Matrix structures in optimization: algebras, fast transforms and BFGS methods

Stefano Fanelli

Rome, September 23th 2011

The *phi* operator

Given a matrix H and two vectors $\mathbf{p}, \mathbf{q}$, set:

$$\phi(H, \mathbf{p}, \mathbf{q}) \quad = \quad H + \frac{1}{\mathbf{q}^T \mathbf{p}} \mathbf{q}\mathbf{q}^T - \frac{1}{\mathbf{p}^T H \mathbf{p}} H\mathbf{p}\mathbf{p}^T H \qquad (1)$$

The *phi* operator

Given a matrix H and two vectors $\mathbf{p}, \mathbf{q}$, set:

$$\phi(H, \mathbf{p}, \mathbf{q}) = H + \frac{1}{\mathbf{q}^T\mathbf{p}}\mathbf{q}\mathbf{q}^T - \frac{1}{\mathbf{p}^T H\mathbf{p}} H\mathbf{p}\mathbf{p}^T H \qquad (1)$$

$$\phi(H, \mathbf{p}, \mathbf{q})^{-1} =$$

$$H^{-1} - \frac{1}{\mathbf{q}^T\mathbf{p}}\left(\mathbf{p}\mathbf{q}^T H^{-1} + H^{-1}\mathbf{q}\mathbf{p}^T\right) + \left(1 + \frac{\mathbf{q}^T H^{-1}\mathbf{q}}{\mathbf{q}^T\mathbf{p}}\right)\frac{\mathbf{p}\mathbf{p}^T}{\mathbf{p}^T\mathbf{q}}$$

The *phi* operator

Given a matrix H and two vectors $\mathbf{p}, \mathbf{q}$, set:

$$\phi(H, \mathbf{p}, \mathbf{q}) = H + \frac{1}{\mathbf{q}^T\mathbf{p}}\mathbf{q}\mathbf{q}^T - \frac{1}{\mathbf{p}^T H\mathbf{p}} H\mathbf{p}\mathbf{p}^T H \qquad (1)$$

$$\phi(H, \mathbf{p}, \mathbf{q})^{-1} =$$

$$H^{-1} - \frac{1}{\mathbf{q}^T\mathbf{p}}\left(\mathbf{p}\mathbf{q}^T H^{-1} + H^{-1}\mathbf{q}\mathbf{p}^T\right) + \left(1 + \frac{\mathbf{q}^T H^{-1}\mathbf{q}}{\mathbf{q}^T\mathbf{p}}\right)\frac{\mathbf{p}\mathbf{p}^T}{\mathbf{p}^T\mathbf{q}}$$

ϕ properties:

- H positive definite (pd) and $\mathbf{q}^T\mathbf{p} > 0 \Rightarrow \phi(H, \mathbf{p}, \mathbf{q})$ pd

The *phi* operator

Given a matrix H and two vectors $\mathbf{p}, \mathbf{q}$, set:

$$\phi(H, \mathbf{p}, \mathbf{q}) = H + \frac{1}{\mathbf{q}^T\mathbf{p}}\mathbf{q}\mathbf{q}^T - \frac{1}{\mathbf{p}^T H\mathbf{p}} H\mathbf{p}\mathbf{p}^T H \qquad (1)$$

$$\phi(H, \mathbf{p}, \mathbf{q})^{-1} =$$

$$H^{-1} - \frac{1}{\mathbf{q}^T\mathbf{p}}\left(\mathbf{p}\mathbf{q}^T H^{-1} + H^{-1}\mathbf{q}\mathbf{p}^T\right) + \left(1 + \frac{\mathbf{q}^T H^{-1}\mathbf{q}}{\mathbf{q}^T\mathbf{p}}\right)\frac{\mathbf{p}\mathbf{p}^T}{\mathbf{p}^T\mathbf{q}}$$

ϕ properties:

- H positive definite (pd) and $\mathbf{q}^T\mathbf{p} > 0 \Rightarrow \phi(H, \mathbf{p}, \mathbf{q})$ pd
- $\phi(H, \mathbf{p}, \mathbf{q})\mathbf{p} = \mathbf{q}$ ϕ satisfies the Secant Equation

$$f(\mathbf{x}_*) = \min_{\mathbf{x} \in R^n} f(\mathbf{x}), \qquad \text{find } \mathbf{x}_* \tag{2}$$

BFGS-methods
generate a minimizing sequence $\{\mathbf{x}_k\}_{k=0}^{+\infty}$ by the iterative scheme:

$$f(\mathbf{x}_*) = \min_{\mathbf{x} \in R^n} f(\mathbf{x}), \qquad \text{find } \mathbf{x}_* \tag{2}$$

BFGS-methods

generate a minimizing sequence $\{\mathbf{x}_k\}_{k=0}^{+\infty}$ by the iterative scheme:

$$\mathbf{x}_0 \in R^n, \quad \mathbf{g}_0 = \nabla f(\mathbf{x}_0), \quad \mathbf{d}_0 = -\mathbf{g}_0, \qquad \text{define } B_0 \ \ pd$$

$$f(\mathbf{x}_*) = \min_{\mathbf{x} \in R^n} f(\mathbf{x}), \quad \text{find } \mathbf{x}_* \tag{2}$$

BFGS-methods

generate a minimizing sequence $\{\mathbf{x}_k\}_{k=0}^{+\infty}$ by the iterative scheme:

$\mathbf{x}_0 \in R^n, \quad \mathbf{g}_0 = \nabla f(\mathbf{x}_0), \quad \mathbf{d}_0 = -\mathbf{g}_0, \quad$ define $B_0 \;\; pd$

For $k = 0, 1, \ldots$

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \end{cases}$$

$$f(\mathbf{x}_*) = \min_{\mathbf{x} \in R^n} f(\mathbf{x}), \quad \text{find } \mathbf{x}_* \tag{2}$$

BFGS-methods

generate a minimizing sequence $\{\mathbf{x}_k\}_{k=0}^{+\infty}$ by the iterative scheme:

$\mathbf{x}_0 \in R^n, \quad \mathbf{g}_0 = \nabla f(\mathbf{x}_0), \quad \mathbf{d}_0 = -\mathbf{g}_0, \quad$ define $B_0 \;\; pd$

For $k = 0, 1, \ldots$

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \\ \mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1}) \end{cases}$$

$$f(\mathbf{x}_*) = \min_{\mathbf{x} \in R^n} f(\mathbf{x}), \quad \text{find } \mathbf{x}_* \tag{2}$$

BFGS-methods

generate a minimizing sequence $\{\mathbf{x}_k\}_{k=0}^{+\infty}$ by the iterative scheme:

$$\mathbf{x}_0 \in R^n, \quad \mathbf{g}_0 = \nabla f(\mathbf{x}_0), \quad \mathbf{d}_0 = -\mathbf{g}_0, \quad \text{define } B_0 \ \ pd$$

$$For \ \ k = 0, 1, \ldots$$

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \\ \mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1}) \\ \qquad \varphi(B, \mathbf{s}, \mathbf{y}) = B + \frac{1}{\mathbf{y}^T \mathbf{s}} \mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T B \mathbf{s}} B\mathbf{s}\mathbf{s}^T B \end{cases}$$

$$f(\mathbf{x}_*) = \min_{\mathbf{x}\in R^n} f(\mathbf{x}), \quad \text{find } \mathbf{x}_* \tag{2}$$

