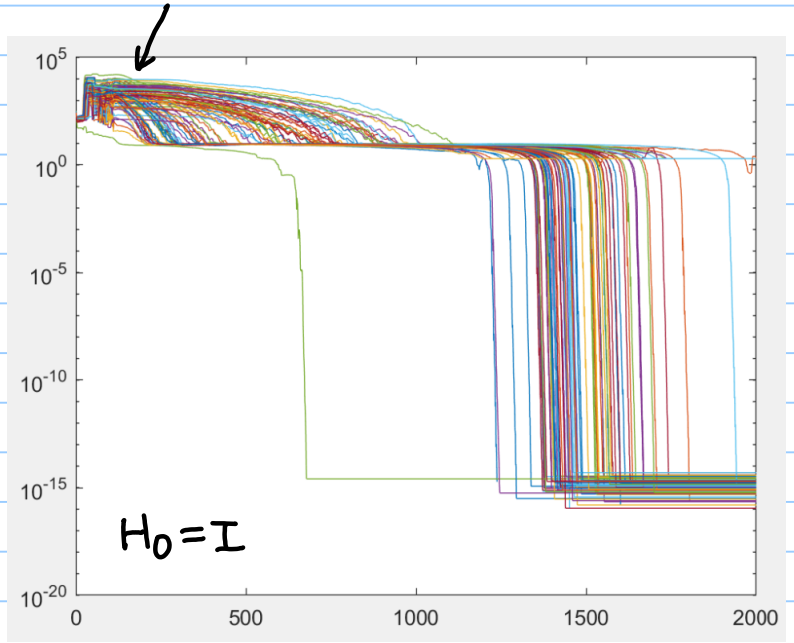
Choice of H_0

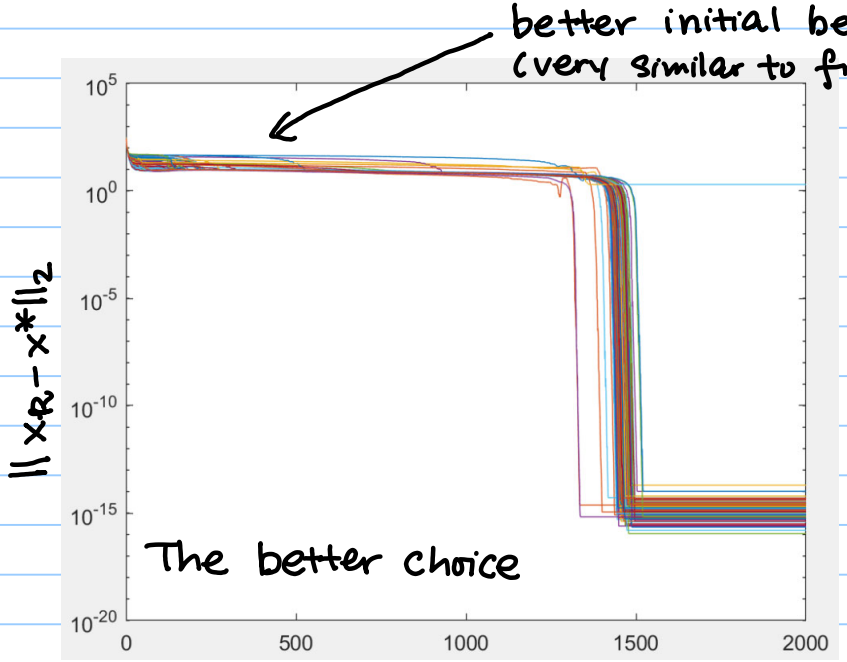
If we set $H_0 = I$, then the first step is gradient descent.

A better choice:

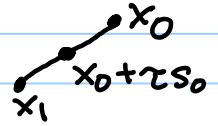
1. Apply one step of GD to get x_1
2. Compute $s_0 = x_1 - x_0$, $y_0 = \nabla f(x_1) - \nabla f(x_0)$
3. set $H_0 \leftarrow (y_0^T s_0 / y_0^T y_0) I$.







Here's the rationale for the choice



Consider $\int_0^1 \underbrace{\nabla^2 f(x_0+\tau s_0)\, s_0}_{=\frac{d}{d\tau}\nabla f(x_0+\tau s_0)}\, d\tau = \underbrace{\nabla f(x_1)-\nabla f(x_0)}_{=y_0}$ $\qquad x_1-x_0=s_0$

Write $\bar{G}_0 := \int_0^1 \nabla^2 f(x_0+\tau s_0)\, d\tau$ = average Hessian of f on $\overline{x_0x_1}$.

Since $\bar{G}_0 s_0 = y_0$, if $\bar{G}_0 \succ 0$ then $\bar{G}_0^{1/2}$ exists, and

$$\frac{y_0^T s_0}{y_0^T y_0} = \frac{s_0^T \bar{G}_0 s_0}{s_0^T \bar{G}_0 \bar{G}_0 s_0} = \frac{s_0^T \bar{G}_0^{1/2} \bar{G}_0^{1/2} s_0}{s_0^T \bar{G}_0^{1/2} \bar{G}_0 \bar{G}_0^{1/2} s_0} = \frac{z^T z}{z^T \bar{G}_0 z}.$$

(set $z := \bar{G}_0^{1/2} s_0$)

$\approx$ one of the eigenvalues of $\bar{G}_0^{-1}$

$\lll$

$\nabla^2 f(x_0)^{-1}$.

