Adaptive Matrix Algebras In Unconstrained Optimization

Stefano Cipolla*, C. Di Fiore*, P. Zellini*

*University of Rome "Tor Vergata"
Department of Mathematics

Splines and PDEs: Recent Advances from Approximation Theory to Structured Numerical Linear Algebra
Cetraro (CS) - July 3-7, 2017

The Problem

"In 1963 I attended a meeting at Imperial College, London, where most of the participants agreed that the general algorithms of that time for nonlinear optimization calculations were unlikely to be successful if there were more than 10 variables, unless one had an approximation to the solution in the region of convergence of Newton's method. However, because I had studied the report of Davidson that presented the first variable metric algorithm, I already had a computer program that would calculate least values of functions of up to 100 variables using only function values and first derivatives."

M. J. D. Powell

$$f : \mathbb{R}^n \to \mathbb{R} \text{ lower bounded,}$$

find $\mathbf{x}_*$ such that

$$f(\mathbf{x}_*) = \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}).$$

Zheng, W., Bo, P., Liu, Y., Wang, W. (2012).
Fast B-spline curve fitting by L-BFGS. Computer Aided Geometric Design, 29(7), 448-462.

Matrix Algebras

Let U be a unitary matrix, let us define

$$\mathcal{L} = \{Ud(\mathbf{z})U^H \ : \ \mathbf{z} \in \mathbb{C}^n\} = \text{sd}\, U, \qquad d(\mathbf{z}) = diag(z_1, \dots, z_n).$$

Given $A \in M_n(\mathbb{C})$ let us define

- $\mathcal{L}_A = arg\, \min_{X \in \mathcal{L}} ||X - A||_F$, where $||A||_F = \sum_{r,t=1}^n \overline{a}_{rt} a_{rt}$;

Properties $\mathcal{L}_A$

- $\mathcal{L}_A$ well defined because $\mathcal{L}$ is a closed subspace of $\mathbb{C}^{n\times n}$ (Pythagoras Theorem);
- $\mathcal{L}_A = Ud(\mathbf{z}_A)U^H$ where $[\mathbf{z}_A]_i = [U^H A U]_{ii}, \ i = 1, \dots, n$;
- $A \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{n\times n}$ $(U^H = U^T)$ $\Rightarrow \mathcal{L}_A \in \mathbb{R}^{n\times n}$;
- A S.P.D (Real Symmetric Positive Definite), $U \in \mathbb{R}^{n\times n}$ $(U^H = U^T)$ $\Rightarrow \mathcal{L}_A$ S.P.D;
- $tr\mathcal{L}_A = \sum_i [\mathbf{z}_A]_i = \text{tr}\, A$;
- $det\mathcal{L}_A = \prod_i [\mathbf{z}_A]_i \geq detA$.

$\chi(M)$ number of FLOPS sufficient to perform matrix-vector product Mx, $\mathbf{x} \in \mathbb{C}^n$.

If $L \in \mathcal{L} = sdU$, then $\chi(L) = \chi(U^T) + \chi(U) + n$.

- $\chi(U) = \chi(U^T) = O(n) \Longrightarrow \chi(L) = O(n)$ for all $L \in \mathcal{L}$.

Generalized Broyden Class : Algorithm Structure [2003-2015-2017]

Algorithm 0.1: Generalized Broyden Class

Data: $\mathbf{x}_0 \in \mathbb{R}^n$; $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$;
$\tilde{B}_0 = B_0$ S.P.D., $\mathbf{d}_0 \in \mathbb{R}^n$, $\mathbf{d}_0^T \mathbf{g}_0 < 0$;

for $k = 0, 1 \ldots$ **do**

$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$;

$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$;

$\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$;

$B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$;

$$\begin{cases} \text{Define } \tilde{B}_{k+1} \text{ S.P.D; compute } \mathbf{d}_{k+1} = -\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1} & \mathcal{NS}; \\ \text{Compute } \mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}; \text{ Define } \tilde{B}_{k+1} \text{ S.P.D} & \mathcal{S}; \end{cases}$$

Remarks

- $\tilde{B}_k$ is a S.P.D. approx. of B_k;
- if $\tilde{B}_k = B_k$ for all $k = 0, 1, \ldots$ we obtain classic Broyden Class methods;
- the $\mathcal{NS}$ algorithm and $\mathcal{S}$ algorithms generate sequences $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$,$\{\mathbf{g}_k\}_{k\in\mathbb{N}}$,$\{B_k\}_{k\in\mathbb{N}}$ COMPLETELY DIFFERENT!

