Spectrum-based limited memory algorithms

Oleg Burdakov

Linköping University, Sweden

Joint work with:
Johannes Brust, Jennifer B. Erway, Lujin Gong, Serge Gratton, Roummel F. Marcia, Ya-xiang Yuan, Liang Zhao, Spartak Zikrin

Overview

1. Combination of limited memory and trust region techniques
2. Combination of limited memory technique and cubic regularization
3. Search over a model-based steepest descent path
4. Dense initialization for limited memory algorithms
5. Future plans

Part 1.
Combination of limited memory and trust region techniques

Oleg Burdakov, Lujin Gong, Spartak Zikrin and Ya-xiang Yuan.
On efficiently combining limited-memory and trust-region techniques.
Math. Prog. Comp. (2017) **9**, pp. 101-134.

Unconstrained minimization

$$\min_{x \in R^n} f(x)$$

Trust region framework

$$x_{k+1} = x_k + s_k$$

Trust region subproblem

$$\min_{s \in R^n:\ \|s\| \le \Delta_k} q_k(s) = g_k^T s + \frac{1}{2} s^T B_k s$$

Trial point $x_k + s^*$
If successful, $x_{k+1} = x_k + s^*$

Unconstrained minimization

$$\min_{x \in R^n} f(x)$$

Trust region framework

$$x_{k+1} = x_k + s_k$$

Trust region subproblem

$$\min_{s \in R^n:\ \|s\| \le \Delta_k} q_k(s) = g_k^T s + \frac{1}{2} s^T B_k s$$

Trial point $x_k + s^*$
If successful, $x_{k+1} = x_k + s^*$

Hessian approximation

Quasi-Newton approximation

$$B_{k+1} = B_k + \Delta B(B_k, s_k, y_k, \ldots)$$

$s_k = x_{k+1} - x_k,$
$y_k = \nabla f(_{k+1}) - \nabla f(_k) = g_{k+1} - g_k$

Limited memory approximation

$$B_{k+1} = \gamma_k I + \underbrace{\begin{bmatrix} V_k \end{bmatrix} \begin{bmatrix} W_k \end{bmatrix} \begin{bmatrix} V_k^T \end{bmatrix}}_{n} \Big\} m \qquad m \ll n$$

Hessian approximation

Quasi-Newton approximation

$$B_{k+1} = B_k + \Delta B(B_k, s_k, y_k, \ldots)$$

$s_k = x_{k+1} - x_k,$
$y_k = \nabla f(_{k+1}) - \nabla f(_k) = g_{k+1} - g_k$

Limited memory approximation

$$B_{k+1} = \gamma_k I + \boxed{V_k} \boxed{W_k} \underbrace{\boxed{V_k^T}\Big\} m}_{n} \qquad m \ll n$$

Limited memory approximation in TR framework

Straightforward implementation of limited memory approximation in TR framework deteriorates its efficiency.

Aim:

to develop a norm which would simplify the solution of the TR subproblem and which would retain the low-cost property of limited memory iterations.

Limited memory approximation in TR framework

Straightforward implementation of limited memory approximation in TR framework deteriorates its efficiency.

Aim:
to develop a norm which would simplify the solution of the TR subproblem and which would retain the low-cost property of limited memory iterations.

New basis

$$\underbrace{V = QR}_{QR\text{-decomposition}} \quad \Longrightarrow \quad B = \gamma I + VWV^T = \gamma I + Q(RWR^T)Q^T$$

$$\underbrace{RWR^T = UDU^T}_{\text{eigenvalue decomposition}} \quad \Longrightarrow \quad B = \gamma I + P_{\|} D P_{\|}^T,$$

where $P_{\|} = QU \in R^{n \times m}$ is orthonormal ($P_{\|}^T P_{\|} = I$).

$P = [P_{\|}, P_{\perp}] \in R^{n \times n}$ is an orthonormal basis in R^n.

New basis

$$\underbrace{V = QR}_{QR\text{-decomposition}} \quad \Longrightarrow \quad B = \gamma I + VWV^T = \gamma I + Q(RWR^T)Q^T$$

$$\underbrace{RWR^T = UDU^T}_{\text{eigenvalue decomposition}} \quad \Longrightarrow \quad B = \gamma I + P_{\|} D P_{\|}^T,$$

where $P_{\|} = QU \in R^{n \times m}$ is orthonormal ($P_{\|}^T P_{\|} = I$).

$P = [P_{\|}, P_{\perp}] \in R^{n \times n}$ is an orthonormal basis in R^n.

New basis

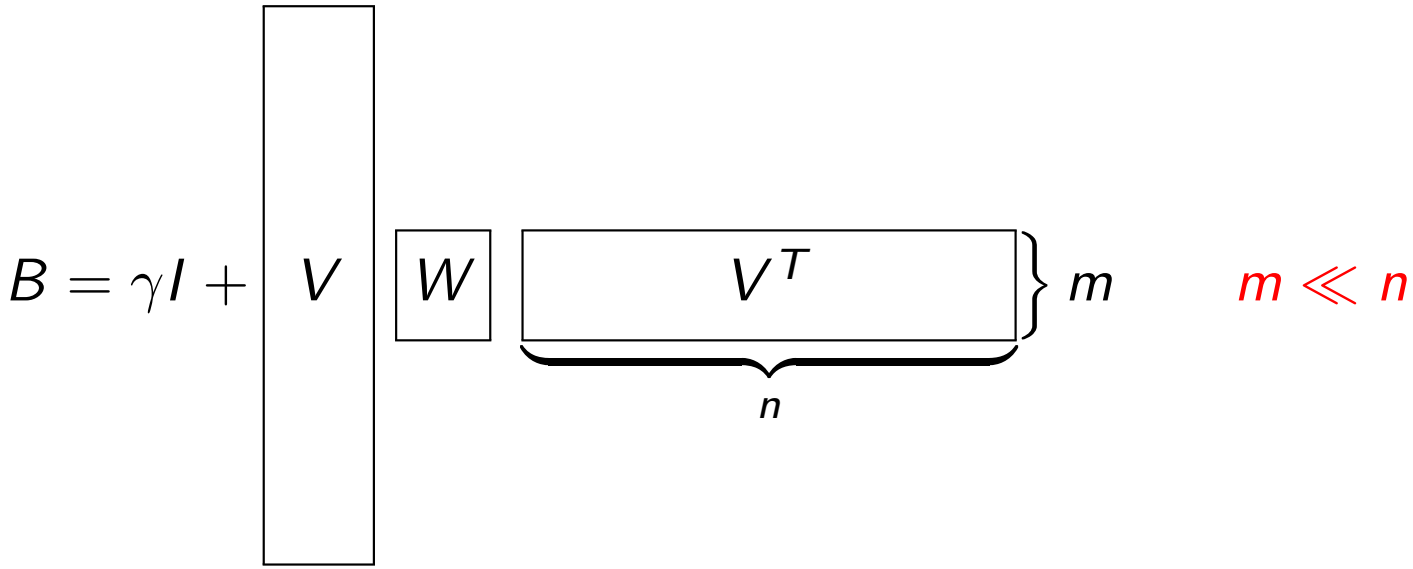
$$\underbrace{V = QR}_{QR\text{-decomposition}} \quad \Longrightarrow \quad B = \gamma I + VWV^T = \gamma I + Q(RWR^T)Q^T$$

$$\underbrace{RWR^T = UDU^T}_{\text{eigenvalue decomposition}} \quad \Longrightarrow \quad B = \gamma I + P_{\|} D P_{\|}^T,$$

where $P_{\|} = QU \in R^{n \times m}$ is orthonormal ($P_{\|}^T P_{\|} = I$).

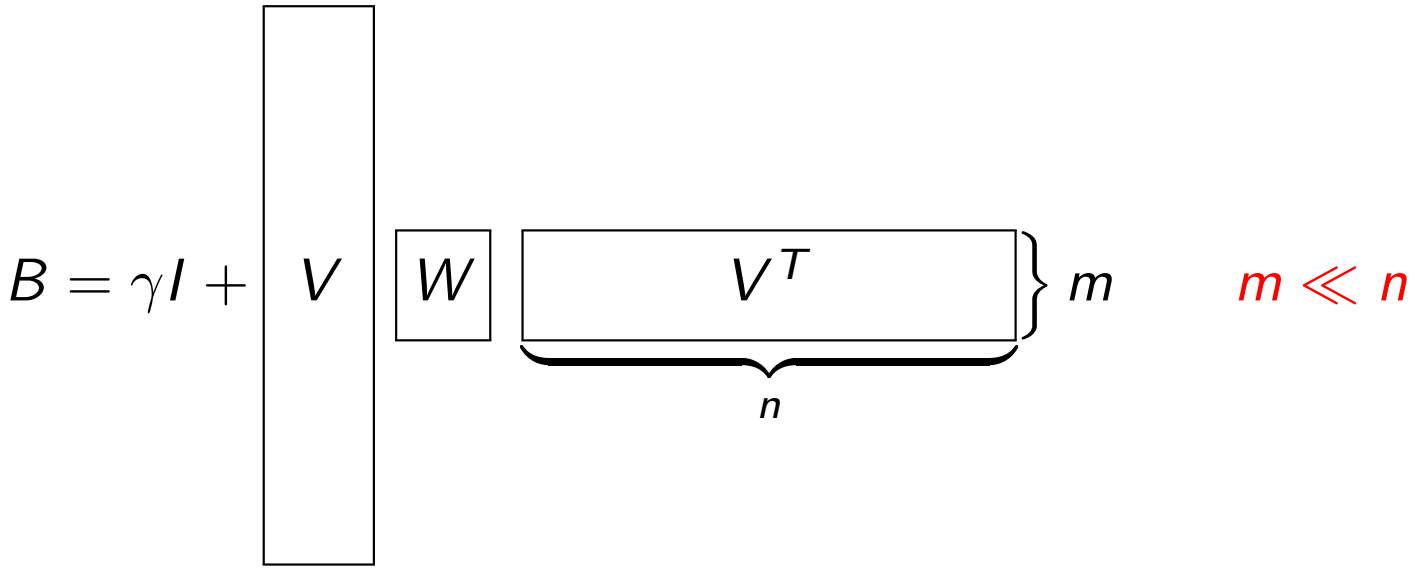
$P = [P_{\|}, P_{\perp}] \in R^{n \times n}$ is an orthonormal basis in R^n.

Eigendecomposition



- $Bu = \gamma u,\ \forall u \in P_\perp \quad \Longrightarrow \quad n - m$ eigenvalues γ with eigenspace $P_\perp$
- $Bu_i = (\gamma + d_i)u_i,\ \forall u_i$, the i-th column of $P_\parallel \quad \Longrightarrow \quad m$ eigenvalues $\lambda_i = \gamma + d_i$, where d_i is the i-th eigenvalue of RWR^T

Eigendecomposition



- $Bu = \gamma u,\ \forall u \in P_{\perp} \quad \Longrightarrow \quad n - m$ eigenvalues γ with eigenspace $P_{\perp}$
- $Bu_i = (\gamma + d_i)u_i,\ \forall u_i$, the i-th column of $P_{\parallel} \quad \Longrightarrow \quad m$ eigenvalues $\lambda_i = \gamma + d_i$, where d_i is the i-th eigenvalue of RWR^T

Model function in the new basis

$$s = P_{\|} v_{\|} + P_{\perp} v_{\perp}$$
$$g = P_{\|} g_{\|} + P_{\perp} g_{\perp}$$

New variables

$$v = (v_{\|}, v_{\perp}) = P^T s$$

Decomposition

$$q_P(v) = q(P_{\|} v_{\|} + P_{\perp} v_{\perp}) = q_{\|}(v_{\|}) + q_{\perp}(v_{\perp})$$

where

$$q_{\|}(v_{\|}) = g_{\|}^T v_{\|} + v_{\|}^T \Lambda v_{\|}/2 = \sum_{i=1}^{m} [(g_{\|})_i (v_{\|})_i + \lambda_i (v_{\|})_i^2/2]$$
$$q_{\perp}(v_{\perp}) = g_{\perp}^T v_{\perp} + \gamma \|v_{\perp}\|_2^2/2$$

Model function in the new basis

$$s = P_{\|} v_{\|} + P_{\perp} v_{\perp}$$
$$g = P_{\|} g_{\|} + P_{\perp} g_{\perp}$$