BFGS-methods

generate a minimizing sequence $\{\mathbf{x}_k\}_{k=0}^{+\infty}$ by the iterative scheme:

$$\mathbf{x}_0 \in R^n, \quad \mathbf{g}_0 = \nabla f(\mathbf{x}_0), \quad \mathbf{d}_0 = -\mathbf{g}_0, \quad \text{define } B_0 \ \ pd$$

$$For \ \ k = 0, 1, \ldots$$

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \\ \mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1}) \\ \qquad \varphi\,(B, \mathbf{s}, \mathbf{y}) = B + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T B\mathbf{s}} B\mathbf{s}\mathbf{s}^T B \\ B_{k+1} = \varphi\,(B_k,\ \mathbf{s}_k,\ \mathbf{y}_k), \quad \mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k,\ \mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k \end{cases}$$

$$f(\mathbf{x}_*) = \min_{\mathbf{x}\in R^n} f(\mathbf{x}), \quad \text{find } \mathbf{x}_* \tag{2}$$

BFGS-methods

generate a minimizing sequence $\{\mathbf{x}_k\}_{k=0}^{+\infty}$ by the iterative scheme:

$$\mathbf{x}_0 \in R^n, \quad \mathbf{g}_0 = \nabla f(\mathbf{x}_0), \quad \mathbf{d}_0 = -\mathbf{g}_0, \quad \text{define } B_0 \ \ pd$$

For $k = 0, 1, \ldots$

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \\ \mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1}) \\ \qquad \varphi(B, \mathbf{s}, \mathbf{y}) = B + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T B\mathbf{s}} B\mathbf{s}\mathbf{s}^T B \\ B_{k+1} = \varphi(B_k, \mathbf{s}_k, \mathbf{y}_k), \quad \mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k, \ \mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k \\ \mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1} \qquad \leftarrow \textit{descent direction} \end{cases}$$

$$f(\mathbf{x}_*) = \min_{\mathbf{x}\in R^n} f(\mathbf{x}), \quad \text{find } \mathbf{x}_* \tag{2}$$

BFGS-methods

generate a minimizing sequence $\{\mathbf{x}_k\}_{k=0}^{+\infty}$ by the iterative scheme:

$$\mathbf{x}_0 \in R^n, \quad \mathbf{g}_0 = \nabla f(\mathbf{x}_0), \quad \mathbf{d}_0 = -\mathbf{g}_0, \quad \text{define } B_0 \ \ pd$$

For $k = 0, 1, \ldots$

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \\ \mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1}) \\ \qquad \varphi(B, \mathbf{s}, \mathbf{y}) = B + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T B\mathbf{s}} B\mathbf{s}\mathbf{s}^T B \\ B_{k+1} = \varphi(B_k, \mathbf{s}_k, \mathbf{y}_k), \quad \mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k, \ \mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k \\ \mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1} \qquad \leftarrow \textit{descent direction} \\ \lambda_k > 0 \ \text{ such that } f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k) \text{ and } \mathbf{s}_k^T\mathbf{y}_k > 0 \end{cases}$$

$$f(\mathbf{x}_*) = \min_{\mathbf{x}\in R^n} f(\mathbf{x}), \quad \text{find } \mathbf{x}_* \tag{2}$$

BFGS-methods

generate a minimizing sequence $\{\mathbf{x}_k\}_{k=0}^{+\infty}$ by the iterative scheme:

$$\mathbf{x}_0 \in R^n, \quad \mathbf{g}_0 = \nabla f(\mathbf{x}_0), \quad \mathbf{d}_0 = -\mathbf{g}_0, \quad \text{define } B_0 \ \ pd$$

For $k = 0, 1, \ldots$

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \\ \mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1}) \\ \qquad \varphi(B, \mathbf{s}, \mathbf{y}) = B + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T B\mathbf{s}} B\mathbf{s}\mathbf{s}^T B \\ B_{k+1} = \varphi(B_k, \mathbf{s}_k, \mathbf{y}_k), \quad \mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k, \ \mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k \\ \mathbf{d}_{k+1} = -B_{k+1}^{-1}\mathbf{g}_{k+1} \qquad \leftarrow \textit{descent direction} \\ \lambda_k > 0 \ \text{ such that } f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k) \text{ and } \mathbf{s}_k^T\mathbf{y}_k > 0 \end{cases}$$

$$\mathbf{s}_k^T\mathbf{y}_k > 0 \quad \Leftrightarrow \quad \nabla f(\mathbf{x_k}) \ \text{ monotone operator } \ \forall k \ \Rightarrow$$
$$\Rightarrow \qquad \text{f algorithmically convex}$$

Secant Methods

Let A_{k+1} be pd and assume $\forall k \quad \mathbf{d}^{(k+1)} = -A_{k+1}^{-1} \nabla f(\mathbf{x}^{(k+1)})$ be descent directions for a BFGS-method. Then, the method is called *secant* if A_{k+1} solves the *secant equation*:

$$A_{k+1}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}) \qquad (3)$$

Secant Methods

Let A_{k+1} be pd and assume $\forall k \quad \mathbf{d}^{(k+1)} = -A_{k+1}^{-1}\nabla f(\mathbf{x}^{(k+1)})$ be descent directions for a BFGS-method. Then, the method is called *secant* if A_{k+1} solves the *secant equation*:

$$A_{k+1}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}) \tag{3}$$

(3) is the n-dimensional generalization of classical 1-dimensional secant method to compute the zeroes of the derivative of a function $F(x) \in C^1(R^1)$, i.e.:

$$x_{k+1} = \frac{F'(x_k)x_{k-1} - F'(x_{k-1})x_k}{F'(x_k) - F'(x_{k-1})} \tag{4}$$

Secant Methods

Let A_{k+1} be pd and assume $\forall k \quad \mathbf{d}^{(k+1)} = -A_{k+1}^{-1}\nabla f(\mathbf{x}^{(k+1)})$ be descent directions for a BFGS-method. Then, the method is called *secant* if A_{k+1} solves the *secant equation*:

$$A_{k+1}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}) \tag{3}$$

(3) is the n-dimensional generalization of classical 1-dimensional secant method to compute the zeroes of the derivative of a function $F(x) \in C^1(R^1)$, i.e.:

$$x_{k+1} = \frac{F'(x_k)x_{k-1} - F'(x_{k-1})x_k}{F'(x_k) - F'(x_{k-1})} \tag{4}$$

(4) can be rewritten, in fact, in the following way:

$$\begin{cases} x_{k+1} = x_k - \frac{F'(x_k)}{a_k} \\ a_k(x_k - x_{k-1}) = F'(x_k) - F'(x_{k-1}) \end{cases}$$

From BFGS to BFGS-type

BFGS method: $B_k \longrightarrow B_{k+1} = \phi(B_k, \mathbf{s}_k, \mathbf{y}_k)$

(superlinear convergence, $O(n^2)$ complexity)

From BFGS to BFGS-type

BFGS method: $B_k \longrightarrow B_{k+1} = \phi(B_k, \mathbf{s}_k, \mathbf{y}_k)$

(superlinear convergence, $O(n^2)$ complexity)

BFGS-type method: $B_k \longrightarrow \underline{\tilde{B}_k} \longrightarrow B_{k+1} = \phi(\underline{\tilde{B}_k}, \mathbf{s}_k, \mathbf{y}_k)$

From BFGS to BFGS-type

BFGS method: $B_k \longrightarrow B_{k+1} = \phi(B_k, \mathbf{s}_k, \mathbf{y}_k)$

(superlinear convergence, $O(n^2)$ complexity)

BFGS-type method: $B_k \longrightarrow \underline{\tilde{B}_k} \longrightarrow B_{k+1} = \phi(\underline{\tilde{B}_k}, \mathbf{s}_k, \mathbf{y}_k)$

Memory-less BFGS: $\tilde{B}_k = I$

From BFGS to BFGS-type

BFGS method: $B_k \longrightarrow B_{k+1} = \phi(B_k, \mathbf{s}_k, \mathbf{y}_k)$

(superlinear convergence, $O(n^2)$ complexity)

BFGS-type method: $B_k \longrightarrow \underline{\tilde{B}_k} \longrightarrow B_{k+1} = \phi(\underline{\tilde{B}_k}, \mathbf{s}_k, \mathbf{y}_k)$