Generalized Broyden Class updating formula:

- $\Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k) = \tilde{B}_k + \frac{1}{\mathbf{y}_k^T \mathbf{s}_k}\mathbf{y}_k\mathbf{y}_k^T - \frac{1}{\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k}\tilde{B}_k\mathbf{s}_k\mathbf{s}_k^T\tilde{B}_k + \phi(\mathbf{s}_k^T\tilde{B}_k\mathbf{s}_k)\mathbf{v}_k\mathbf{v}_k^T$,

  where $\phi \in [0, 1)$ is a parameter and $\mathbf{v}_k = \frac{\mathbf{y}_k}{\mathbf{y}_k^T\mathbf{s}_k} - \frac{\tilde{B}_k\mathbf{s}_k}{\mathbf{s}_k^T\tilde{B}_k\mathbf{s}_k}$.

Generalized Broyden Class Updates : Properties

Algorithm 0.2: Generalized B.C.

Data: $\mathbf{x}_0 \in \mathbb{R}^n$; $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$;
$\tilde{B}_0 = B_0$ S.P.D, $\mathbf{d}_0 \in \mathbb{R}^n$, $\mathbf{d}_0^T \mathbf{g}_0 < 0$;

for $k = 0, 1 \ldots$ **do**

$\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$;
$\quad \mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$;
$\quad \mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$;
$\quad B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$;

$$\begin{cases} \boxed{\tilde{B}_{k+1}}; \quad \mathbf{d}_{k+1} = -\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1} & \mathcal{NS}; \\ \mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}; \quad \boxed{\tilde{B}_{k+1}}; & \mathcal{S}; \end{cases}$$

Properties

- $\Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)\mathbf{s}_k = \mathbf{y}_k$
  $B_{k+1}\mathbf{s}_k = \mathbf{y}_k \rightarrow$ $\mathcal{S}$ecant Algorithm;
  $\tilde{B}_{k+1}\mathbf{s}_k \neq \mathbf{y}_k \rightarrow$ $\mathcal{N}$on $\mathcal{S}$ecant Algorithm;
- $\mathbf{g}_k^T \mathbf{d}_k < 0$ and λ_k such that $(0 < c_1 < c_2 < 1)$:
  $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) + c_1 \lambda_k \mathbf{g}_k^T \mathbf{d}_k$
  $\nabla f(\mathbf{x}_k + \lambda_k \mathbf{d}_k) \geq c_2 \mathbf{g}_k^T \mathbf{d}_k$
  $\Downarrow$
  $\mathbf{s}_k^T \mathbf{y}_k > 0$ and $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$.
- $\mathbf{s}_k^T \mathbf{y}_k > 0$ and $\tilde{B}_k$ S.P.D. $\Rightarrow \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$ S.P.D;

$$\begin{cases} \tilde{B}_{k+1} \text{ S.P.D.} \ \Rightarrow \mathbf{g}_{k+1}^T \mathbf{d}_{k+1} < 0 \ (\mathcal{NS}); \\ B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k) \text{ S.P.D.} \ \Rightarrow \mathbf{g}_{k+1}^T \mathbf{d}_{k+1} < 0 \ (\mathcal{S}). \end{cases}$$

Generalized Broyden Class : Complexity

Algorithm 0.3: Generalized B.C.