New variables

$$v = (v_{\|}, v_{\perp}) = P^T s$$

Decomposition

$$q_P(v) = q(P_{\|} v_{\|} + P_{\perp} v_{\perp}) = q_{\|}(v_{\|}) + q_{\perp}(v_{\perp})$$

where

$$q_{\|}(v_{\|}) = g_{\|}^T v_{\|} + v_{\|}^T \Lambda v_{\|}/2 = \sum_{i=1}^{m} [(g_{\|})_i (v_{\|})_i + \lambda_i (v_{\|})_i^2/2]$$
$$q_{\perp}(v_{\perp}) = g_{\perp}^T v_{\perp} + \gamma \|v_{\perp}\|_2^2/2$$

Model function in the new basis

$$s = P_{\|} v_{\|} + P_{\perp} v_{\perp}$$
$$g = P_{\|} g_{\|} + P_{\perp} g_{\perp}$$

New variables

$$v = (v_{\|}, v_{\perp}) = P^T s$$

Decomposition

$$q_P(v) = q(P_{\|} v_{\|} + P_{\perp} v_{\perp}) = q_{\|}(v_{\|}) + q_{\perp}(v_{\perp})$$

where

$$q_{\|}(v_{\|}) = g_{\|}^T v_{\|} + v_{\|}^T \Lambda v_{\|}/2 = \sum_{i=1}^{m} [(g_{\|})_i (v_{\|})_i + \lambda_i (v_{\|})_i^2/2]$$
$$q_{\perp}(v_{\perp}) = g_{\perp}^T v_{\perp} + \gamma \|v_{\perp}\|_2^2/2$$

New vector norm

Norm definition

$$\|s\|_{P,\infty} = \max\{\|P_{\|}^T s\|_\infty, \|P_{\perp}^T s\|_2\} = \max\{\|v_{\|}\|_\infty, \|v_{\perp}\|_2\}$$

Norm equivalence

$$\frac{\|s\|_2}{\sqrt{m+1}} \leq \|s\|_{P,\infty} \leq \|s\|_2$$

The norm doesn't depend on n or k

Trust region

$$\|s\|_{P,\infty} \leq \Delta \iff \begin{cases} |(v_{\|})_i| \leq \Delta, & i = 1, \ldots, m \\ \|v_{\perp}\|_2 \leq \Delta \end{cases}$$

New vector norm

Norm definition

$$\|s\|_{P,\infty} = \max\{\|P_{\|}^T s\|_\infty, \|P_{\perp}^T s\|_2\} = \max\{\|v_{\|}\|_\infty, \|v_{\perp}\|_2\}$$

Norm equivalence

$$\frac{\|s\|_2}{\sqrt{m+1}} \leq \|s\|_{P,\infty} \leq \|s\|_2$$

The norm doesn't depend on n or k

Trust region

$$\|s\|_{P,\infty} \leq \Delta \iff \begin{cases} |(v_{\|})_i| \leq \Delta, & i = 1, \ldots, m \\ \|v_{\perp}\|_2 \leq \Delta \end{cases}$$

New vector norm

Norm definition

$$\|s\|_{P,\infty} = \max\{\|P_{\|}^T s\|_\infty, \|P_{\perp}^T s\|_2\} = \max\{\|v_{\|}\|_\infty, \|v_{\perp}\|_2\}$$

Norm equivalence

$$\frac{\|s\|_2}{\sqrt{m+1}} \leq \|s\|_{P,\infty} \leq \|s\|_2$$

The norm doesn't depend on n or k

Trust region

$$\|s\|_{P,\infty} \leq \Delta \iff \begin{cases} |(v_{\|})_i| \leq \Delta, & i = 1, \ldots, m \\ \|v_{\perp}\|_2 \leq \Delta \end{cases}$$

TR subproblem decomposition

$$\min_{\|s\|_{P,\infty} \le \Delta} q(s) = \min_{\|v_{\|}\|_\infty \le \Delta} q_{\|}(v_{\|}) + \min_{\|v_\perp\|_2 \le \Delta} q_\perp(v_\perp)$$

$$= \sum_{i=1}^{m} \min_{|(v_{\|})_i| \le \Delta} [(g_{\|})_i (v_{\|})_i + \lambda_i (v_{\|})_i^2/2]$$

$$+ \min_{\|v_\perp\|_2 \le \Delta} [g_\perp^T v_\perp + \gamma \|v_\perp\|_2^2/2]$$

$$\Downarrow$$

$$(v_{\|}^*)_i = \begin{cases} -(g_{\|})_i/\lambda_i, & \text{if } |(g_{\|})_i| < \lambda_i \Delta \\ -\Delta, & \text{if } (g_{\|})_i = 0 \text{ and } \lambda_i < 0 \\ -\text{sign}((g_{\|})_i)\Delta, & \text{otherwise} \end{cases} \qquad i = 1, \ldots, m$$

$$v_\perp^* = -\min\{1/\gamma, \Delta/\|g_\perp\|\} g_\perp = -\alpha g_\perp$$

TR subproblem decomposition

$$\min_{\|s\|_{P,\infty}\le\Delta} q(s) = \min_{\|v_{\|}\|_\infty\le\Delta} q_{\|}(v_{\|}) + \min_{\|v_\perp\|_2\le\Delta} q_\perp(v_\perp)$$

$$= \sum_{i=1}^{m} \min_{|(v_{\|})_i|\le\Delta} [(g_{\|})_i (v_{\|})_i + \lambda_i (v_{\|})_i^2/2]$$

$$+ \min_{\|v_\perp\|_2\le\Delta} [g_\perp^T v_\perp + \gamma\|v_\perp\|_2^2/2]$$

$$\Downarrow$$

$$(v_{\|}^*)_i = \begin{cases} -(g_{\|})_i/\lambda_i, & \text{if } |(g_{\|})_i| < \lambda_i\Delta \\ -\Delta, & \text{if } (g_{\|})_i = 0 \text{ and } \lambda_i < 0 \\ -\text{sign}((g_{\|})_i)\Delta, & \text{otherwise} \end{cases} \qquad i = 1,\dots,m$$

$$v_\perp^* = -\min\{1/\gamma, \Delta/\|g_\perp\|\} g_\perp = -\alpha g_\perp$$

Solution to TR subproblem

$$
\begin{aligned}
s^* = Pv^* &= -P_{\|}\mathrm{diag}(a)P_{\|}^T g - \alpha P_{\perp} P_{\perp}^T g \\
&= -\alpha g - P_{\|}\left(\mathrm{diag}(a) - \alpha I\right) P_{\|}^T g
\end{aligned}
$$

Convergence properties

Theorem

Let $f : R^n \mapsto R^1$ be twice continuously differentiable and bounded from below on R^n. Suppose that there exists a scalar $c > 0$ such that

$$\|f''(x)\| \leq c, \quad \forall x \in R^n.$$

Then

$$\lim_{k \to \infty} \|\nabla f(x_k)\| = 0.$$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V \quad \Rightarrow \quad$ cheap Cholesky factorization $V^T V = R^T R$
  (updating is even cheaper)
- implicit $Q \ (= VR^{-1}) \quad \Rightarrow \quad P_{\|}^T g = U^T R^{-T} V^T g$

  available $V^T g \quad \Rightarrow \quad$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s = -\alpha g - P_{\|} w$,
  where $w = (\mathrm{diag}(a) - \alpha I) P_{\|}^T g$ and $P_{\|} = QU = VR^{-1}U$

  available $V^T g \quad \Rightarrow \quad$ cheap $V^T s = \begin{bmatrix} S^T s \\ Y^T s \end{bmatrix}$
  $= -\alpha V^T g - (V^T V) R^{-1} U w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V$ $\Rightarrow$ cheap Cholesky factorization $V^T V = R^T R$
  (updating is even cheaper)
- implicit Q $(= VR^{-1})$ $\Rightarrow$ $P_{\|}^T g = U^T R^{-T} V^T g$
  available $V^T g$ $\Rightarrow$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s = -\alpha g - P_{\|} w$,
  where $w = (\mathrm{diag}(a) - \alpha I) P_{\|}^T g$ and $P_{\|} = QU = VR^{-1}U$
  available $V^T g$ $\Rightarrow$ cheap $V^T s = \begin{bmatrix} S^T s \\ Y^T s \end{bmatrix}$
  $= -\alpha V^T g - (V^T V) R^{-1} U w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V \quad \Rightarrow \quad$ cheap Cholesky factorization $V^T V = R^T R$ (updating is even cheaper)
- implicit $Q\ (= VR^{-1}) \quad \Rightarrow \quad P_{\|}^T g = U^T R^{-T} V^T g$

  available $V^T g \quad \Rightarrow \quad$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s = -\alpha g - P_{\|} w$,
  where $w = (\mathrm{diag}(a) - \alpha I)\, P_{\|}^T g$ and $P_{\|} = QU = VR^{-1}U$

  available $V^T g \quad \Rightarrow \quad$ cheap $V^T s = \begin{bmatrix} S^T s \\ Y^T s \end{bmatrix}$
  $= -\alpha V^T g - (V^T V) R^{-1} U w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V$ $\Rightarrow$ cheap Cholesky factorization $V^T V = R^T R$ (updating is even cheaper)
- implicit Q $(= VR^{-1})$ $\Rightarrow$ $P_{\|}^T g = U^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$

- $s = -\alpha g - P_{\|} w$,
  where $w = (\mathrm{diag}(a) - \alpha I) P_{\|}^T g$ and $P_{\|} = QU = VR^{-1}U$

  available $V^T g$ $\Rightarrow$ cheap $V^T s = \begin{bmatrix} S^T s \\ Y^T s \end{bmatrix}$
  $= -\alpha V^T g - (V^T V) R^{-1} U w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V$ $\Rightarrow$ cheap Cholesky factorization $V^T V = R^T R$ (updating is even cheaper)
- implicit Q $(= VR^{-1})$ $\Rightarrow$ $P_{\|}^T g = U^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s = -\alpha g - P_{\|} w$,

  where $w = (\mathrm{diag}(a) - \alpha I) P_{\|}^T g$ and $P_{\|} = QU = VR^{-1}U$

  available $V^T g$ $\Rightarrow$ cheap $V^T s = \begin{bmatrix} S^T s \\ Y^T s \end{bmatrix}$
  $= -\alpha V^T g - (V^T V) R^{-1} U w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V$ $\Rightarrow$ cheap Cholesky factorization $V^T V = R^T R$
  (updating is even cheaper)
- implicit Q $(= VR^{-1})$ $\Rightarrow$ $P_{\|}^T g = U^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s = -\alpha g - P_{\|} w$,
  where $w = (\mathrm{diag}(a) - \alpha I) P_{\|}^T g$ and $P_{\|} = QU = VR^{-1}U$

  available $V^T g$ $\Rightarrow$ cheap $V^T s = \begin{bmatrix} S^T s \\ Y^T s \end{bmatrix}$
  $= -\alpha V^T g - (V^T V) R^{-1} U w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V \quad \Rightarrow \quad$ cheap Cholesky factorization $V^T V = R^T R$
  (updating is even cheaper)
- implicit $Q \ (= VR^{-1}) \quad \Rightarrow \quad P_{\|}^T g = U^T R^{-T} V^T g$

  available $V^T g \quad \Rightarrow \quad$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s = -\alpha g - P_{\|} w$,
  where $w = (\mathrm{diag}(a) - \alpha I) P_{\|}^T g$ and $P_{\|} = QU = VR^{-1}U$

  available $V^T g \quad \Rightarrow \quad$ cheap $V^T s = \begin{bmatrix} S^T s \\ Y^T s \end{bmatrix}$
  $= -\alpha V^T g - (V^T V) R^{-1} U w$

Computational cost of one iteration

Typical values: $m = 10,\ n \geq 10^3$

- eigenvalue decomposition of $R_k W_k R_k^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $P_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $P_{\|} \cdot \left(\mathrm{diag}(a) - \frac{1}{\gamma} I\right) P_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the cost is approximately the same as for LBFGS

Computational cost of one iteration

Typical values: $m = 10,\ n \geq 10^3$

- eigenvalue decomposition of $R_k W_k R_k^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $P_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $P_{\|} \cdot \left(\mathrm{diag}(a) - \frac{1}{\gamma} I\right) P_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the cost is approximately the same as for LBFGS