Memory-less BFGS: $\tilde{B}_k = I$

Limited memory BFGS (L-BFGS):

From BFGS to BFGS-type

BFGS method: $B_k \longrightarrow B_{k+1} = \phi(B_k, \mathbf{s}_k, \mathbf{y}_k)$

(superlinear convergence, $O(n^2)$ complexity)

BFGS-type method: $B_k \longrightarrow \underline{\tilde{B}_k} \longrightarrow B_{k+1} = \phi(\underline{\tilde{B}_k}, \mathbf{s}_k, \mathbf{y}_k)$

Memory-less BFGS: $\tilde{B}_k = I$

Limited memory BFGS (L-BFGS): $\tilde{B}_k = \delta_k I - Q_k \Lambda_k^{-1} Q_k^T$,
$\delta_k = \frac{\mathbf{q}^T \mathbf{q}}{\mathbf{p}^T \mathbf{q}}$, Q_k $n \times 2m$, Λ_k $2m \times 2m$ depending upon
$\mathbf{p}_i, \mathbf{q_i}, i = k-1, \ldots, k-m$.

From BFGS to BFGS-type

BFGS method: $B_k \longrightarrow B_{k+1} = \phi(B_k, \mathbf{s}_k, \mathbf{y}_k)$

(superlinear convergence, $O(n^2)$ complexity)

BFGS-type method: $B_k \longrightarrow \underline{\tilde{B}_k} \longrightarrow B_{k+1} = \phi(\underline{\tilde{B}_k}, \mathbf{s}_k, \mathbf{y}_k)$

Memory-less BFGS: $\tilde{B}_k = I$

Limited memory BFGS (L-BFGS): $\tilde{B}_k = \delta_k I - Q_k \Lambda_k^{-1} Q_k^T$,
$\delta_k = \frac{\mathbf{q}^T \mathbf{q}}{\mathbf{p}^T \mathbf{q}}$, Q_k $n \times 2m$, Λ_k $2m \times 2m$ depending upon
$\mathbf{p}_i, \mathbf{q_i}, i = k-1, \ldots, k-m$.

$\mathcal{LQN}$: $\tilde{B}_k =$ good approximation of B_k

Local Optimization BFGS-type algorithms

Given an approssimation B_k of $\nabla^2 f(\mathbf{w}_k)$, let us define the matrix $\mathcal{L}^U_{B_k}$:

$$\|\mathcal{L}^U_{B_k} - B_k\|_{Fr.} = \min_{X \in \mathcal{L}} \|X - B_k\|_{Fr.}, \ \ \|\cdot\|_{Fr.} = Frob.norm$$

where $\mathcal{L}^U \subset \mathbb{C}^{n \times n}$= algebra of matrices simultaneously diagonalized by a fast unitary transform U.

Local Optimization BFGS-type algorithms

Given an approssimation B_k of $\nabla^2 f(\mathbf{w}_k)$, let us define the matrix $\mathcal{L}^U_{B_k}$:

$$\|\mathcal{L}^U_{B_k} - B_k\|_{Fr.} = \min_{X \in \mathcal{L}} \|X - B_k\|_{Fr.}, \ \ \|\cdot\|_{Fr.} = Frob.norm$$

where $\mathcal{L}^U \subset \mathbb{C}^{n\times n}$= algebra of matrices simultaneously diagonalized by a fast unitary transform U. One can define descent methods $\mathcal{L}QN$ [DFLZ]:

Local Optimization BFGS-type algorithms

Given an approssimation B_k of $\nabla^2 f(\mathbf{w}_k)$, let us define the matrix $\mathcal{L}^U_{B_k}$:

$$\|\mathcal{L}^U_{B_k} - B_k\|_{Fr.} = \min_{X \in \mathcal{L}} \|X - B_k\|_{Fr.}, \ \ \|\cdot\|_{Fr.} = Frob.norm$$

where $\mathcal{L}^U \subset \mathbb{C}^{n\times n}$= algebra of matrices simultaneously diagonalized by a fast unitary transform U. One can define descent methods $\mathcal{L}QN$ [DFLZ]:

$\mathbf{x}_0 \in R^n, \quad \mathbf{d}_0 = -\mathbf{g}_0$
$For \ \ k = 0, 1, \ldots$

Local Optimization BFGS-type algorithms

Given an approssimation B_k of $\nabla^2 f(\mathbf{w}_k)$, let us define the matrix $\mathcal{L}^U_{B_k}$:

$$\|\mathcal{L}^U_{B_k} - B_k\|_{Fr.} = \min_{X \in \mathcal{L}} \|X - B_k\|_{Fr.}, \ \ \|\cdot\|_{Fr.} = Frob.norm$$

where $\mathcal{L}^U \subset \mathbb{C}^{n\times n}$= algebra of matrices simultaneously diagonalized by a fast unitary transform U. One can define descent methods $\mathcal{L}QN$ [DFLZ]:

$$\mathbf{x}_0 \in R^n, \quad \mathbf{d}_0 = -\mathbf{g}_0$$
$$For \ \ k = 0, 1, \ldots$$
$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \quad \lambda_k > 0 \\ \\ \\ \\ \end{cases}$$

Local Optimization BFGS-type algorithms

Given an approssimation B_k of $\nabla^2 f(\mathbf{w}_k)$, let us define the matrix $\mathcal{L}^U_{B_k}$:

$$\|\mathcal{L}^U_{B_k} - B_k\|_{Fr.} = \min_{X \in \mathcal{L}} \|X - B_k\|_{Fr.}, \ \ \|\cdot\|_{Fr.} = Frob.norm$$

where $\mathcal{L}^U \subset \mathbb{C}^{n\times n}$= algebra of matrices simultaneously diagonalized by a fast unitary transform U. One can define descent methods $\mathcal{L}QN$ [DFLZ]:

$$\begin{array}{l} \mathbf{x}_0 \in R^n, \quad \mathbf{d}_0 = -\mathbf{g}_0 \\ For \ \ k = 0, 1, \ldots \\ \left\{ \begin{array}{l} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \quad \lambda_k > 0 \\ B_{k+1} = \varphi(\ \mathcal{L}^U_{B_k}\ ,\ \underbrace{\mathbf{x}_{k+1} - \mathbf{x}_k}_{\mathbf{s}_k},\ \underbrace{\mathbf{g}_{k+1} - \mathbf{g}_k}_{\mathbf{y}_k}), \quad \mathbf{g}_k = \nabla f(\mathbf{x}_k) \end{array} \right. \end{array}$$

Local Optimization BFGS-type algorithms

Given an approssimation B_k of $\nabla^2 f(\mathbf{w}_k)$, let us define the matrix $\mathcal{L}^U_{B_k}$:

$$\|\mathcal{L}^U_{B_k} - B_k\|_{Fr.} = \min_{X \in \mathcal{L}} \|X - B_k\|_{Fr.}, \ \ \|\cdot\|_{Fr.} = Frob.norm$$

where $\mathcal{L}^U \subset \mathbb{C}^{n\times n}$= algebra of matrices simultaneously diagonalized by a fast unitary transform U. One can define descent methods $\mathcal{L}QN$ [DFLZ]:

$$\begin{array}{l} \mathbf{x}_0 \in R^n, \quad \mathbf{d}_0 = -\mathbf{g}_0 \\ For \ \ k = 0, 1, \ldots \\ \left\{ \begin{array}{l} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \quad \lambda_k > 0 \\ B_{k+1} = \varphi(\ \mathcal{L}^U_{B_k}\ , \underbrace{\mathbf{x}_{k+1} - \mathbf{x}_k}_{\mathbf{s}_k}, \underbrace{\mathbf{g}_{k+1} - \mathbf{g}_k}_{\mathbf{y}_k}), \quad \mathbf{g}_k = \nabla f(\mathbf{x}_k) \\ \\ \mathbf{d}_{k+1} = -B^{-1}_{k+1}\mathbf{g}_{k+1} \end{array} \right. \end{array}$$