Data: $\mathbf{x}_0 \in \mathbb{R}^n$; $\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$;
$\tilde{B}_0 = B_0$ S.P.D, $\mathbf{d}_0 \in \mathbb{R}^n$, $\mathbf{d}_0^T \mathbf{g}_0 < 0$;
for $k = 0, 1 \ldots$ **do**
$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$;
$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$;
$\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$;
$B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$;
$$\begin{cases} \boxed{\tilde{B}_{k+1}}; \quad \mathbf{d}_{k+1} = -\tilde{B}_{k+1}^{-1}\mathbf{g}_{k+1} & \mathcal{NS}; \\ \mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1}; \quad \boxed{\tilde{B}_{k+1}}; & \mathcal{S}; \end{cases}$$

Complexity

- if $\tilde{B}_k = B_k$ for all $k = 0, 1 \ldots$ we obtain *Broyden Class* algorithms whose complexity is :
  $O(n^2)$ FLOPS per step;
  $O(n^2)$ memory allocations;
- if $\tilde{B}_k \neq B_k$ algorithm's complexity is :
  ■ Time Complexity per Step :
  - number of FLOPS sufficient to calculate $\tilde{B}_k^{-1}$ where $\tilde{B}_k$ is an approximation of B_k;
  - number of FLOPS sufficient to multiply the matrix $\tilde{B}_k^{-1}$ by a vector;
  - $O(n)$ more FLOPS ;
  ■ Space Complexity :
  - number of memory allocation sufficient to store $\tilde{B}_k^{-1}$;
  - $O(n)$ more memory allocation.

- $B_{k+1}^{-1} = \Psi(\tilde{B}_k^{-1}, \mathbf{s}_k, \mathbf{y}_k)$ = Low Rank Correction of $\tilde{B}_k^{-1}$.
- EX.: $\phi = 0$ (BFGS) $B_{k+1}^{-1} = V_k^T \tilde{B}_k^{-1} V_k + \rho_k \mathbf{s}_k \mathbf{s}_k^T$ where $\rho_k := 1/\mathbf{s}_k^T \mathbf{y}_k$, $V_k := (I - \rho_k \mathbf{y}_k \mathbf{s}_k^T)$.

Global convergence of Generalized $B.C.\mathcal{S}$ with $\phi \in [0, 1)$

Algorithm 0.4: G.B.C.

Data: $\mathbf{x}_0 \in \mathbb{R}^n$;
$\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$,
$\tilde{B}_0 = B_0$ S.P.D,;
$\mathbf{d}_0 \in \mathbb{R}^n$, $\mathbf{d}_0^T \mathbf{g}_0 < 0$;
for $k = 0, 1 \ldots$ **do**
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$;
  $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$;
  $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$;
  $B_{k+1} = \Phi(\tilde{B}_k, \mathbf{s}_k, \mathbf{y}_k)$;
  $\mathbf{d}_{k+1} = -B_{k+1}^{-1} \mathbf{g}_{k+1}$;

$\mathcal{S}$ecant Global Convergence [2015-2017]

If $\tilde{B}_k$ is such that

$$\begin{cases} \mathrm{tr}\, B_k \geq \mathrm{tr}\, \tilde{B}_k & (1a) \\ \det B_k \leq \det \tilde{B}_k & (1b) \\ \dfrac{||B_k \mathbf{s}_k||^2}{(\mathbf{s}_k^T B_k \mathbf{s}_k)^2} \leq \dfrac{||\tilde{B}_k \mathbf{s}_k||^2}{(\mathbf{s}_k^T \tilde{B}_k \mathbf{s}_k)^2}. & (1c) \end{cases}$$

and there exists $M > 0$ such that

$$\frac{||\mathbf{y}_k||^2}{\mathbf{y}_k^T \mathbf{s}_k} \leq M, \qquad (2)$$

then

$$\liminf ||\mathbf{g}_k|| = 0.$$

NOTE 1: (1a) and (1b) are verified if $\tilde{B}_k = \mathcal{L}_{B_k}$ for some $\mathcal{L} =$ sd U. NOTE 2: (2) is verified if f is convex.