Computational cost of one iteration

Typical values: $m = 10,\ n \geq 10^3$

- eigenvalue decomposition of $R_k W_k R_k^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $P_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $P_{\|} \cdot \left(\mathrm{diag}(a) - \frac{1}{\gamma} I\right) P_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the cost is approximately the same as for LBFGS

Computational cost of one iteration

Typical values: $m = 10,\ n \geq 10^3$

- eigenvalue decomposition of $R_k W_k R_k^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $P_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $P_{\|} \cdot \left(\text{diag}(a) - \frac{1}{\gamma} I\right) P_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the cost is approximately the same as for LBFGS

Computational cost of one iteration

Typical values: $m = 10,\ n \geq 10^3$

- eigenvalue decomposition of $R_k W_k R_k^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $P_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $P_{\|} \cdot \left(\text{diag}(a) - \frac{1}{\gamma} I\right) P_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the cost is approximately the same as for LBFGS

Computational cost of one iteration

Typical values: $m = 10,\ n \geq 10^3$

- eigenvalue decomposition of $R_k W_k R_k^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $P_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $P_{\|} \cdot \left(\mathrm{diag}(a) - \frac{1}{\gamma} I\right) P_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the cost is approximately the same as for LBFGS

Computational cost of one iteration

Typical values: $m = 10,\ n \geq 10^3$

- eigenvalue decomposition of $R_k W_k R_k^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $P_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $P_{\|} \cdot \left(\mathrm{diag}(a) - \frac{1}{\gamma} I\right) P_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the cost is approximately the same as for LBFGS

Computational cost of one iteration

Typical values: $m = 10,\ n \geq 10^3$

- eigenvalue decomposition of $R_k W_k R_k^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $P_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $P_{\|} \cdot \left(\text{diag}(a) - \frac{1}{\gamma} I\right) P_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the cost is approximately the same as for LBFGS

Computational cost of one iteration

Typical values: $m = 10,\ n \geq 10^3$

- eigenvalue decomposition of $R_k W_k R_k^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $P_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $P_{\|} \cdot \left(\text{diag}(a) - \frac{1}{\gamma} I\right) P_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the cost is approximately the same as for LBFGS

Alternative approach

Another vector norm

$$\|s\|_{P,2} = \max\{\|P_{\|}^T s\|_2, \|P_{\perp}^T s\|_2\} = \max\{\|v_{\|}\|_2, \|v_{\perp}\|_2\}$$

Norm equivalence

$$\frac{1}{\sqrt{2}}\|s\|_2 \leq \|s\|_{P,2} \leq \|s\|_2$$

Trust region

$$\|s\|_{P,2} \leq \Delta \iff \begin{cases} \|v_{\|}\|_2 \leq \Delta \\ \|v_{\perp}\|_2 \leq \Delta \end{cases}$$

TR subproblem decomposition

$$\min_{\|s\|_{P,2} \leq \Delta} q(s) = \min_{\|v_{\|}\|_2 \leq \Delta} q_{\|}(v_{\|}) + \min_{\|v_{\perp}\|_2 \leq \Delta} q_{\perp}(v_{\perp})$$

Alternative approach

Another vector norm

$$\|s\|_{P,2} = \max\{\|P_{\|}^T s\|_2, \|P_{\perp}^T s\|_2\} = \max\{\|v_{\|}\|_2, \|v_{\perp}\|_2\}$$

Norm equivalence

$$\frac{1}{\sqrt{2}}\|s\|_2 \leq \|s\|_{P,2} \leq \|s\|_2$$

Trust region

$$\|s\|_{P,2} \leq \Delta \iff \begin{cases} \|v_{\|}\|_2 \leq \Delta \\ \|v_{\perp}\|_2 \leq \Delta \end{cases}$$

TR subproblem decomposition

$$\min_{\|s\|_{P,2} \leq \Delta} q(s) = \min_{\|v_{\|}\|_2 \leq \Delta} q_{\|}(v_{\|}) + \min_{\|v_{\perp}\|_2 \leq \Delta} q_{\perp}(v_{\perp})$$

Alternative approach

Another vector norm

$$\|s\|_{P,2} = \max\{\|P_{\|}^T s\|_2, \|P_{\perp}^T s\|_2\} = \max\{\|v_{\|}\|_2, \|v_{\perp}\|_2\}$$

Norm equivalence

$$\frac{1}{\sqrt{2}}\|s\|_2 \leq \|s\|_{P,2} \leq \|s\|_2$$

Trust region

$$\|s\|_{P,2} \leq \Delta \quad \Longleftrightarrow \quad \begin{cases} \|v_{\|}\|_2 \leq \Delta \\ \|v_{\perp}\|_2 \leq \Delta \end{cases}$$

TR subproblem decomposition

$$\min_{\|s\|_{P,2} \leq \Delta} q(s) = \min_{\|v_{\|}\|_2 \leq \Delta} q_{\|}(v_{\|}) + \min_{\|v_{\perp}\|_2 \leq \Delta} q_{\perp}(v_{\perp})$$

Alternative approach

Another vector norm

$$\|s\|_{P,2} = \max\{\|P_{\|}^T s\|_2, \|P_{\perp}^T s\|_2\} = \max\{\|v_{\|}\|_2, \|v_{\perp}\|_2\}$$

Norm equivalence

$$\frac{1}{\sqrt{2}}\|s\|_2 \leq \|s\|_{P,2} \leq \|s\|_2$$

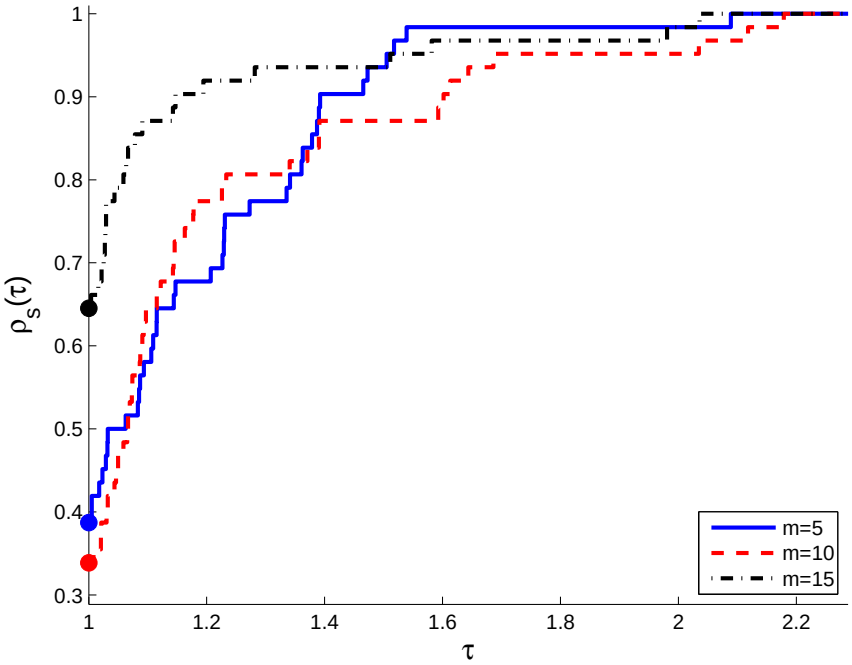
Trust region

$$\|s\|_{P,2} \leq \Delta \quad \Longleftrightarrow \quad \begin{cases} \|v_{\|}\|_2 \leq \Delta \\ \|v_{\perp}\|_2 \leq \Delta \end{cases}$$

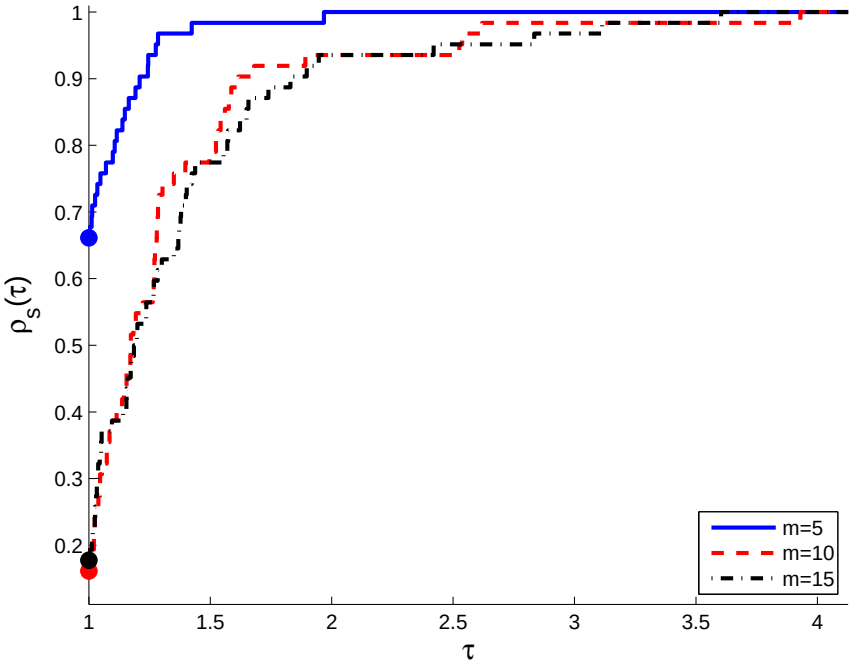
TR subproblem decomposition

$$\min_{\|s\|_{P,2} \leq \Delta} q(s) = \min_{\|v_{\|}\|_2 \leq \Delta} q_{\|}(v_{\|}) + \min_{\|v_{\perp}\|_2 \leq \Delta} q_{\perp}(v_{\perp})$$