Local Optimization BFGS-type algorithms

Given an approssimation B_k of $\nabla^2 f(\mathbf{w}_k)$, let us define the matrix $\mathcal{L}^U_{B_k}$:

$$\|\mathcal{L}^U_{B_k} - B_k\|_{Fr.} = \min_{X \in \mathcal{L}} \|X - B_k\|_{Fr.}, \ \ \|\cdot\|_{Fr.} = Frob.norm$$

where $\mathcal{L}^U \subset \mathbb{C}^{n\times n}$= algebra of matrices simultaneously diagonalized by a fast unitary transform U. One can define descent methods $\mathcal{L}QN$ [DFLZ]:

$$\begin{array}{l} \mathbf{x}_0 \in R^n, \quad \mathbf{d}_0 = -\mathbf{g}_0 \\ For \ \ k = 0, 1, \ldots \\ \left\{ \begin{array}{l} \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \mathbf{d}_k \quad \lambda_k > 0 \\ B_{k+1} = \varphi(\ \mathcal{L}^U_{B_k}\ , \underbrace{\mathbf{x}_{k+1} - \mathbf{x}_k}_{\mathbf{s}_k}, \underbrace{\mathbf{g}_{k+1} - \mathbf{g}_k}_{\mathbf{y}_k}), \quad \mathbf{g}_k = \nabla f(\mathbf{x}_k) \\ \\ \mathbf{d}_{k+1} = -B^{-1}_{k+1}\mathbf{g}_{k+1} \end{array} \right. \end{array}$$

The classical BFGS method [NW] *and the more recent minimization methods introduced in* [BDFZ], [DFZ2], [DFZ3] *are examples of* $\mathcal{L}QN$ *algorithms, (being*
$\mathcal{L}^U = \mathbb{C}^{n\times n}$, $\mathcal{L}^U = \{\alpha I\}, \{Circulant - Hartley - type\}$)

The *step* λ_k is determined such that:

$$\lambda_k \mid \mathbf{s}_k^T \mathbf{y}_k > 0 \quad \& \quad f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$$

The *step* λ_k is determined such that:

$$\lambda_k \mid \mathbf{s}_k^T \mathbf{y}_k > 0 \quad \& \quad f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$$

The *updating function* φ in $B_{k+1} = \varphi\left(\mathcal{L}^U_{B_k}, \mathbf{s}_k, \mathbf{y}_k\right)$ is

$$\varphi\left(\square, \mathbf{s}, \mathbf{y}\right) = \square + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T\square\mathbf{s}}\square\mathbf{s}\mathbf{s}^T\square.$$

The *step* λ_k is determined such that:

$$\lambda_k \mid \mathbf{s}_k^T \mathbf{y}_k > 0 \quad \& \quad f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$$

The *updating function* φ in $B_{k+1} = \varphi\left(\mathcal{L}^U_{B_k}, \mathbf{s}_k, \mathbf{y}_k\right)$ is

$$\varphi\left(\square, \mathbf{s}, \mathbf{y}\right) = \square + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T\square\mathbf{s}}\square\mathbf{s}\mathbf{s}^T\square.$$

The choice of λ_k and the properties of φ and $\mathcal{L}^U_{B_k}$ imply:

- B_{k+1} *inherites positive definiteness from* B_k

The *step* λ_k is determined such that:

$$\lambda_k \mid \mathbf{s}_k^T \mathbf{y}_k > 0 \quad \& \quad f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$$

The *updating function* φ in $B_{k+1} = \varphi\left(\mathcal{L}^U_{B_k}, \mathbf{s}_k, \mathbf{y}_k\right)$ is

$$\varphi\left(\square, \mathbf{s}, \mathbf{y}\right) = \square + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T\square\mathbf{s}}\square\mathbf{s}\mathbf{s}^T\square.$$

The choice of λ_k and the properties of φ and $\mathcal{L}^U_{B_k}$ imply:

- B_{k+1} *inherites positive definiteness from* B_k
- $B_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k) = \mathbf{g}_{k+1} - \mathbf{g}_k$, $\Rightarrow$ $\mathcal{L}QN$ *secant* methods

The *step* λ_k is determined such that:

$$\lambda_k \mid \mathbf{s}_k^T \mathbf{y}_k > 0 \quad \& \quad f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$$

The *updating function* φ in $B_{k+1} = \varphi\left(\mathcal{L}^U_{B_k}, \mathbf{s}_k, \mathbf{y}_k\right)$ is

$$\varphi\left(\square, \mathbf{s}, \mathbf{y}\right) = \square + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T\square\mathbf{s}}\square\mathbf{s}\mathbf{s}^T\square.$$

The choice of λ_k and the properties of φ and $\mathcal{L}^U_{B_k}$ imply:

- B_{k+1} *inherites positive definiteness from* B_k
- $B_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k) = \mathbf{g}_{k+1} - \mathbf{g}_k$, $\Rightarrow$ $\mathcal{L}QN$ *secant* methods
- The structured space $\mathcal{L}^U \Rightarrow \mathcal{L}QN$ has low complexity

The *step* λ_k is determined such that:

$$\lambda_k \mid \mathbf{s}_k^T \mathbf{y}_k > 0 \quad \& \quad f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$$

The *updating function* φ in $B_{k+1} = \varphi\left(\mathcal{L}^U_{B_k}, \mathbf{s}_k, \mathbf{y}_k\right)$ is

$$\varphi\left(\square, \mathbf{s}, \mathbf{y}\right) = \square + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T - \frac{1}{\mathbf{s}^T\square\mathbf{s}}\square\mathbf{s}\mathbf{s}^T\square.$$

The choice of λ_k and the properties of φ and $\mathcal{L}^U_{B_k}$ imply:

- B_{k+1} *inherites positive definiteness from* B_k
- $B_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k) = \mathbf{g}_{k+1} - \mathbf{g}_k$, $\Rightarrow$ $\mathcal{L}QN$ *secant* methods
- The structured space $\mathcal{L}^U \Rightarrow \mathcal{L}QN$ has low complexity
- Every iteration of $\mathcal{L}QN$ has in our case a cost $O(nlogn)$

A Local Optimization Quasi-Newton(QN) Theorem

The following result holds (see [DFZ4], [NW]):

A Local Optimization Quasi-Newton(QN) Theorem

The following result holds (see [DFZ4], [NW]):

Theorem 1

Assume $f(\mathbf{x}) \in C^2$ and consider the unconstrained problem:

$$\min \ f(\mathbf{x}), \ \mathbf{x} \in \mathbb{C}^{\mathbf{n}}$$

A Local Optimization Quasi-Newton(QN) Theorem

The following result holds (see [DFZ4], [NW]):

Theorem 1

Assume $f(\mathbf{x}) \in C^2$ and consider the unconstrained problem:

$$\min \ f(\mathbf{x}), \ \mathbf{x} \in \mathbb{C}^{\mathbf{n}}$$

If in an iterative scheme *of BFGS-type*

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu_k {B^{(k)}}^{-1} \nabla f(\mathbf{x}^{(k)}), \ \left(B^{(k)} = \varphi(\tilde{B}^{(k-1)}, \ldots), \forall k \right)$

the following conditions are satisfied $\forall k$:

A Local Optimization Quasi-Newton(QN) Theorem

The following result holds (see [DFZ4], [NW]):