Algorithm 0.5: $\mathcal{L}^{(k)}B.C.\mathcal{S}$

Data: $\mathbf{x}_0 \in \mathbb{R}^n$;
$\mathbf{g}_0 = \nabla f(\mathbf{x}_0)$, $\mathcal{L}^{(0)}$;
B_0 S.D.P, $\mathbf{d}_0 \in \mathbb{R}^n$, $\mathbf{d}_0^T \mathbf{g}_0 < 0$;

for $k = 0, 1 \ldots$ **do**
- $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$;
- $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$;
- $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$;
- Define $\mathcal{L}^{(k)}$;
- $B_{k+1} = \Phi(\mathcal{L}^{(k)}_{B_k}, \mathbf{s}_k, \mathbf{y}_k)$;
- $\mathbf{d}_{k+1} = -B_{k+1}^{-1} \mathbf{g}_{k+1}$;

Remark

The choice $\mathcal{L}^{(k)} \equiv \mathcal{L}$ for $k = 0, 1, \ldots$ is allowed...

...BUT CONVERGENCE IS NOT GUARANTEED!

For every

$$\mathcal{L} = \{U d(\mathbf{z}) U^H \ : \ \mathbf{z} \in \mathbb{C}^n\}$$

we have

$$\mathrm{tr}\, B_k = \mathrm{tr}\, \mathcal{L}_{B_k}$$

$$\det B_k \leq \det \mathcal{L}_{B_k}.$$

BUT WHAT ABOUT CONDITION (1c) **?**

IT IS SATISFIED IF $\tilde{B}_k \mathbf{s}_k = B_k \mathbf{s}_k$!

$$\Downarrow$$

ADAPTIVE CHOICE, i.e., find

$\mathcal{L}^{(k)} = \{U_k d(\mathbf{z}) U_k^H \ : \ \mathbf{z} \in \mathbb{C}^n\}$ **s.t.**

$$\boxed{\mathcal{L}^{(k)}_{B_k} \mathbf{s}_k = B_k \mathbf{s}_k}$$

How to guarantee convergence [2017]

Theorem

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. For every fixed integers m and r with $1 \le m \le n$, $mr \le n$ and for any $V_1 \in \mathbb{R}^{n\times r}$ such that $V_1^T V_1 = I_r$, there exists an orthogonal matrix $L \in \mathbb{R}^{n\times n}$ such that if $\mathcal{L} = sd\, L$ and $\mathcal{L}_A$ is the best approximation of A in $\mathcal{L}$, then

$$p_j(\mathcal{L}_A)V_1 = p_j(A)V_1 \tag{3}$$

for any polynomial p_j of degree $j \le m-1$.

THE ORTHOGONAL MATRIX L CAN BE CONSTRUCTED AS THE PRODUCT OF mr HOUSEHOLDER MATRICES!

(L has mr fixed columns)

Corollary

Consider $A \in \mathbb{R}^{n\times n}$ and $V_1 \in \mathbb{R}^{n\times r}$. The computational cost to produce an orthogonal matrix U and $\mathcal{L}_A$ such that $\mathcal{L}_A V_1 = AV_1$, where $\mathcal{L} = sd\, U$ is : $O(n) + 2\chi(A)$ when $m = 2,\ r = 1$ and $O(n) + 4\chi(A)$ when $m = 2,\ r = 2$.

$\mathcal{L}^{(k)}$QN on Quadratic Problems

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \text{ where } f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}, \quad A \text{ S.P.D..}$$

Algorithm 0.6: Generalized BFGS, $H_k = B_k^{-1}$

Data: $\mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{g}_0 = A\mathbf{x}_0 - \mathbf{b}$, H_0 spd,
$\mathbf{d}_0 = -H_0\mathbf{g}_0$, k=0;

for $k = 0, 1 \ldots$ **do**
- $\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k$; `/* ` λ_k ` exact l.s. */`
- $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$;
- $\mathbf{g}_{k+1} = A\mathbf{x}_{k+1} - \mathbf{b}$;
- $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$;
- $\rho_k = 1/\mathbf{s}_k^T\mathbf{y}_k$;
- Define $\tilde{H}_k$ spd ;
- $H_{k+1} = (I - \rho_k\mathbf{s}_k\mathbf{y}_k^T)\tilde{H}_k(I - \rho_k\mathbf{y}_k\mathbf{s}_k^T) + \rho_k\mathbf{s}_k\mathbf{s}_k^T$;
- Set $\mathbf{d}_{k+1} = -H_{k+1}\mathbf{g}_{k+1}$;

Theorem (PCG matching)