Performance profiles for EIG$(\infty, 2)$ and $m = 10, 20, 30$

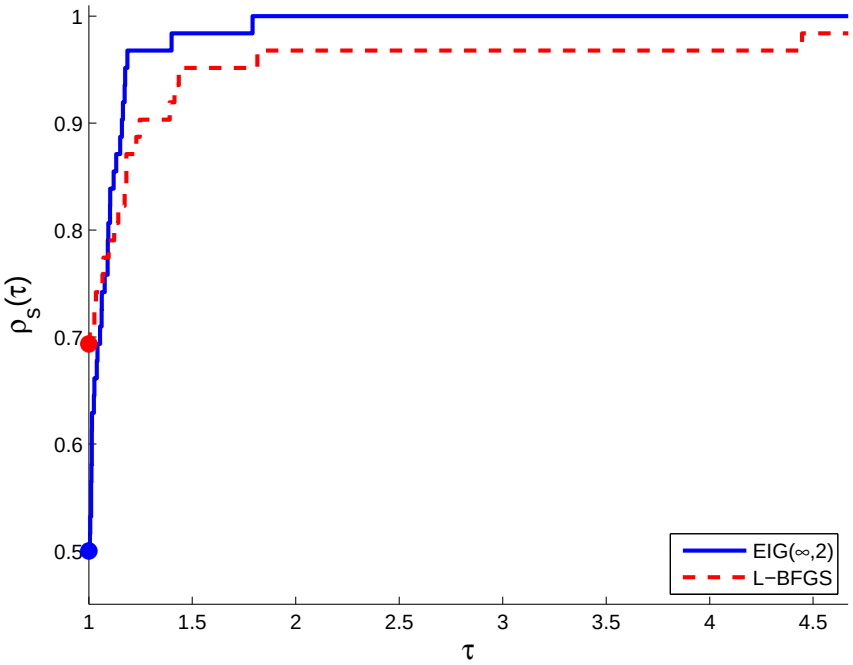


(a) Number of iterations

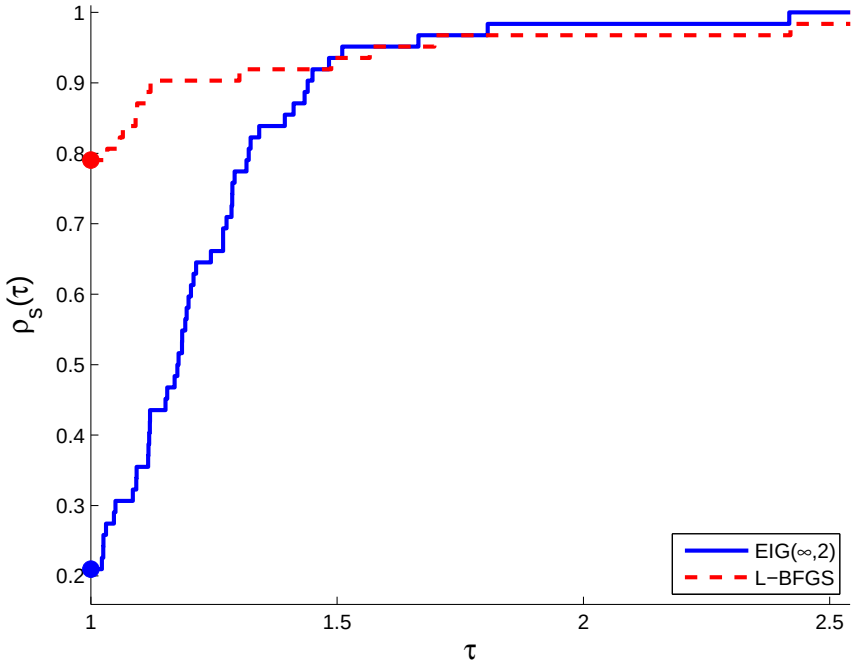


(b) CPU time

Performance profiles for EIG$(\infty, 2)$ and L-BFGS



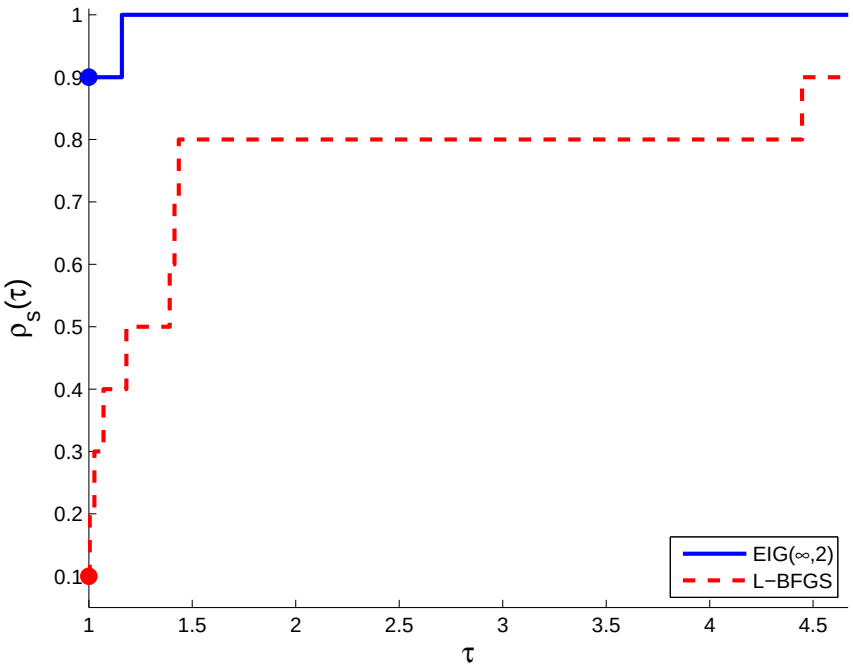
(c) Number of iterations



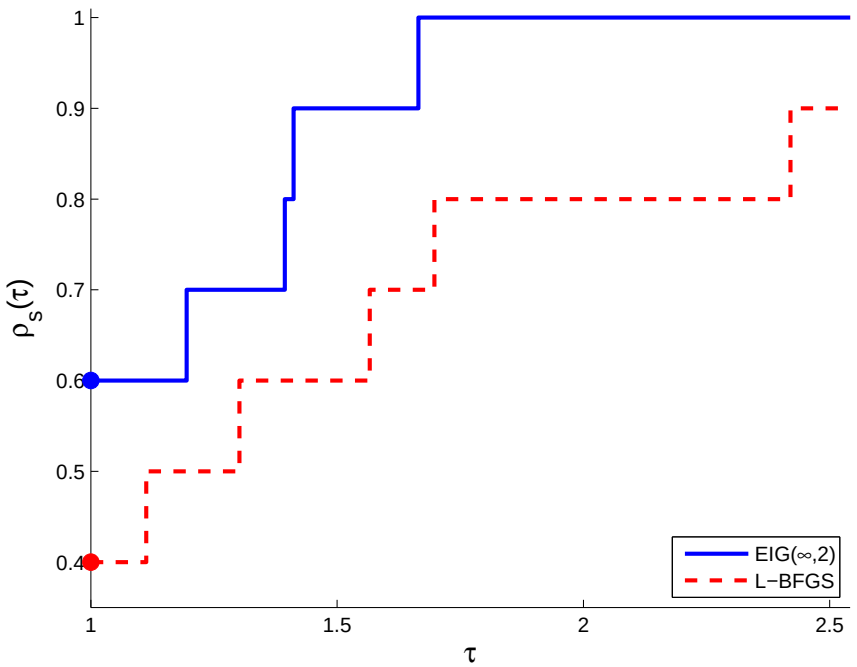
(d) CPU time

Performance profiles for EIG$(\infty, 2)$ and L-BFGS, reduced test set

The reduced test set: problems (10 in total), where the step-size one was rejected by L-BFGS in, at least, 30% of iterations



(e) Number of iterations



(f) CPU time

Part 2. Combination of limited memory technique and cubic regularization

Joint work with:
Ya-xiang Yuan and Liang Zhao

Unconstrained minimization

$$\min_{x \in R^n} f(x)$$

Cubic regularization framework

$$x_{k+1} = x_k + s_k$$

CR subproblem:

$$\min_{s \in R^n} m_k(s) = g_k^T s + \frac{1}{2} s^T B_k s + \frac{\mu}{3} (\varphi_k(s))^3$$

Traditional choice: $\varphi_k(s) = \|s\|_2$
Trial point $x_k + s^*$
If successful, $x_{k+1} = x_k + s^*$

Unconstrained minimization

$$\min_{x \in R^n} f(x)$$

Cubic regularization framework

$$x_{k+1} = x_k + s_k$$

CR subproblem:

$$\min_{s \in R^n} m_k(s) = g_k^T s + \frac{1}{2} s^T B_k s + \frac{\mu}{3} (\varphi_k(s))^3$$

Traditional choice: $\varphi_k(s) = \|s\|_2$
Trial point $x_k + s^*$
If successful, $x_{k+1} = x_k + s^*$

Convergence analysis

Standard assumptions, and

$$c_1\|s\|_2 \leq \varphi_k(s) \leq c_2\|s\|_2, \quad \forall k \geq 0, s \in R^n,$$

$$\|B_k\| \leq c_3,$$

$$m_k(s_k) \leq c_4 m_k(s_k^C)$$

imply

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Eigendecomposition-based cubic regularization

Eigenvalue decomposition: $B = U\Lambda U^T$

Eigenvalues: $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$; eigenvectors: $U \in R^{n\times n}$, $U^T U = I$

Special choice of $\varphi(s)$: $\|s\|_U = \|U^T s\|_3$

Norm equivalence:

$$n^{-1/6}\|s\|_2 \leq \|s\|_U \leq \|s\|_2$$

Remark: the bounds doesn't depend on U or the iteration number

Eigendecomposition-based cubic regularization

Eigenvalue decomposition: $B = U\Lambda U^T$

Eigenvalues: $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$; eigenvectors: $U \in R^{n\times n}$, $U^T U = I$

Special choice of $\varphi(s)$: $\|s\|_U = \|U^T s\|_3$

Norm equivalence:

$$n^{-1/6}\|s\|_2 \leq \|s\|_U \leq \|s\|_2$$

Remark: the bounds doesn't depend on U or the iteration number

Eigendecomposition-based CR model

$$m_U(s) = g^T s + \frac{1}{2} s^T B s + \frac{\mu}{3} \|s\|_U^3$$

- If $\varphi(s) = \|s\|_2$, the corresponding CR model has, at most, two local minima (Martinez, 1994).
  It is not easy to find a global minimizer $\Rightarrow$ approximate solution.
- If $\varphi(s) = \|s\|_U$, the model $m_U(s)$ may have 2^n distinct local minima, i.e. only one of them is global.
- Exact global minimizer of $m_U(s)$ is obtained in closed form.

Eigendecomposition-based CR model

$$m_U(s) = g^T s + \frac{1}{2} s^T B s + \frac{\mu}{3} \|s\|_U^3$$

- If $\varphi(s) = \|s\|_2$, the corresponding CR model has, at most, two local minima (Martinez, 1994).
  It is not easy to find a global minimizer $\Rightarrow$ approximate solution.
- If $\varphi(s) = \|s\|_U$, the model $m_U(s)$ may have 2^n distinct local minima, i.e. only one of them is global.
- Exact global minimizer of $m_U(s)$ is obtained in closed form.

Eigendecomposition-based CR model

$$m_U(s) = g^T s + \frac{1}{2} s^T B s + \frac{\mu}{3} \|s\|_U^3$$

- If $\varphi(s) = \|s\|_2$, the corresponding CR model has, at most, two local minima (Martinez, 1994).
  It is not easy to find a global minimizer $\Rightarrow$ approximate solution.
- If $\varphi(s) = \|s\|_U$, the model $m_U(s)$ may have 2^n distinct local minima, i.e. only one of them is global.
- Exact global minimizer of $m_U(s)$ is obtained in closed form.

Eigendecomposition-based CR model

$$m_U(s) = g^T s + \frac{1}{2} s^T B s + \frac{\mu}{3} \|s\|_U^3$$

- If $\varphi(s) = \|s\|_2$, the corresponding CR model has, at most, two local minima (Martinez, 1994).
  It is not easy to find a global minimizer $\Rightarrow$ approximate solution.
- If $\varphi(s) = \|s\|_U$, the model $m_U(s)$ may have 2^n distinct local minima, i.e. only one of them is global.
- Exact global minimizer of $m_U(s)$ is obtained in closed form.

Decomposition of $m_U(s)$ in space of new variables

Denote $\bar{s} = U^T s$ (new variables) and $\bar{g} = U^T g$

CR model in the space of new variables:

$$\bar{m}(\bar{s}) = \bar{g}^T \bar{s} + \frac{1}{2} \bar{s}^T \Lambda \bar{s} + \frac{\mu}{3} \|\bar{s}\|_3^3$$

CR subproblem decomposition:

$$\min_{\bar{s} \in R^n} \bar{m}(\bar{s}) = \sum_{i=1}^{n} \min_{\bar{s}_i} \left(\bar{g}_i \bar{s}_i + \frac{\lambda_i}{2} \bar{s}_i^2 + \frac{\mu}{3} |\bar{s}_i|^3 \right)$$

In Martinez and Raydan (2015), $\bar{s}_i^3$ is used instead of our $|\bar{s}_i|^3$

Decomposition of $m_U(s)$ in space of new variables

Denote $\bar{s} = U^T s$ (new variables) and $\bar{g} = U^T g$
CR model in the space of new variables:

$$\bar{m}(\bar{s}) = \bar{g}^T \bar{s} + \frac{1}{2}\bar{s}^T \Lambda \bar{s} + \frac{\mu}{3}\|\bar{s}\|_3^3$$

CR subproblem decomposition:

$$\min_{\bar{s}\in R^n} \bar{m}(\bar{s}) = \sum_{i=1}^{n} \min_{\bar{s}_i} \left(\bar{g}_i \bar{s}_i + \frac{\lambda_i}{2}\bar{s}_i^2 + \frac{\mu}{3}|\bar{s}_i|^3 \right)$$

In Martinez and Raydan (2015), $\bar{s}_i^3$ is used instead of our $|\bar{s}_i|^3$

Decomposition of $m_U(s)$ in space of new variables

Denote $\bar{s} = U^T s$ (new variables) and $\bar{g} = U^T g$
CR model in the space of new variables:

$$\bar{m}(\bar{s}) = \bar{g}^T \bar{s} + \frac{1}{2}\bar{s}^T \Lambda \bar{s} + \frac{\mu}{3}\|\bar{s}\|_3^3$$