Theorem 1

Assume $f(\mathbf{x}) \in C^2$ and consider the unconstrained problem:

$$\min \ f(\mathbf{x}), \ \mathbf{x} \in \mathbb{C}^{\mathbf{n}}$$

If in an iterative scheme *of BFGS-type*

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu_k {B^{(k)}}^{-1} \nabla f(\mathbf{x}^{(k)}), \ \left(B^{(k)} = \varphi(\tilde{B}^{(k-1)}, \ldots), \forall k \right)$

the following conditions are satisfied $\forall k$:

$$cond(B^{(k)}) \leq N$$

$$\frac{\|\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)})\|^2}{(\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}))^T \lambda_k \mathbf{d}^{(k)}} = \frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k} \leq M$$

A Local Optimization Quasi-Newton(QN) Theorem

The following result holds (see [DFZ4], [NW]):

Theorem 1

Assume $f(\mathbf{x}) \in C^2$ and consider the unconstrained problem:

$$\min \ f(\mathbf{x}), \ \mathbf{x} \in \mathbb{C}^{\mathbf{n}}$$

If in an iterative scheme *of BFGS-type*

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu_k {B^{(k)}}^{-1} \nabla f(\mathbf{x}^{(k)}), \ \left(B^{(k)} = \varphi(\tilde{B}^{(k-1)}, \ldots), \forall k \right)$

the following conditions are satisfied $\forall k$:

$$cond(B^{(k)}) \leq N$$

$$\frac{\|\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)})\|^2}{(\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}))^T \lambda_k \mathbf{d}^{(k)}} = \frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k} \leq M$$

$$\Longrightarrow \quad \exists \{\mathbf{x}^{(\mathbf{k_i})}\} : \ \lim_{\mathbf{k_i} \to +\infty} \nabla f(\mathbf{x}^{(\mathbf{k_i})}) = \mathbf{0}$$

Let us consider the case :

$$\mathcal{L} = \{\mathcal{L}^I\} = \{diag(\gamma_1, ...\gamma_n),\ \gamma_i \in C\}$$

Let us consider the case :

$$\mathcal{L} = \{\mathcal{L}^I\} = \{diag(\gamma_1, ...\gamma_n),\ \gamma_i \in C\}$$

Let $\mathcal{D}_{k+1} \in \mathcal{L}^I$: $\|\mathcal{D}_{k+1} - B_{k+1}\|_{Fr.} = \min_{X \in \mathcal{L}^I} \|X - B_{k+1}\|_{Fr.}$

Let us consider the case :

$$\mathcal{L} = \{\mathcal{L}^I\} = \{diag(\gamma_1, ...\gamma_n),\ \gamma_i \in C\}$$

Let $\mathcal{D}_{k+1} \in \mathcal{L}^I : \|\mathcal{D}_{k+1} - B_{k+1}\|_{Fr.} = \min_{X \in \mathcal{L}^I} \|X - B_{k+1}\|_{Fr.}$

Let $Diag(|z|^2),\ z \in \mathbb{C}^n$, denote:

$$\begin{pmatrix} z_1^2 & 0 & \dots & 0 & 0 \\ 0 & z_2^2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \\ 0 & 0 & \dots & z_{n-1}^2 & 0 \\ 0 & 0 & \dots & 0 & z_n^2 \end{pmatrix}$$

Let us consider the case :

$$\mathcal{L} = \{\mathcal{L}^I\} = \{diag(\gamma_1, ...\gamma_n),\ \gamma_i \in C\}$$

Let $\mathcal{D}_{k+1} \in \mathcal{L}^I : \|\mathcal{D}_{k+1} - B_{k+1}\|_{Fr.} = \min_{X \in \mathcal{L}^I} \|X - B_{k+1}\|_{Fr.}$

Let $Diag(|z|^2),\ z \in \mathbb{C}^n$, denote:

$$\begin{pmatrix} z_1^2 & 0 & \dots & 0 & 0 \\ 0 & z_2^2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \\ 0 & 0 & \dots & z_{n-1}^2 & 0 \\ 0 & 0 & \dots & 0 & z_n^2 \end{pmatrix}$$

Then (see [CCDF]) every iteration has in this case a cost $O(n)$ and:

$$\mathcal{D}_{k+1} = \mathcal{D}_k + \frac{Diag(|\mathbf{y}_k|^2)}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{Diag(|\mathcal{D}_k \mathbf{s}_k|^2)}{\mathbf{s}_k^T \mathcal{D}_k \mathbf{s}_k}$$

Structured matrices in Local and Global Optimization

1. Local Optimization phase

Structured matrices in Local and Global Optimization

1. Local Optimization phase
2. Tunneling (Repelling) phase

Structured matrices in Local and Global Optimization

1. Local Optimization phase
2. Tunneling (Repelling) phase
3. α Branch and Bound (αBB) Convergence Scheme

Structured matrices in Local and Global Optimization

1. Local Optimization phase
2. Tunneling (Repelling) phase
3. α Branch and Bound (αBB) Convergence Scheme

- **Build a monotone sequence of local minima (maxima)**

Structured matrices in Local and Global Optimization

1. Local Optimization phase
2. Tunneling (Repelling) phase
3. α Branch and Bound (αBB) Convergence Scheme

- **Build a monotone sequence of local minima (maxima)**
- **Determine a set of possible global minimizers**

Structured matrices in Local and Global Optimization

1. Local Optimization phase
2. Tunneling (Repelling) phase
3. α Branch and Bound (αBB) Convergence Scheme

- **Build a monotone sequence of local minima (maxima)**
- **Determine a set of possible global minimizers**
- **Prove sufficient conditions for the convergence**

Structured matrices in Local and Global Optimization

1. Local Optimization phase
2. Tunneling (Repelling) phase
3. α Branch and Bound (αBB) Convergence Scheme

- **Build a monotone sequence of local minima (maxima)**
- **Determine a set of possible global minimizers**
- **Prove sufficient conditions for the convergence**

1. Structured approximation of the Hessian matrix

Structured matrices in Local and Global Optimization

1. Local Optimization phase
2. Tunneling (Repelling) phase
3. α Branch and Bound (αBB) Convergence Scheme

- **Build a monotone sequence of local minima (maxima)**
- **Determine a set of possible global minimizers**
- **Prove sufficient conditions for the convergence**

1. Structured approximation of the Hessian matrix
2. Low rank approximation of the Repeller matrix

Structured matrices in Local and Global Optimization

1. Local Optimization phase
2. Tunneling (Repelling) phase
3. α Branch and Bound (αBB) Convergence Scheme

- **Build a monotone sequence of local minima (maxima)**
- **Determine a set of possible global minimizers**
- **Prove sufficient conditions for the convergence**

1. Structured approximation of the Hessian matrix
2. Low rank approximation of the Repeller matrix
3. $\Longrightarrow$ Convergence is accelerated

Some preliminary results on Global Optimization

Classical "box-constrained" problems

$$\begin{cases} \min \ f(\mathbf{x}) \\ \mathbf{x^L} \leq \mathbf{x} \leq \mathbf{x^U} \end{cases}$$

Some preliminary results on Global Optimization

Classical "box-constrained" problems

$$\begin{cases} \min \ f(\mathbf{x}) \\ \mathbf{x^L} \leq \mathbf{x} \leq \mathbf{x^U} \end{cases}$$

Main features of αBB algorithm ([FLO],[FLOV])

- Tighter box constraints can be attained by partitioning the rectangle of initial box constraints into smaller rectangles by halving on the middle point of the longest side (*Bisection*)

Some preliminary results on Global Optimization

Classical "box-constrained" problems

$$\begin{cases} \min \ f(\mathbf{x}) \\ \mathbf{x^L} \leq \mathbf{x} \leq \mathbf{x^U} \end{cases}$$

Main features of αBB algorithm ([FLO],[FLOV])

- Tighter box constraints can be attained by partitioning the rectangle of initial box constraints into smaller rectangles by halving on the middle point of the longest side (*Bisection*)
- The method selects the sub-rectangle associated to the minimum value of the corresponding lower bounds (*Branch and Bound*)