Let us consider Algorithm 0.6. If

$$\tilde{H}_k\mathbf{g}_{k+1} = \beta_k H_0\mathbf{g}_{k+1} \text{ where } \beta_k \neq 0,$$

then we have :

$$\mathbf{g}_{k+1}^T\mathbf{s}_j = 0 \text{ for all } j = 0, \ldots, k;$$

$$\mathbf{s}_{k+1}^T A\mathbf{s}_j = 0 \text{ for all } j = 0, \ldots, k;$$

$$span\{\mathbf{s}_0, \ldots, \mathbf{s}_{k+1}\} = span\{H_0\mathbf{g}_0, \ldots, H_0\mathbf{g}_{k+1}\};$$

If $H_0 = I$, find an $\mathcal{L}^{(k)}$ such that

$$\mathcal{L}_{B_k}^{(k)}\mathbf{s}_k = B_k\mathbf{s}_k \text{ and } \mathcal{L}_{B_k}^{(k)}\mathbf{g}_{k+1} = c_k\mathbf{g}_{k+1}.$$

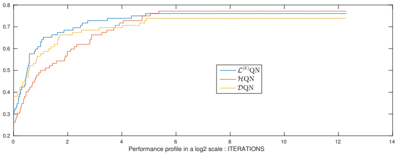
Numerical Results ($\phi = 0$) $\mathcal{L}^{(k)}$QN vs $\mathcal{H}$QN [2003] vs $\mathcal{D}$QN [2007]

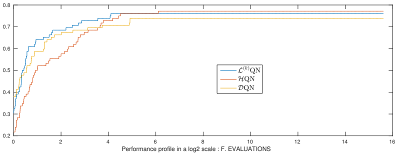
$$\mathcal{P} = \{\text{93 test problems from CUTEST}\}, \ \ \mathcal{S} = \{\mathcal{L}^{(k)}QN, \ \mathcal{H}QN, \ \mathcal{D}QN\}$$

$$r_{p,s} = \frac{t_{p,s}}{\min_{s\in\mathcal{S}}\{t_{p,s}\}} \in [1, r_M], \quad \rho_s(\tau) = \frac{1}{|\mathcal{P}|} size\{p \in \mathcal{P} \mid r_{p,s} \leq \tau\} \ \ [2002]$$

$\rho_s(1)$ probability to win;

$\rho_s^* := \lim_{\tau \to r_M^-} \rho_s(\tau)$ is the probability that the solver solves a problem;





$\rho_s(\tau)$ is the probability for solver $s \in \mathcal{S}$ that a performance ratio $r_{p,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio.

[1] R. H. Byrd, J. Nocedal, Y. Yuan, *Convergence of a Class of Quasi-Newton Methods on Convex Problems*, SIAM Journal of Numerical Analysis, 24-5, 1171-1190 (1987).

[2] E. D. Dolan, J. J. Moré, *Benchmarking optimization software with performance profiles* Mathematical programming 91(2) 201-213 (2002).

[3] C.Di Fiore, S.Fanelli, F.Lepore, P.Zellini, *Matrix algebras in Quasi-Newton methods for unconstrained minimization*, Numerische Mathematik, 94, 479-500 (2003).

[4] A. Bortoletti, C. Di Fiore, S. Fanelli, P. Zellini, *A new class of quasi-Newtonian methods for optimal learning in MLP-networks*, IEEE Transactions on Neural Networks, 14, 263-273 (2003).

[5] Cai, J.F, R.H. Chan, and C. Di Fiore. Minimization of a detail-preserving regularization functional for impulse noise removal. Journal of Mathematical Imaging and Vision 29.1 (2007): 79-91.

[6] S. Cipolla, C. Di Fiore, F. Tudisco, P. Zellini, *Adaptive Matrix Algebras in unconstrained minimization*, Linear Algebra and its Applications, 471, 544-568, (2015).

[7] S. Cipolla, C. Di Fiore, P. Zellini, *Low complexity matrix projections preserving actions on vectors*, submitted, (2017).

[8] S. Cipolla, C. Di Fiore, P. Zellini, *A Quasi Newton-type approach to restricted Broyden Class*, in preparation, (2017).