CR subproblem decomposition:

$$\min_{\bar{s}\in R^n} \bar{m}(\bar{s}) = \sum_{i=1}^{n} \min_{\bar{s}_i} \left(\bar{g}_i \bar{s}_i + \frac{\lambda_i}{2}\bar{s}_i^2 + \frac{\mu}{3}|\bar{s}_i|^3 \right)$$

In Martinez and Raydan (2015), $\bar{s}_i^3$ is used instead of our $|\bar{s}_i|^3$

Decomposition of $m_U(s)$ in space of new variables

Denote $\bar{s} = U^T s$ (new variables) and $\bar{g} = U^T g$
CR model in the space of new variables:

$$\bar{m}(\bar{s}) = \bar{g}^T \bar{s} + \frac{1}{2}\bar{s}^T \Lambda \bar{s} + \frac{\mu}{3}\|\bar{s}\|_3^3$$

CR subproblem decomposition:

$$\min_{\bar{s}\in R^n} \bar{m}(\bar{s}) = \sum_{i=1}^{n} \min_{\bar{s}_i} \left(\bar{g}_i \bar{s}_i + \frac{\lambda_i}{2}\bar{s}_i^2 + \frac{\mu}{3}|\bar{s}_i|^3 \right)$$

In Martinez and Raydan (2015), $\bar{s}_i^3$ is used instead of our $|\bar{s}_i|^3$

Solution to CR subproblem

In the new space:

$$\bar{s}^* = -C\bar{g},$$

where $C = \mathrm{diag}(c_1, \ldots, c_n)$ and

$$c_i = \frac{2}{\lambda_i + \sqrt{\lambda_i^2 + 4\mu|\bar{g}_i|}}$$

In the original space:

$$s^* = U\bar{s}^* = -UCU^T g$$

Solution to CR subproblem

In the new space:

$$\bar{s}^* = -C\bar{g},$$

where $C = \mathrm{diag}(c_1, \ldots, c_n)$ and

$$c_i = \frac{2}{\lambda_i + \sqrt{\lambda_i^2 + 4\mu|\bar{g}_i|}}$$

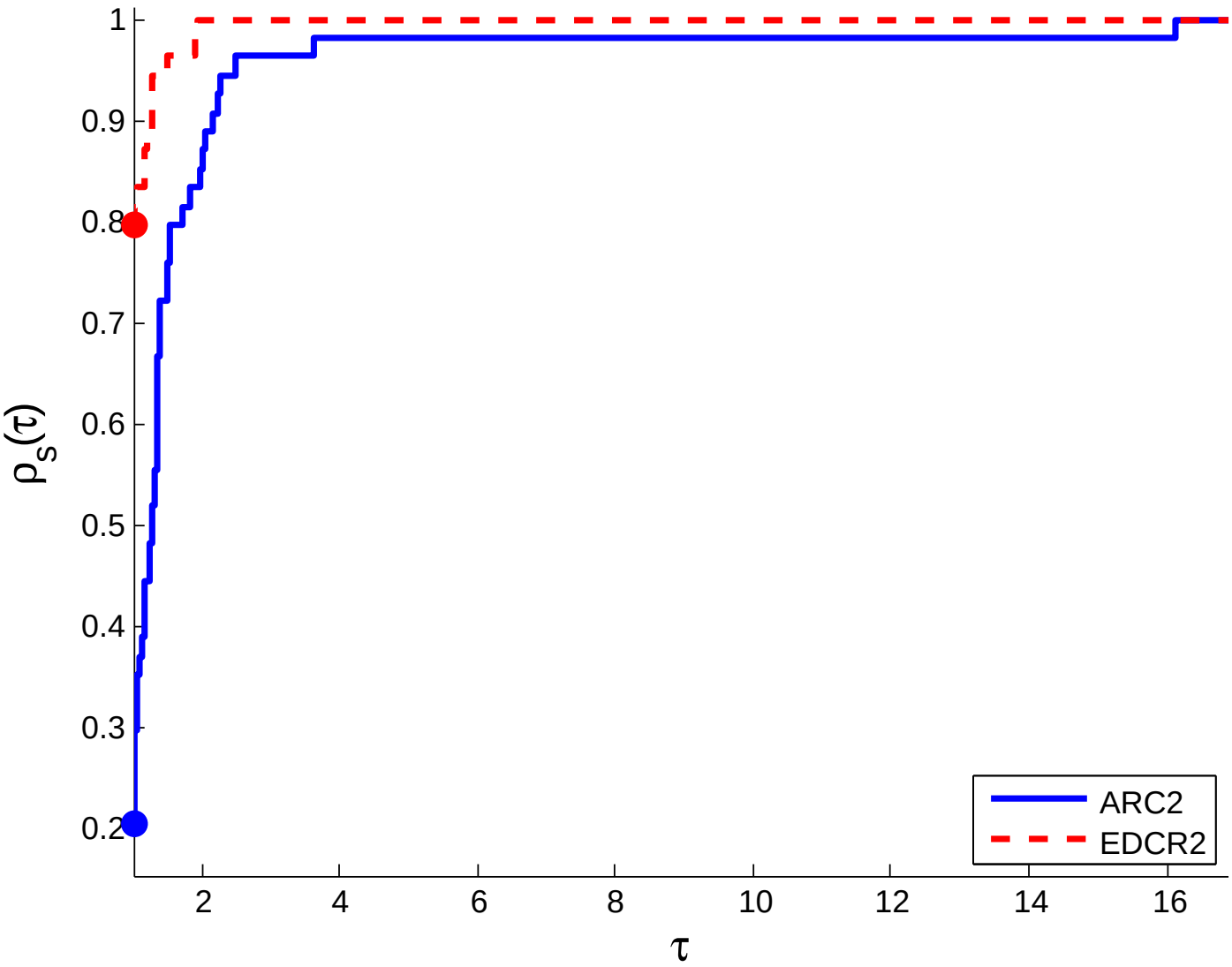
In the original space:

$$s^* = U\bar{s}^* = -UCU^T g$$

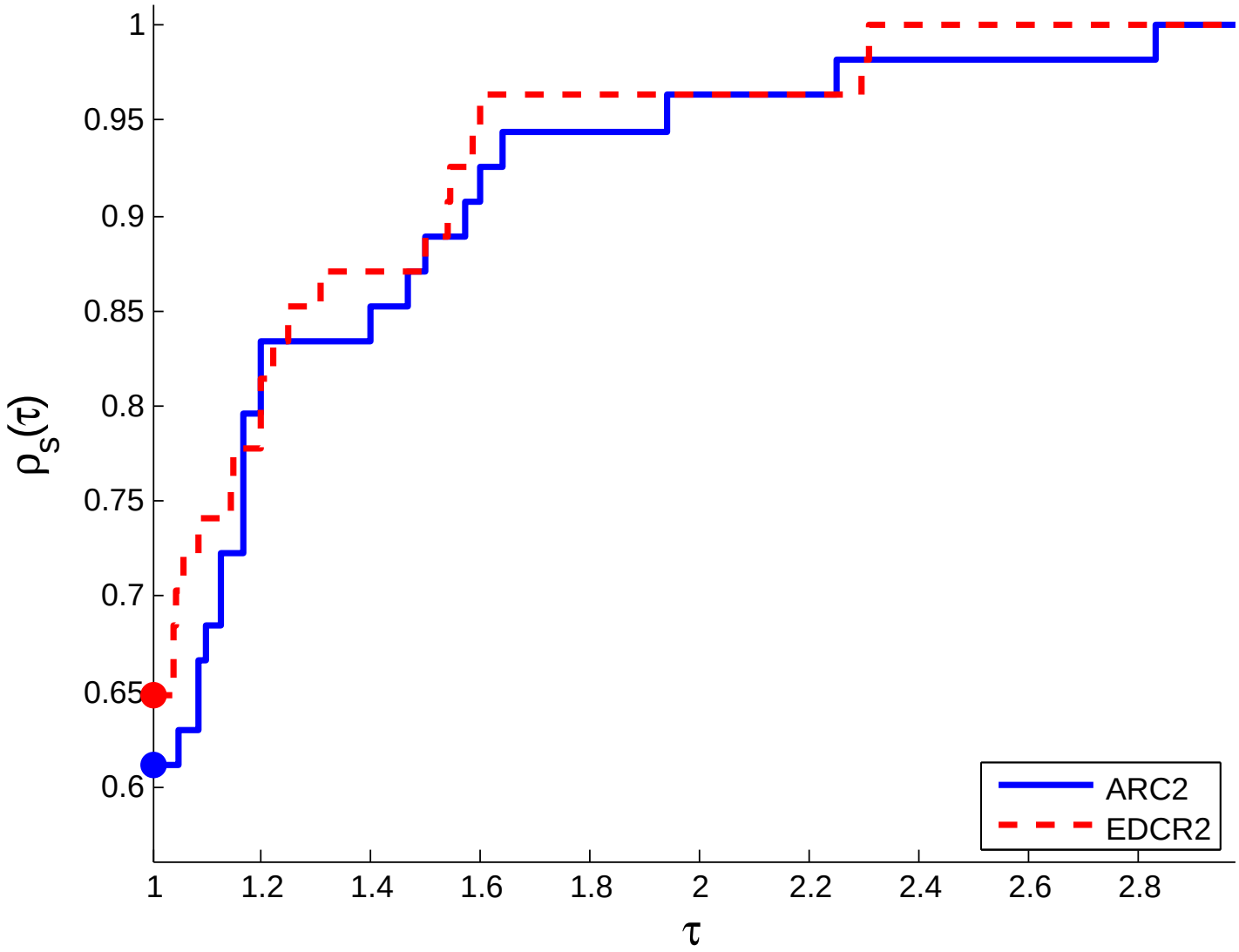
Numerical experiments

- CUTEst set of test problems
- $1000 \leq n \leq 1500$
- exact Hessian
- standard Euclidean norm for cubic regularization (ARC2) vs. eigendecomposition-based cubic regularization (EDCR2)
- CPU time was averaged over five runs

CPU time



Number of iterations



The new norm vs the Euclidean norm

Conclusion:

the eigendecomposition-based norm is more suitable for the cubic regularization than the Euclidean norm.

Hessian approximation

Quasi-Newton approximation

$$B_{k+1} = B_k + \Delta B(B_k, s_k, y_k, \ldots)$$

$s_k = x_{k+1} - x_k,$
$y_k = \nabla f(_{k+1}) - \nabla f(_k) = g_{k+1} - g_k$

Compact representation for limited-memory QN approximation

$$B_{k+1} = \gamma_k I + \begin{bmatrix} V_k \end{bmatrix} \begin{bmatrix} W_k \end{bmatrix} \underbrace{\begin{bmatrix} V_k^T \end{bmatrix}}_{n} \Big\} m \qquad m \ll n$$

Hessian approximation

Quasi-Newton approximation

$$B_{k+1} = B_k + \Delta B(B_k, s_k, y_k, \ldots)$$

$s_k = x_{k+1} - x_k,$
$y_k = \nabla f(_{k+1}) - \nabla f(_k) = g_{k+1} - g_k$

Compact representation for limited-memory QN approximation

$$B_{k+1} = \gamma_k I + \boxed{V_k} \boxed{W_k} \underbrace{\boxed{V_k^T}\Big\} m}_{n} \qquad m \ll n$$

Limited-memory approximation in CR framework

Straightforward implementation of limited-memory approximation in CR framework deteriorates its efficiency.

Aim:

to make use of the norm $\varphi_k(s) = \|s\|_U$ in order to simplify the solution of the CR subproblem and retain the low-cost property of limited-memory iterations.

Q:

How to efficiently calculate the eigendecomposition of B_k?

Tool:

Implicit eigendecomposition of limited-memory Hessian approximation introduced in

Burdakov, Gong, Zikrin and Yuan.

On efficiently combining limited-memory and trust-region techniques.

Mathematical Programming Computation (2017)

Limited-memory approximation in CR framework

Straightforward implementation of limited-memory approximation in CR framework deteriorates its efficiency.

Aim:

to make use of the norm $\varphi_k(s) = \|s\|_U$ in order to simplify the solution of the CR subproblem and retain the low-cost property of limited-memory iterations.

Q:

How to efficiently calculate the eigendecomposition of B_k?

Tool:

Implicit eigendecomposition of limited-memory Hessian approximation introduced in

Burdakov, Gong, Zikrin and Yuan.