Some preliminary results on Global Optimization

Classical "box-constrained" problems

$$\begin{cases} \min \ f(\mathbf{x}) \\ \mathbf{x^L} \leq \mathbf{x} \leq \mathbf{x^U} \end{cases}$$

Main features of αBB algorithm ([FLO],[FLOV])

- Tighter box constraints can be attained by partitioning the rectangle of initial box constraints into smaller rectangles by halving on the middle point of the longest side (*Bisection*)
- The method selects the sub-rectangle associated to the minimum value of the corresponding lower bounds (*Branch and Bound*)
- A *nondecreasing* sequence for the lower bounds on $f(\mathbf{x})$ and a *nonincreasing* sequence for the upper bounds on $f(\mathbf{x})$ are computed by the algorithm

Let $\mathbf{x}^L_{c(m)} \leq \mathbf{x}_{c(m)} \leq \mathbf{x}^U_{c(m)}$ denote the *current box* at iteration m. Set:

$$\alpha_{\mathbf{x}_{c(m)}} = \max\left\{0, -\frac{1}{2}\min \lambda_{(\nabla^2 f(\mathbf{x}_{c(m)})}\right\}$$

Let $\mathbf{x}^L_{c(m)} \leq \mathbf{x}_{c(m)} \leq \mathbf{x}^U_{c(m)}$ denote the *current box* at iteration m. Set:

$$\alpha_{\mathbf{x}_{c(m)}} = \max\left\{0, -\frac{1}{2}\min \lambda_{(\nabla^2 f(\mathbf{x}_{c(m)})}\right\}$$

$$L_{c(m)}(\mathbf{x}_{c(m)}) = f(\mathbf{x}_{c(m)}) + \alpha_{\mathbf{x}_{c(m)}}(\mathbf{x}^L_{c(m)} - \mathbf{x}_{c(m)})(\mathbf{x}^U_{c(m)} - \mathbf{x}_{c(m)})$$

Let $\mathbf{x}^L_{c(m)} \leq \mathbf{x}_{c(m)} \leq \mathbf{x}^U_{c(m)}$ denote the *current box* at iteration m.
Set:

$$\alpha_{\mathbf{x}_{c(m)}} = \max\left\{0, -\frac{1}{2}\min \lambda_{(\nabla^2 f(\mathbf{x}_{c(m)})}\right\}$$

$$L_{c(m)}(\mathbf{x}_{c(m)}) = f(\mathbf{x}_{c(m)}) + \alpha_{\mathbf{x}_{c(m)}}(\mathbf{x}^L_{c(m)} - \mathbf{x}_{c(m)})(\mathbf{x}^U_{c(m)} - \mathbf{x}_{c(m)})$$

Hence:

$$L_{c(m)}(\mathbf{x}_{c(m)}) \leq f(\mathbf{x}_{c(m)}), \quad \forall\ \mathbf{x}_{c(m)}$$

Let $\mathbf{x}^L_{c(m)} \leq \mathbf{x}_{c(m)} \leq \mathbf{x}^U_{c(m)}$ denote the *current box* at iteration m.
Set:

$$\alpha_{\mathbf{x}_{c(m)}} = \max\left\{0, -\frac{1}{2}\min \lambda_{(\nabla^2 f(\mathbf{x}_{c(m)})}\right\}$$

$$L_{c(m)}(\mathbf{x}_{c(m)}) = f(\mathbf{x}_{c(m)}) + \alpha_{\mathbf{x}_{c(m)}}(\mathbf{x}^L_{c(m)} - \mathbf{x}_{c(m)})(\mathbf{x}^U_{c(m)} - \mathbf{x}_{c(m)})$$

Hence:

$$L_{c(m)}(\mathbf{x}_{c(m)}) \leq f(\mathbf{x}_{c(m)}), \quad \forall\ \mathbf{x}_{c(m)}$$

$$\inf_{\mathbf{x}_{c(m)}} L_{c(m)}(\mathbf{x}_{c(m)}) \leq \inf_{\mathbf{x}_{c(m)}} f(\mathbf{x}_{c(m)})$$

$$f\left((\mathbf{x}^L_{c(m)} + \mathbf{x}^U_{c(m)}/2\right) \geq \inf_{\mathbf{x}_{c(m)}} f(\mathbf{x}_{c(m)})$$

The following global convergence theorem holds ([FLO], [FLOV]):

Theorem 2

Consider the box-constrained problem. Assume $f(\mathbf{x}) \in C^2 \quad \Longrightarrow$
$\|\nabla^2 f(\mathbf{x})^{-1}\| \leq c, \quad \forall m \quad \exists\ \alpha^*_m = \max_{\mathbf{x}_{c(m)}} \ \alpha_{\mathbf{x}_{c(m)}}$
Set:

$$f^L_{c(m)} = \inf_{\mathbf{x}_{c(m)}} \ L_{c(m)}(\mathbf{x}_{c(m)})$$

$$f^U_{c(m)} = f\left((\mathbf{x}^L_{c(m)} + \mathbf{x}^U_{c(m)})/2\right)$$

then, it follows $\forall m$:

$$f^L_{c(m)} \leq f^L_{c(m+1)} \leq \min_{\mathbf{x}_{c(m+1)}} f(\mathbf{x}_{c(m+1)}) \equiv \min_{\mathbf{x}} f(\mathbf{x})$$

$$f^U_{c(m)} \geq f^U_{c(m+1)} \geq \min_{\mathbf{x}} f(\mathbf{x}) \geq f^L_{c(m)}$$

Moreover, $\forall \epsilon_a > 0, \quad \exists m^* : \quad \forall m \geq m^*$:

$$\begin{cases} f^U_{c(m)} - f^L_{c(m)} < \epsilon_a \\ \|\mathbf{x}^U_{c(m)} - \mathbf{x}^L_{c(m)}\|_2 \leq \sqrt{4\epsilon_a / c} \end{cases}$$

A Global Optimization Quasi-Newton(QN) Theorem

By combining Theorem 1 and Theorem 2, one can prove (see [F]):

A Global Optimization Quasi-Newton(QN) Theorem

By combining Theorem 1 and Theorem 2, one can prove (see [F]):

Theorem 3

Assume $f(\mathbf{x}) \in C^2$ and consider the box-constrained problem:

$$\begin{cases} \min \ f(\mathbf{x}) \\ \mathbf{x^L} \leq \mathbf{x} \leq \mathbf{x^U} \end{cases}$$

A Global Optimization Quasi-Newton(QN) Theorem

By combining Theorem 1 and Theorem 2, one can prove (see [F]):

Theorem 3

Assume $f(\mathbf{x}) \in C^2$ and consider the box-constrained problem:

$$\begin{cases} \min \ f(\mathbf{x}) \\ \mathbf{x^L} \leq \mathbf{x} \leq \mathbf{x^U} \end{cases}$$

If in an iterative scheme *of BFGS-type*

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu_k \mathcal{B}^{(k)^{-1}} \nabla f(\mathbf{x}^{(k)})$, $\left(\mathcal{B}^{(k)} = \varphi(\tilde{B}^{(k-1)}, \ldots), \forall k\right)$

$\mathbf{x^L} \leq \ \mathbf{x}^{(k)} \leq \mathbf{x^U}$, the following conditions are satisfied $\forall k$:

A Global Optimization Quasi-Newton(QN) Theorem

By combining Theorem 1 and Theorem 2, one can prove (see [F]):

Theorem 3

Assume $f(\mathbf{x}) \in C^2$ and consider the box-constrained problem:

$$\begin{cases} \min \ f(\mathbf{x}) \\ \mathbf{x^L} \leq \mathbf{x} \leq \mathbf{x^U} \end{cases}$$