So setting $H_0 = \frac{y_0^T s_0}{y_0^T y_0}$ attempts to make the size of H_0 similar to that of $\nabla^2 f(x_0)^{-1}$.

This choice of H_0 is reported to be the most successful (among other choices) in practice. It is also used (in a more dynamic) way in the limited BFGS method, to be presented next.

Limited Memory BFGS

BFGS: $x_{k+1} = x_k - \alpha_k H_k \nabla f_k$

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T \qquad (*)$$

$\rho_k = 1/y_k^T s_k$, $V_k = I - \rho_k y_k s_k^T$, $s_k = x_{k+1} - x_k$, $y_k = \nabla f_{k+1} - \nabla f_k$.

It requires $O(n^2)$ memory to store H_k. ($O(n^3)$ time if you use $(*)$ verbatim, but only $O(n^2)$ time if you use it wisely.)

Note that if we store $s_0, y_0, s_1, y_1, \cdots, s_{k-1}, y_{k-1}$, we can use them to compute $H_k \underbrace{\nabla f_k}_{q}$ recursively as:

$$H_k q = (I - \rho_k y_k s_k^T)^T H_{k-1} \underbrace{(I - \rho_k y_k s_k^T) q}_{q - \rho_k y_k (s_k^T q)} + \rho_k \underbrace{s_k s_k^T q}_{= s_k (s_k^T q)}$$

- $(I - \rho_k y_k s_k^T)^T$: applying it to a vector takes $O(n)$ time
- H_{k-1}: recur! if $H_0 = \beta I$, applying it to a vector also takes $O(n)$ time
- $(I - \rho_k y_k s_k^T) q$: $O(n)$ time

But it is not helpful when $k \approx n$, as storing s_i, y_i, $i=0,\cdots,k-1$ is as costly as storing H_k.

The idea of L-BFGS is to store only s_i, y_i, $i = k-m, \cdots, k-1$, and use them to implicitly represent a modified version of H_k:

$$H_k = (V_{k-1}^T \cdots (V_{k-m+1}^T (V_{k-m}^T H_k^0 V_{k-m} + \rho_{k-m} s_{k-m} s_{k-m}^T) V_{k-m+1} + \rho_{k-m+1} s_{k-m+1} s_{k-m+1}^T) \cdots V_{k-1}) + \rho_{k-1} s_{k-1} s_{k-1}^T$$

↑ some choice of initial inverse Hessian approx. at x_k

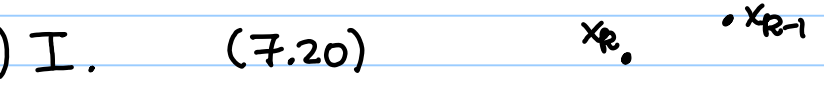
When $k \le m$, this is almost the same H_k as before.

When $k > m$, this is a truncated version of the original H_k.

Also, H_k^0 is reassigned as some (crude) inverse Hessian approx. at x_k,

As in BFGS,

set $H_k^0 = \left(\frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}\right) I$. (7.20)



(Unlike BFGS, this H_k^0 varies with k.)

We may rewrite H_k as

$$H_k = (V_{k-1}^T \cdots V_{k-m}^T) H_k^0 (V_{k-m} \cdots V_{k-1})$$
$$+ \rho_{k-m} (V_{k-1}^T \cdots V_{k-m+1}^T) s_{k-m} s_{k-m}^T (V_{k-m+1} \cdots V_{k-1})$$
$$+ \rho_{k-m+1} (V_{k-1}^T \cdots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \cdots V_{k-1})$$
$$+ \cdots$$
$$+ \rho_{k-1} s_{k-1} s_{k-1}^T .$$

$(\rho_i = 1/y_i^T s_i \ , \ V_i = I - \rho_i y_i s_i^T)$

From this expression, the following efficient procedure for computing $H_k \nabla f_k$ can be derived.

Algorithm 7.4 (L-BFGS two-loop recursion).

$q \leftarrow \nabla f_k$;
for $i = k-1, k-2, \ldots, k-m$
  $\alpha_i \leftarrow \rho_i s_i^T q$;
  $q \leftarrow q - \alpha_i y_i$;
end (for)
$r \leftarrow H_k^0 q$;
for $i = k-m, k-m+1, \ldots, k-1$
  $\beta \leftarrow \rho_i y_i^T r$;
  $r \leftarrow r + s_i(\alpha_i - \beta)$
end (for)
stop with result $H_k \nabla f_k = r$.

It's a bit tricky to see that it does what is supposed to do.

But it is easy to see that it takes $O(mn)$ time, and $O(mn)$ storage.

Algorithm 7.5 (L-BFGS).

Choose starting point x_0, integer $m > 0$;
$k \leftarrow 0$;
repeat
    Choose H_k^0 (for example, by using (7.20));
    Compute $p_k \leftarrow -H_k \nabla f_k$ from Algorithm 7.4;
    Compute $x_{k+1} \leftarrow x_k + \alpha_k p_k$, where α_k is chosen to
        satisfy the Wolfe conditions;
    if $k > m$
        Discard the vector pair $\{s_{k-m}, y_{k-m}\}$ from storage;
    Compute and save $s_k \leftarrow x_{k+1} - x_k$, $y_k = \nabla f_{k+1} - \nabla f_k$;
    $k \leftarrow k + 1$;
until convergence.

see above