On efficiently combining limited-memory and trust-region techniques.

Mathematical Programming Computation (2017)

Limited-memory approximation in CR framework

Straightforward implementation of limited-memory approximation in CR framework deteriorates its efficiency.

Aim:

to make use of the norm $\varphi_k(s) = \|s\|_U$ in order to simplify the solution of the CR subproblem and retain the low-cost property of limited-memory iterations.

Q:

How to efficiently calculate the eigendecomposition of B_k?

Tool:

Implicit eigendecomposition of limited-memory Hessian approximation introduced in

Burdakov, Gong, Zikrin and Yuan.

On efficiently combining limited-memory and trust-region techniques.

Mathematical Programming Computation (2017)

Limited-memory approximation in CR framework

Straightforward implementation of limited-memory approximation in CR framework deteriorates its efficiency.

Aim:

to make use of the norm $\varphi_k(s) = \|s\|_U$ in order to simplify the solution of the CR subproblem and retain the low-cost property of limited-memory iterations.

Q:

How to efficiently calculate the eigendecomposition of B_k?

Tool:

Implicit eigendecomposition of limited-memory Hessian approximation introduced in

Burdakov, Gong, Zikrin and Yuan.

On efficiently combining limited-memory and trust-region techniques.

Mathematical Programming Computation (2017)

New basis

$$\underbrace{V = QR}_{QR\text{-decomposition}} \quad \Longrightarrow \quad B = \gamma I + VWV^T = \gamma I + Q(RWR^T)Q^T$$

$$\underbrace{RWR^T = PDP^T}_{\text{eigenvalue decomposition}} \quad \Longrightarrow \quad B = \gamma I + U_{\|} D U_{\|}^T,$$

where $U_{\|} = QP \in R^{n \times m}$ is orthonormal ($U_{\|}^T U_{\|} = I$).

$U = [U_{\|}, U_{\perp}] \in R^{n \times n}$ is an orthonormal basis in R^n.

New basis

$$\underbrace{V = QR}_{QR\text{-decomposition}} \quad \Longrightarrow \quad B = \gamma I + VWV^T = \gamma I + Q(RWR^T)Q^T$$

$$\underbrace{RWR^T = PDP^T}_{\text{eigenvalue decomposition}} \quad \Longrightarrow \quad B = \gamma I + U_{\|} D U_{\|}^T,$$

where $U_{\|} = QP \in R^{n \times m}$ is orthonormal ($U_{\|}^T U_{\|} = I$).

$U = [U_{\|}, U_{\perp}] \in R^{n \times n}$ is an orthonormal basis in R^n.

New basis

$$\underbrace{V = QR}_{QR\text{-decomposition}} \quad \Longrightarrow \quad B = \gamma I + VWV^T = \gamma I + Q(RWR^T)Q^T$$

$$\underbrace{RWR^T = PDP^T}_{\text{eigenvalue decomposition}} \quad \Longrightarrow \quad B = \gamma I + U_{\|} D U_{\|}^T,$$

where $U_{\|} = QP \in R^{n \times m}$ is orthonormal ($U_{\|}^T U_{\|} = I$).

$U = [U_{\|}, U_{\perp}] \in R^{n \times n}$ is an orthonormal basis in R^n.

Eigenvalues of B

- $Bu = \gamma u,\ \forall u \in U_\perp \implies n - m$ eigenvalues γ with eigenspace $U_\perp$
- $Bu_i = (\gamma + d_i)u_i,\ \forall u_i$, the i-th column of $U_\parallel \implies m$ eigenvalues $\lambda_i = \gamma + d_i$, where d_i is the i-th eigenvalue of RWR^T

Eigenvalues of B

- $Bu = \gamma u,\ \forall u \in U_\perp \quad \Longrightarrow \quad n - m$ eigenvalues γ with eigenspace $U_\perp$
- $Bu_i = (\gamma + d_i)u_i,\ \forall u_i$, the i-th column of $U_\parallel \quad \Longrightarrow \quad m$ eigenvalues $\lambda_i = \gamma + d_i$, where d_i is the i-th eigenvalue of RWR^T

Induced splitting

$$U = [U_{\|}, U_{\perp}] \quad \Rightarrow$$

$$\bar{s}^*_{\|} = U_{\|}^T s^*, \quad \bar{s}^*_{\perp} = U_{\perp}^T s^*$$
$$\bar{g}_{\|} = U_{\|}^T g, \quad \bar{g}_{\perp} = U_{\perp}^T g$$

Choice of $U_\perp$

Observations:

- γ is a multiple eigenvalue $\Rightarrow$ $U_\perp$ is not uniquely defined
- s^* depends on the choice of $U_\perp$
- $\bar{g}_\perp$ is used for computing $\bar{s}^*_\perp$, and it requires $U_\perp^T g$
- $U_\perp^T g$ may be prohibitively too expensive,
  unless to choose the large matrix $U_\perp$ in a special way

Our suggestion:

$$u_{m+1} = g_\perp / \|g_\perp\|_2, \quad \text{where} \quad g_\perp = (I_n - U_\parallel U_\parallel^T) g$$

$\Rightarrow$ s^* does not depend on the rest of the columns of $U_\perp$.
If $g_\perp = 0$, s^* does not depend on $U_\perp$.

Choice of $U_\perp$

Observations:

- γ is a multiple eigenvalue $\Rightarrow$ $U_\perp$ is not uniquely defined
- s^* depends on the choice of $U_\perp$
- $\bar{g}_\perp$ is used for computing $\bar{s}^*_\perp$, and it requires $U_\perp^T g$
- $U_\perp^T g$ may be prohibitively too expensive,
  unless to choose the large matrix $U_\perp$ in a special way

Our suggestion:

$$u_{m+1} = g_\perp / \|g_\perp\|_2, \quad \text{where} \quad g_\perp = (I_n - U_\parallel U_\parallel^T) g$$

$\Rightarrow$ s^* does not depend on the rest of the columns of $U_\perp$.
If $g_\perp = 0$, s^* does not depend on $U_\perp$.

Choice of $U_\perp$

Observations:

- γ is a multiple eigenvalue $\Rightarrow$ $U_\perp$ is not uniquely defined
- s^* depends on the choice of $U_\perp$
- $\bar{g}_\perp$ is used for computing $\bar{s}^*_\perp$, and it requires $U_\perp^T g$
- $U_\perp^T g$ may be prohibitively too expensive,
  unless to choose the large matrix $U_\perp$ in a special way

Our suggestion:

$$u_{m+1} = g_\perp / \|g_\perp\|_2, \quad \text{where} \quad g_\perp = (I_n - U_\| U_\|^T) g$$

$\Rightarrow$ s^* does not depend on the rest of the columns of $U_\perp$.
If $g_\perp = 0$, s^* does not depend on $U_\perp$.

Solution to CR subproblem

In the new space:

$$\bar{s}_{\|}^{*} = -C_{\|}\bar{g}_{\|} \quad \text{where} \quad C_{\|} = \operatorname{diag}(c_1, \ldots, c_m)$$

$$\bar{s}_{\perp}^{*} = -\alpha^{*}\bar{g}_{\perp} \quad \text{where} \quad \alpha^{*} = \frac{2}{\gamma + \sqrt{\gamma^2 + 4\mu\|g_{\perp}\|_2}}$$

In the original space:

$$s^{*} = U_{\|}\bar{s}_{\|}^{*} + U_{\perp}\bar{s}_{\perp}^{*} = -\alpha^{*}g + U_{\|}(\alpha^{*}I_m - C_{\|})U_{\|}^{T}g$$

Solution to CR subproblem

In the new space:

$$\bar{s}_{\|}^{*} = -C_{\|}\bar{g}_{\|} \quad \text{where} \quad C_{\|} = \operatorname{diag}(c_1, \ldots, c_m)$$

$$\bar{s}_{\perp}^{*} = -\alpha^{*}\bar{g}_{\perp} \quad \text{where} \quad \alpha^{*} = \frac{2}{\gamma + \sqrt{\gamma^2 + 4\mu\|g_{\perp}\|_2}}$$

In the original space:

$$s^{*} = U_{\|}\bar{s}_{\|}^{*} + U_{\perp}\bar{s}_{\perp}^{*} = -\alpha^{*}g + U_{\|}(\alpha^{*}I_m - C_{\|})U_{\|}^{T}g$$

Predicted function value calculation

$$m(s^*) = q(s^*) + \frac{\mu}{3}\left(\|C_{\|}\bar{g}_{\|}\|_3^3 + (\alpha^*)^3\|g_{\perp}\|_2^3\right)$$

where

$$q(s^*) = \bar{g}_{\|}^T\left(\frac{C_{\|}^2\Lambda_{\|}}{2} + C_{\|}\right)\bar{g}_{\|} + \frac{\delta(\alpha^*)^2 - 2\alpha^*}{2}\|g_{\perp}\|_2^2$$

is the quadratic model value

Alternative eigendecomposition-based cubic regularization

$$\varphi_U(s) = \sqrt[3]{\|U_{\|}^T s\|_3^3 + \|U_{\perp}^T s\|_2^3}$$

Properties:

- $\varphi_U(s)$ is a vector norm
- thought $\varphi_U(s)$ is not identical with $\|s\|_U$, their global minimizers are the same
- $(2m)^{-1/6}\|s\|_2 \leq \varphi_U(s) \leq \|s\|_2$

Alternative eigendecomposition-based cubic regularization

$$\varphi_U(s) = \sqrt[3]{\|U_{\|}^T s\|_3^3 + \|U_{\perp}^T s\|_2^3}$$

Properties:

- $\varphi_U(s)$ is a vector norm
- thought $\varphi_U(s)$ is not identical with $\|s\|_U$, their global minimizers are the same
- $(2m)^{-1/6}\|s\|_2 \leq \varphi_U(s) \leq \|s\|_2$

Alternative eigendecomposition-based cubic regularization

$$\varphi_U(s) = \sqrt[3]{\|U_{\|}^T s\|_3^3 + \|U_{\perp}^T s\|_2^3}$$

Properties:

- $\varphi_U(s)$ is a vector norm
- thought $\varphi_U(s)$ is not identical with $\|s\|_U$, their global minimizers are the same
- $(2m)^{-1/6}\|s\|_2 \leq \varphi_U(s) \leq \|s\|_2$

Alternative eigendecomposition-based cubic regularization

$$\varphi_U(s) = \sqrt[3]{\|U_{\|}^T s\|_3^3 + \|U_{\perp}^T s\|_2^3}$$

Properties:

- $\varphi_U(s)$ is a vector norm
- thought $\varphi_U(s)$ is not identical with $\|s\|_U$, their global minimizers are the same
- $(2m)^{-1/6}\|s\|_2 \leq \varphi_U(s) \leq \|s\|_2$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V \quad \Rightarrow \quad$ cheap Cholesky factorization $V^T V = R^T R$
  (updating is even cheaper)
- implicit $Q \ (= VR^{-1}) \quad \Rightarrow \quad U_{\|}^T g = P^T R^{-T} V^T g$

  available $V^T g \quad \Rightarrow \quad$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s^* = -\alpha^* g + Vw$,
  where $w = R^{-1} P (\alpha^* I_m - C_{\|}) P^T R^{-T} V^T g$

  available $V^T g \quad \Rightarrow \quad$ cheap $V^T s^* = \begin{bmatrix} S^T s^* \\ Y^T s^* \end{bmatrix}$
  $= -\alpha^* V^T g + (V^T V) w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V$ $\Rightarrow$ cheap Cholesky factorization $V^T V = R^T R$
  (updating is even cheaper)
- implicit Q $(= VR^{-1})$ $\Rightarrow$ $U_{\|}^T g = P^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s^* = -\alpha^* g + Vw$,
  where $w = R^{-1} P (\alpha^* I_m - C_{\|}) P^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T s^* = \begin{bmatrix} S^T s^* \\ Y^T s^* \end{bmatrix}$
  $= -\alpha^* V^T g + (V^T V) w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V \quad \Rightarrow \quad$ cheap Cholesky factorization $V^T V = R^T R$ (updating is even cheaper)
- implicit $Q \ (= VR^{-1}) \quad \Rightarrow \quad U_{\|}^T g = P^T R^{-T} V^T g$

  available $V^T g \quad \Rightarrow \quad$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s^* = -\alpha^* g + Vw$,
  where $w = R^{-1} P (\alpha^* I_m - C_{\|}) P^T R^{-T} V^T g$

  available $V^T g \quad \Rightarrow \quad$ cheap $V^T s^* = \begin{bmatrix} S^T s^* \\ Y^T s^* \end{bmatrix}$
  $= -\alpha^* V^T g + (V^T V) w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V$ $\Rightarrow$ cheap Cholesky factorization $V^T V = R^T R$ (updating is even cheaper)
- implicit Q $(= VR^{-1})$ $\Rightarrow$ $U_{\|}^T g = P^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s^* = -\alpha^* g + Vw$,
  where $w = R^{-1} P (\alpha^* I_m - C_{\|}) P^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T s^* = \begin{bmatrix} S^T s^* \\ Y^T s^* \end{bmatrix}$
  $= -\alpha^* V^T g + (V^T V) w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V$ $\Rightarrow$ cheap Cholesky factorization $V^T V = R^T R$ (updating is even cheaper)
- implicit Q $(= VR^{-1})$ $\Rightarrow$ $U_{\|}^T g = P^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s^* = -\alpha^* g + Vw$,
  where $w = R^{-1} P (\alpha^* I_m - C_{\|}) P^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T s^* = \begin{bmatrix} S^T s^* \\ Y^T s^* \end{bmatrix}$
  $= -\alpha^* V^T g + (V^T V) w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V$ $\Rightarrow$ cheap Cholesky factorization $V^T V = R^T R$
  (updating is even cheaper)
- implicit Q $(= VR^{-1})$ $\Rightarrow$ $U_{\|}^T g = P^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s^* = -\alpha^* g + Vw$,
  where $w = R^{-1} P(\alpha^* I_m - C_{\|}) P^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T s^* = \begin{bmatrix} S^T s^* \\ Y^T s^* \end{bmatrix}$
  $= -\alpha^* V^T g + (V^T V) w$

Implementation issues for L-BFGS

In L-BFGS update $B = \gamma I + VWV^T$,
$V = [S, Y] \in R^{n \times m}$, up to $m/2$ couples $\{s_i, y_i\}$ are stored,
$W \in R^{m \times m}$ involves scalar products $s_i^T s_j$ and $y_i^T s_j$

Important observations

- available $V^T V$ $\Rightarrow$ cheap Cholesky factorization $V^T V = R^T R$
  (updating is even cheaper)
- implicit Q $(= VR^{-1})$ $\Rightarrow$ $U_{\|}^T g = P^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T y = \begin{bmatrix} S^T y \\ Y^T y \end{bmatrix}$
- $s^* = -\alpha^* g + Vw$,
  where $w = R^{-1} P(\alpha^* I_m - C_{\|}) P^T R^{-T} V^T g$

  available $V^T g$ $\Rightarrow$ cheap $V^T s^* = \begin{bmatrix} S^T s^* \\ Y^T s^* \end{bmatrix}$
  $= -\alpha^* V^T g + (V^T V) w$

Computational cost of one iteration

Typical values: $m = 10$, $n \geq 10^3$

- eigenvalue decomposition of $RWR^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $U_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $U_{\|} \cdot \left(\alpha^* I_m - C_{\|}\right) U_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the major cost is $O(2mn)$,
the same as for L-BFGS with line-search

Computational cost of one iteration

Typical values: $m = 10$, $n \geq 10^3$

- eigenvalue decomposition of $RWR^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $U_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $U_{\|} \cdot \left(\alpha^* I_m - C_{\|}\right) U_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the major cost is $O(2mn)$,
the same as for L-BFGS with line-search

Computational cost of one iteration

Typical values: $m = 10$, $n \geq 10^3$

- eigenvalue decomposition of $RWR^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $U_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $U_{\|} \cdot \left(\alpha^* I_m - C_{\|}\right) U_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the major cost is $O(2mn)$,
the same as for L-BFGS with line-search

Computational cost of one iteration

Typical values: $m = 10$, $n \geq 10^3$

- eigenvalue decomposition of $RWR^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $U_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $U_{\|} \cdot \left(\alpha^* I_m - C_{\|}\right) U_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the major cost is $O(2mn)$,
the same as for L-BFGS with line-search

Computational cost of one iteration

Typical values: $m = 10$, $n \geq 10^3$

- eigenvalue decomposition of $RWR^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $U_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $U_{\|} \cdot \left(\alpha^* I_m - C_{\|}\right) U_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the major cost is $O(2mn)$,
the same as for L-BFGS with line-search

Computational cost of one iteration

Typical values: $m = 10,\ n \geq 10^3$

- eigenvalue decomposition of $RWR^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $U_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $U_{\|} \cdot \left(\alpha^* I_m - C_{\|}\right) U_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the major cost is $O(2mn)$,
the same as for L-BFGS with line-search

Computational cost of one iteration

Typical values: $m = 10$, $n \geq 10^3$

- eigenvalue decomposition of $RWR^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $U_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $U_{\|} \cdot \left(\alpha^* I_m - C_{\|}\right) U_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the major cost is $O(2mn)$,
the same as for L-BFGS with line-search

Computational cost of one iteration

Typical values: $m = 10$, $n \geq 10^3$

- eigenvalue decomposition of $RWR^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $U_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $U_{\|} \cdot \left(\alpha^* I_m - C_{\|}\right) U_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the major cost is $O(2mn)$,
the same as for L-BFGS with line-search

Computational cost of one iteration

Typical values: $m = 10$, $n \geq 10^3$

- eigenvalue decomposition of $RWR^T \in R^{m \times m} \Rightarrow O[(m^2/n)mn]$
- multiplication $U_{\|}^T \cdot g \Rightarrow \approx mn$
- multiplication $U_{\|} \cdot \left(\alpha^* I_m - C_{\|}\right) U_{\|}^T g \Rightarrow \approx mn$
- multiplication $V^T \cdot s \Rightarrow$ low cost
- updating $W \in R^{m \times m} \Rightarrow$ low cost
- implicit QR-factorization of $V \in R^{n \times m} \Rightarrow$ low cost
- quadratic function evaluation $q(s^*) \Rightarrow$ low cost

Conclusion: the major cost is $O(2mn)$,
the same as for L-BFGS with line-search

Part 3. Search over a model-based steepest descent path

Joint work with: Serge Gratton

$$q(s) = g^T s + \frac{1}{2} s^T B s$$

The steepest descent path $s(t)$:

$$\dot{s} = -\nabla q(s(t)), \quad s(0) = 0.$$

Let P be eigenvectors of B. New variables: $v = P^T s$.

The components of $v(t) = P^T s(t)$ are calculated as

$$v_i(t) = -\frac{\bar{g}_i}{\lambda_i} \left(1 - e^{-\lambda_i t}\right),$$

where λ_i are eigenvalues of B, and $\bar{g} = P^T g$.

Part 3. Search over a model-based steepest descent path

Joint work with: Serge Gratton

$$q(s) = g^T s + \frac{1}{2} s^T B s$$

The steepest descent path $s(t)$:

$$\dot{s} = -\nabla q(s(t)), \quad s(0) = 0.$$

Let P be eigenvectors of B. New variables: $v = P^T s$.

The components of $v(t) = P^T s(t)$ are calculated as

$$v_i(t) = -\frac{\bar{g}_i}{\lambda_i} \left(1 - e^{-\lambda_i t}\right),$$

where λ_i are eigenvalues of B, and $\bar{g} = P^T g$.

Part 3. Search over a model-based steepest descent path

Joint work with: Serge Gratton

$$q(s) = g^T s + \frac{1}{2} s^T B s$$

The steepest descent path $s(t)$:

$$\dot{s} = -\nabla q(s(t)), \quad s(0) = 0.$$

Let P be eigenvectors of B. New variables: $v = P^T s$.

The components of $v(t) = P^T s(t)$ are calculated as

$$v_i(t) = -\frac{\bar{g}_i}{\lambda_i} \left(1 - e^{-\lambda_i t}\right),$$

where λ_i are eigenvalues of B, and $\bar{g} = P^T g$.

Part 3. Search over a model-based steepest descent path

Joint work with: Serge Gratton

$$q(s) = g^T s + \frac{1}{2} s^T B s$$

The steepest descent path $s(t)$:

$$\dot{s} = -\nabla q(s(t)), \quad s(0) = 0.$$

Let P be eigenvectors of B. New variables: $v = P^T s$.

The components of $v(t) = P^T s(t)$ are calculated as

$$v_i(t) = -\frac{\bar{g}_i}{\lambda_i} \left(1 - e^{-\lambda_i t}\right),$$

where λ_i are eigenvalues of B, and $\bar{g} = P^T g$.

Trial points produced by the steepest descant path

- A curvilinear search along the path
- An approximate solution to the TR subproblem
- An approximate solutions to the model with cubic regularization

Trial points produced by the steepest descant path

- A curvilinear search along the path
- An approximate solution to the TR subproblem
- An approximate solutions to the model with cubic regularization

Trial points produced by the steepest descant path

- A curvilinear search along the path
- An approximate solution to the TR subproblem
- An approximate solutions to the model with cubic regularization

Part 4.
Dense initialization for limited memory algorithms

Johannes Brust, Oleg Burdakov, Jennifer B. Erway, and Roummel F. Marcia.
Dense initializations for limited-memory quasi-Newton methods.
*arXiv:*1710.02396 [math.OC]

Dense initialization

Diagonal initialization:

$$B_0 = \gamma_k I = \gamma_k P P^T = \gamma_k P_{\|} P_{\|}^T + \gamma_k P_{\perp} P_{\perp}^T.$$

Dense initialization:

$$\widehat{B}_0 = \gamma_k P_{\|} P_{\|}^T + \gamma_k^{\perp} P_{\perp} P_{\perp}^T.$$

Dense initialization

Diagonal initialization:

$$B_0 = \gamma_k I = \gamma_k P P^T = \gamma_k P_{\|} P_{\|}^T + \gamma_k P_{\perp} P_{\perp}^T.$$

Dense initialization:

$$\widehat{B}_0 = \gamma_k P_{\|} P_{\|}^T + \gamma_k^{\perp} P_{\perp} P_{\perp}^T.$$

Eigendecomposition

$$\widehat{B}_k = \widehat{B}_0 - \left[\widehat{B}_0 S_k \;\; Y_k\right] \begin{bmatrix} S_k^T \widehat{B}_0 S_k & L_k \\ L_k^T & -D_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^T \widehat{B}_0 \\ Y_k^T \end{bmatrix}$$

- $Bu = \gamma_k^{\perp} u, \;\; \forall u \in P_{\perp} \;\; \Longrightarrow$
  $n - m$ eigenvalues $\gamma^{\perp}$ with eigenspace $P_{\perp}$
- $Bu_i = (\gamma_k + d_i)u_i, \;\; \forall u_i$, the i-th column of $P_{\|} \;\; \Longrightarrow$
  m eigenvalues $\lambda_i = \gamma + d_i$, where d_i is the i-th eigenvalue of RWR^T

Eigendecomposition

$$\widehat{B}_k = \widehat{B}_0 - \left[\widehat{B}_0 S_k \;\; Y_k\right] \begin{bmatrix} S_k^T \widehat{B}_0 S_k & L_k \\ L_k^T & -D_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^T \widehat{B}_0 \\ Y_k^T \end{bmatrix}$$

- $Bu = \gamma_k^{\perp} u, \;\; \forall u \in P_{\perp} \quad \Longrightarrow$
  $n - m$ eigenvalues $\gamma^{\perp}$ with eigenspace $P_{\perp}$
- $Bu_i = (\gamma_k + d_i)u_i, \;\; \forall u_i$, the i-th column of $P_{\parallel} \quad \Longrightarrow$
  m eigenvalues $\lambda_i = \gamma + d_i$, where d_i is the i-th eigenvalue of RWR^T

Eigendecomposition

$$\widehat{B}_k = \widehat{B}_0 - \left[\widehat{B}_0 S_k \;\; Y_k\right] \begin{bmatrix} S_k^T \widehat{B}_0 S_k & L_k \\ L_k^T & -D_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^T \widehat{B}_0 \\ Y_k^T \end{bmatrix}$$

- $Bu = \gamma_k^{\perp} u, \;\; \forall u \in P_{\perp} \;\; \Longrightarrow$
  $n - m$ eigenvalues $\gamma^{\perp}$ with eigenspace $P_{\perp}$
- $Bu_i = (\gamma_k + d_i)u_i, \;\; \forall u_i$, the i-th column of $P_{\|} \;\; \Longrightarrow$
  m eigenvalues $\lambda_i = \gamma + d_i$, where d_i is the i-th eigenvalue of RWR^T

Convergence analysis

The dense initialization $\widehat{B}_0$ allows for retaining global convergence property, provided that $\gamma_k^{\perp}$ is uniformly bonded above.