If in an iterative scheme *of BFGS-type*

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu_k \mathcal{B}^{(k)^{-1}} \nabla f(\mathbf{x}^{(k)})$, $\left(\mathcal{B}^{(k)} = \varphi(\tilde{B}^{(k-1)}, \ldots), \forall k\right)$

$\mathbf{x^L} \leq \ \mathbf{x}^{(k)} \leq \mathbf{x^U}$, the following conditions are satisfied $\forall k$:

$$cond(\mathcal{B}^{(k)}) \leq N$$

$$\frac{\|\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)})\|^2}{(\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}))^T \lambda_k \mathbf{d}^{(k)}} = \frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k} \leq M$$

A Global Optimization Quasi-Newton(QN) Theorem

By combining Theorem 1 and Theorem 2, one can prove (see [F]):

Theorem 3

Assume $f(\mathbf{x}) \in C^2$ and consider the box-constrained problem:

$$\begin{cases} \min \ f(\mathbf{x}) \\ \mathbf{x^L} \leq \mathbf{x} \leq \mathbf{x^U} \end{cases}$$

If in an iterative scheme *of BFGS-type*

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mu_k \mathcal{B}^{(k)^{-1}} \nabla f(\mathbf{x}^{(k)})$, $\left(\mathcal{B}^{(k)} = \varphi(\tilde{B}^{(k-1)}, \ldots), \forall k\right)$

$\mathbf{x^L} \leq \ \mathbf{x}^{(k)} \leq \mathbf{x^U}$, the following conditions are satisfied $\forall k$:

$$cond(\mathcal{B}^{(k)}) \leq N$$

$$\frac{\|\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)})\|^2}{(\nabla f(\mathbf{x}^{(k+1)}) - \nabla f(\mathbf{x}^{(k)}))^T \lambda_k \mathbf{d}^{(k)}} = \frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T \mathbf{s}_k} \leq M$$

then the algorithm is convergent to the global minimum of $f(\mathbf{x})$

Matrix Structures in a Global Minimization scheme

Let $\mathbf{x}^{(\tilde{k})}$ be an approximation of a local minimizer for $f(\mathbf{x}) \in C^1$.

Matrix Structures in a Global Minimization scheme

Let $\mathbf{x}^{(\tilde{k})}$ be an approximation of a local minimizer for $f(\mathbf{x}) \in C^1$.
A matrix $\mathcal{A}^{(\tilde{k})}$ is called a *repeller matrix* for $\mathbf{x}^{(\tilde{k})}$ if $\exists \hat{\mathbf{x}}$:

$$\begin{cases} \hat{\mathbf{x}} = \mathbf{x}^{(\tilde{k})} - \mathcal{A}^{(\tilde{k})} \nabla f(\mathbf{x}^{(\tilde{k})}) \\ f(\hat{\mathbf{x}}) < f(\mathbf{x}^{(\tilde{k})}) \end{cases}$$

Matrix Structures in a Global Minimization scheme

Let $\mathbf{x}^{(\tilde{k})}$ be an approximation of a local minimizer for $f(\mathbf{x}) \in C^1$.
A matrix $\mathcal{A}^{(\tilde{k})}$ is called a *repeller matrix* for $\mathbf{x}^{(\tilde{k})}$ if $\exists \hat{\mathbf{x}}$:

$$\begin{cases} \hat{\mathbf{x}} = \mathbf{x}^{(\tilde{k})} - \mathcal{A}^{(\tilde{k})} \nabla f(\mathbf{x}^{(\tilde{k})}) \\ f(\hat{\mathbf{x}}) < f(\mathbf{x}^{(\tilde{k})}) \end{cases}$$

The repeller matrix $\mathcal{A}^{(\tilde{k}_m)}_{c(m)}$ in every box $c(m)$ and for any given computed local minimizer $\mathbf{x}^{(\tilde{k}_m)}_{c(m)}$ can be approximated in the following way:

Matrix Structures in a Global Minimization scheme

Let $\mathbf{x}^{(\tilde{k})}$ be an approximation of a local minimizer for $f(\mathbf{x}) \in C^1$.
A matrix $\mathcal{A}^{(\tilde{k})}$ is called a *repeller matrix* for $\mathbf{x}^{(\tilde{k})}$ if $\exists \hat{\mathbf{x}}$:

$$\begin{cases} \hat{\mathbf{x}} = \mathbf{x}^{(\tilde{k})} - \mathcal{A}^{(\tilde{k})} \nabla f(\mathbf{x}^{(\tilde{k})}) \\ f(\hat{\mathbf{x}}) < f(\mathbf{x}^{(\tilde{k})}) \end{cases}$$

The repeller matrix $\mathcal{A}^{(\tilde{k}_m)}_{c(m)}$ in every box $c(m)$ and for any given computed local minimizer $\mathbf{x}^{(\tilde{k}_m)}_{c(m)}$ can be approximated in the following way:

$$\mathcal{A}^{(\tilde{k}_m)}_{c(m)} \approx \lambda^{(\tilde{k}_m)}_{c(m)} I + (I/\mu + R_{c(m)})^{-1}, \quad 2 \leq rank(R_{c(m)}) \leq 4$$

Matrix Structures in a Global Minimization scheme

Let $\mathbf{x}^{(\tilde{k})}$ be an approximation of a local minimizer for $f(\mathbf{x}) \in C^1$.
A matrix $\mathcal{A}^{(\tilde{k})}$ is called a *repeller matrix* for $\mathbf{x}^{(\tilde{k})}$ if $\exists \hat{\mathbf{x}}$:

$$\begin{cases} \hat{\mathbf{x}} = \mathbf{x}^{(\tilde{k})} - \mathcal{A}^{(\tilde{k})} \nabla f(\mathbf{x}^{(\tilde{k})}) \\ f(\hat{\mathbf{x}}) < f(\mathbf{x}^{(\tilde{k})}) \end{cases}$$

The repeller matrix $\mathcal{A}_{c(m)}^{(\tilde{k}_m)}$ in every box $c(m)$ and for any given computed local minimizer $\mathbf{x}_{c(m)}^{(\tilde{k}_m)}$ can be approximated in the following way:

$$\mathcal{A}_{c(m)}^{(\tilde{k}_m)} \approx \lambda_{c(m)}^{(\tilde{k}_m)} I + (I/\mu + R_{c(m)})^{-1}, \quad 2 \leq rank(R_{c(m)}) \leq 4$$

being, by terminal attractors theory, $\lambda_{c(m)}^{(\tilde{k}_m)}$ the *maximal scalar repeller* i.e.:

$$\lambda_{c(m)}^{(\tilde{k}_m)} = \frac{\epsilon_a}{\|\nabla f(\mathbf{x}_{c(m)}^{(\tilde{k}_m)})\|^2}, \quad \|\nabla f(\mathbf{x}_{c(m)}^{(\tilde{k}_m)})\| << \sqrt{\epsilon_a}, \quad \epsilon_a \text{ desired precision}$$

Matrix Structures in a Global Minimization scheme

Let $\mathbf{x}^{(\tilde{k})}$ be an approximation of a local minimizer for $f(\mathbf{x}) \in C^1$.
A matrix $\mathcal{A}^{(\tilde{k})}$ is called a *repeller matrix* for $\mathbf{x}^{(\tilde{k})}$ if $\exists \hat{\mathbf{x}}$:

$$\begin{cases} \hat{\mathbf{x}} = \mathbf{x}^{(\tilde{k})} - \mathcal{A}^{(\tilde{k})} \nabla f(\mathbf{x}^{(\tilde{k})}) \\ f(\hat{\mathbf{x}}) < f(\mathbf{x}^{(\tilde{k})}) \end{cases}$$

The repeller matrix $\mathcal{A}_{c(m)}^{(\tilde{k}_m)}$ in every box $c(m)$ and for any given computed local minimizer $\mathbf{x}_{c(m)}^{(\tilde{k}_m)}$ can be approximated in the following way:

$$\mathcal{A}_{c(m)}^{(\tilde{k}_m)} \approx \lambda_{c(m)}^{(\tilde{k}_m)} I + (I/\mu + R_{c(m)})^{-1}, \quad 2 \leq rank(R_{c(m)}) \leq 4$$

being, by terminal attractors theory, $\lambda_{c(m)}^{(\tilde{k}_m)}$ the *maximal scalar repeller* i.e.:

$$\lambda_{c(m)}^{(\tilde{k}_m)} = \frac{\epsilon_a}{\|\nabla f(\mathbf{x}_{c(m)}^{(\tilde{k}_m)})\|^2}, \quad \|\nabla f(\mathbf{x}_{c(m)}^{(\tilde{k}_m)})\| << \sqrt{\epsilon_a}, \quad \epsilon_a \text{ desired precision}$$

$R_{c(m)}$ with the following structure:

$$\begin{cases} R = \mu_1 \mathbf{p}\mathbf{p}^T + \mu_2 \mathbf{q}\mathbf{q}^T + \mu_3 \mathbf{p}\mathbf{r}^T + \mu_4 \mathbf{r}\mathbf{q}^T \\ \mathbf{p},\ \mathbf{q},\ \mathbf{r} \text{ suitable vectors} \quad \mu_1,\ \mu_2,\ \mu_3,\ \mu_4 \text{ scalars,} \end{cases}$$

The main steps of each optimization cycle of the Algorithm are:

1. Compute a local minimum $\mathbf{x}_{c(m)}^{(\tilde{k}_m)}$ in the box $c(m)$
2. Apply a scalar repeller $\lambda_{c(m)}^{(\tilde{k}_m)}$ and compute $\mathbf{x}_{c(m)}^{(\tilde{k}_{m+1})}$
3. Approximate $\mathcal{A}_{c(m)}^{(\tilde{k}_m)}$ with a $R_{c(m)}$ correction, $rank(R_{c(m)}) = 2$
4. Compute $\mathbf{x}_{c(m)}^{(\tilde{k}_{m+2})}$
5. If $f(\mathbf{x}_{c(m)}^{(\tilde{k}_{m+2})}) < f(\mathbf{x}_{c(m)}^{(\tilde{k}_m)})$ set $\mathbf{x}_{c(m)}^{(0)} = \mathbf{x}_{c(m)}^{(\tilde{k}_{m+2})}$ and start a new local search in $c(m)$
6. Else: approximate $\mathcal{A}_{c(m)}^{(\tilde{k}_m)}$ with a $R_{c(m)}$ correction, $\text{rank}(R_{c(m)}) = 3, 4$
7. Repeat 4. and 5.
8. Else: define a new box $c(m+1)$

Computational advantages of the Algorithm

- Every application of Shermann-Morrison-Woodbury formula in the tunneling phase has in our case a cost $O(n)$

Computational advantages of the Algorithm

- Every application of Shermann-Morrison-Woodbury formula in the tunneling phase has in our case a cost $O(n)$
- The one-dimensional optimal search of μ_0 can be efficiently performed by applying Armijo-Goldstein method

Computational advantages of the Algorithm

- Every application of Shermann-Morrison-Woodbury formula in the tunneling phase has in our case a cost $O(n)$
- The one-dimensional optimal search of μ_0 can be efficiently performed by applying Armijo-Goldstein method
- A satisfactory application of the algorithm depends on:

$$\begin{cases} \longrightarrow \text{the structure of eigenvalues of } \mathcal{A}_{c(m)}^{(\tilde{k}_m)} \\ \longrightarrow \text{the condition number of } \lambda_{c(m)}^{(\tilde{k}_m)} I + (I/\mu_0 + R_{c(m)}(\mu_0))^{-1} \end{cases}$$

Computational advantages of the Algorithm

- Every application of Shermann-Morrison-Woodbury formula in the tunneling phase has in our case a cost $O(n)$
- The one-dimensional optimal search of μ_0 can be efficiently performed by applying Armijo-Goldstein method
- A satisfactory application of the algorithm depends on:

$$\begin{cases} \longrightarrow \text{the structure of eigenvalues of } \mathcal{A}_{c(m)}^{(\tilde{k}_m)} \\ \longrightarrow \text{the condition number of } \lambda_{c(m)}^{(\tilde{k}_m)} I + (I/\mu_0 + R_{c(m)}(\mu_0))^{-1} \end{cases}$$

- The number of box-iterations and/or the operations performed in each iteration is in general considerably reduced with respect to the classical αBB procedure

References

[CBB] B.C. Cetin, J. Barhen, J.W. Burdick, Terminal repeller unconstrained subenergy tunneling for fast global optimization, *J. Optim. Theory Appl.*, 77, pp. 97-126, 1993.

[DFZ0] C.Di Fiore, S.Fanelli, P.Zellini, Optimisation strategies for nonconvex functions and applications to neural networks, *ICONIP 2001*, Shanghai, 1, pp.453–458, 2001.

[DFZ1] C.Di Fiore, S.Fanelli, P.Zellini, Computational experiences of a novel algorithm for optimal learning in MLP-networks, *ICONIP 2002*, Singapore, 1, pp.317–321, 2002.

[DFLZ] C.Di Fiore, S. Fanelli, F. Lepore, P. Zellini, Matrix algebras in Quasi-Newton methods for unconstrained optimization, *Numerische Mathematik*, 94, pp. 479–500, 2003.

[BDFZ] 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, pp. 263–273, 2003.

[CCDF] J.F. Chai, R.H. Chan, C. Di Fiore, Minimization of a detailed-preserving regularization functional for impulse noise removal, *Journal of Math. Imaging Vis.*, 29, pp. 79-91, (2007).

[DFZ2] C.Di Fiore, S.Fanelli, P.Zellini, An efficient generalization of Battiti-Shanno's Quasi-Newton Algorithm for learning in MLP-networks, *ICONIP'04*, Calcutta, pp.483–488, 2004.

[DFZ3] C. Di Fiore, S. Fanelli, P. Zellini, Low complexity minimization algorithms, *Numerical Linear Algebra with Applications*, 12, pp.755–768, 2005.

[DFZ4] C. Di Fiore, S. Fanelli, P. Zellini, Low complexity secant quasi-Newton minimization algorithms for non convex functions, *Journal of Computational and Applied Mathematics*, 210, pp.167–174, 2007.

[F] S. Fanelli, A BFGS-type algorithm for box-constrained global optimization, *Journal of Optim. Theory and Applications*, 149, 1, pp.175–196, 2011.

[FLO] C. A. Floudas, *Deterministic global optimization*, Kluwer, 2000.

[FLOV] C. A. Floudas, V. Visweswaran, A primal relaxed dual global optimization approach, *J. Optim. Theory Appl.*, 78(2), pp.187-225, 1993.

[MMA] A. Musrrat, P. Millie, A. Ajith, Simplex differential evolution, *Acta Polytechnica Hungarica*, 6, pp.95-115, 2009.

[NW] J. Nocedal, S.J. Wright, *Numerical Optimization*,Springer-Verlag, 1999.

[OT] I. Oseledets, E. Tyrtyshnikov, A unifying approach to the construction of circulant preconditioners, *Linear Algebra and its Appl.*, 418, pp. 435-449, 2006.

[P] M.J.D.Powell, Some global convergence properties of a variable metric algorithm for minimization without exact line search, *Nonlinear Programming, SIAM- AMS Proc.*, 9, pp. 53–72, 1976.

[T] E.Tyrtyshnikov, private communication

[YLT] K.F.C. Yiu, Y. Liu, K.L. Teo, A hybrid descent method for global optimization, *J. Glob. Optim.*, 28, pp. 229-238, 2004.