Some possible choices of $\gamma_k^{\perp}$

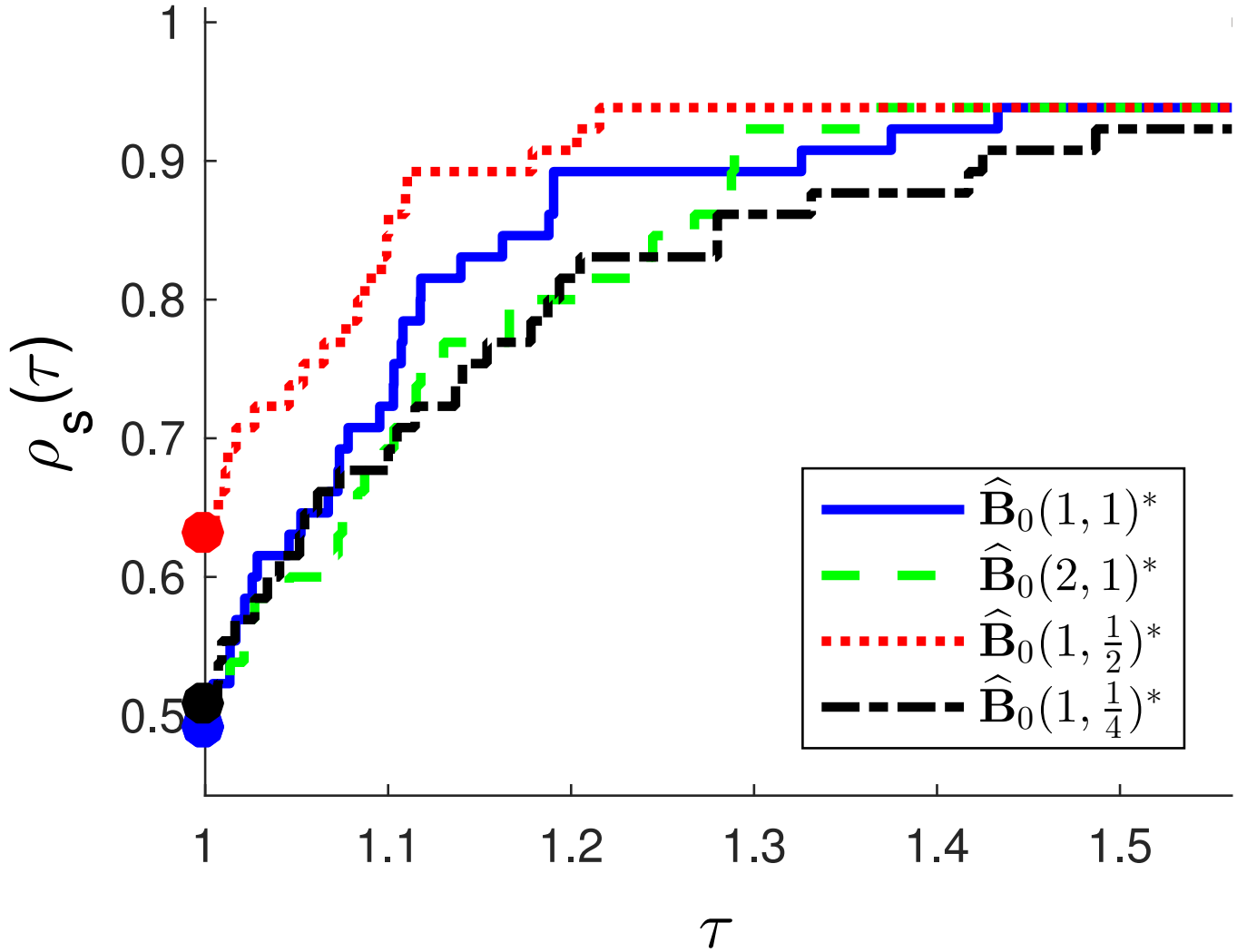
Parametrized family of choices:

$$\gamma_k^{\perp}(c, \lambda) = \lambda c \gamma_k^{\max} + (1 - \lambda)\gamma_k,$$

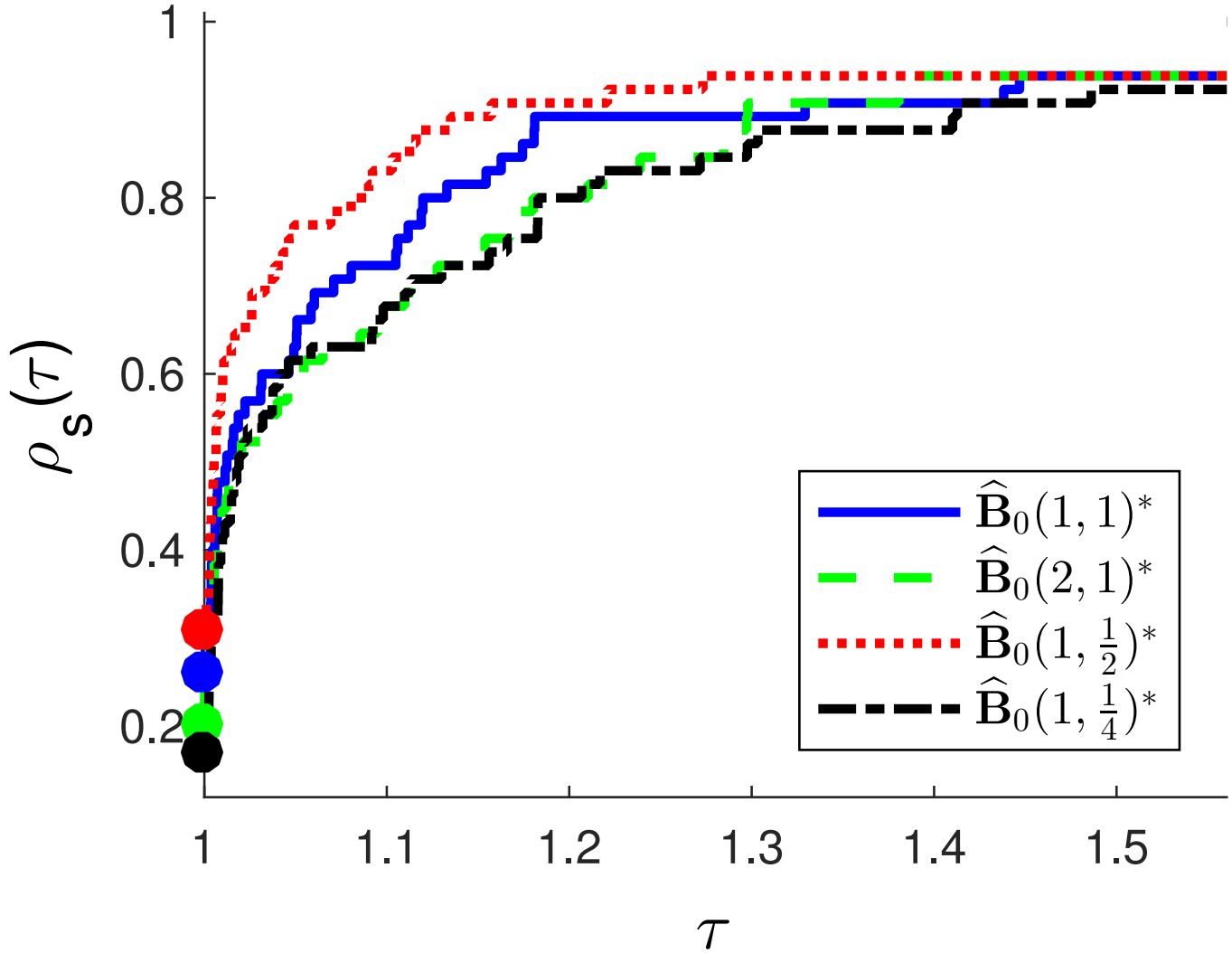
where

$$\gamma_k^{\max} = \max_{1 \le i \le k} \gamma_i = \max_{1 \le i \le k} \frac{y_i^T y_i}{s_i^T y_i}$$

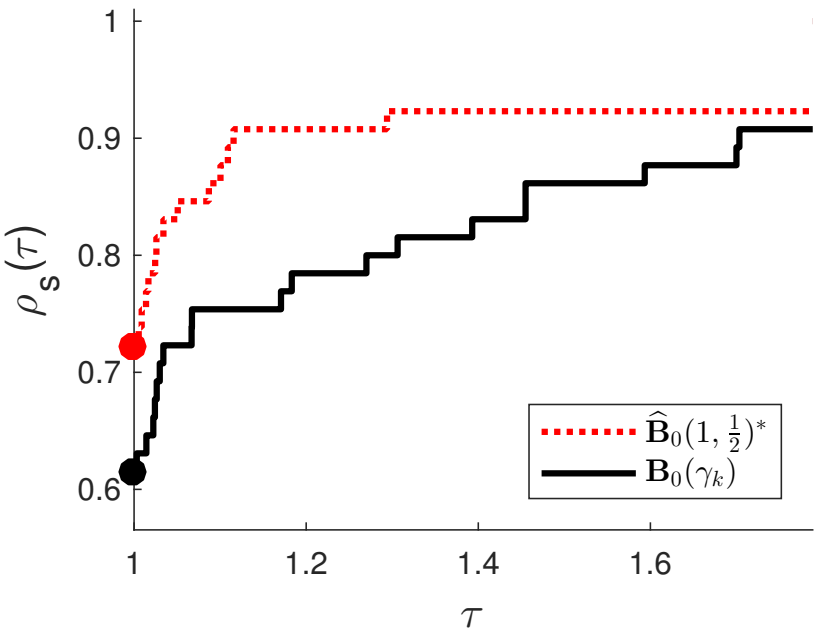
Numerical experiments for $\gamma_k^{\perp}(c, \lambda)$ (iterations)

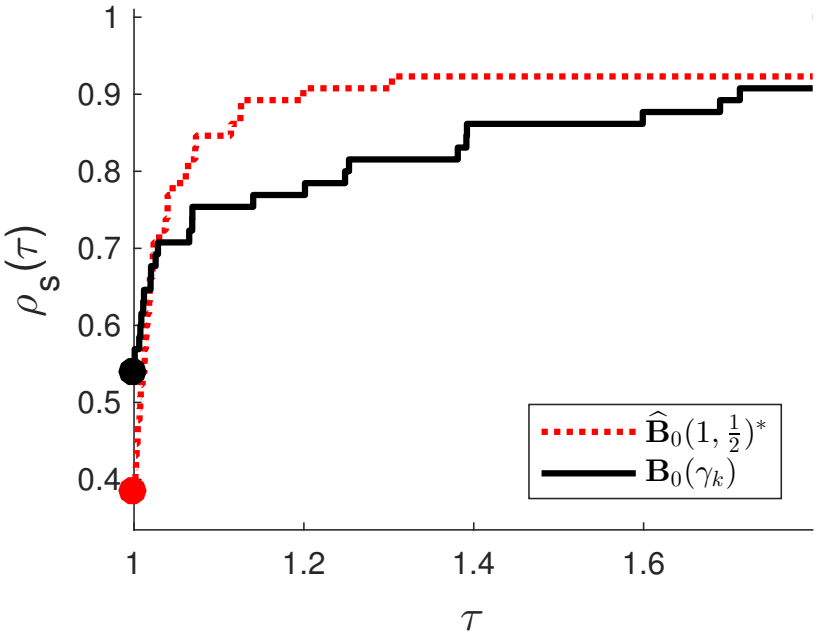


Numerical experiments for $\gamma_k^{\perp}(c, \lambda)$ (CPU time)



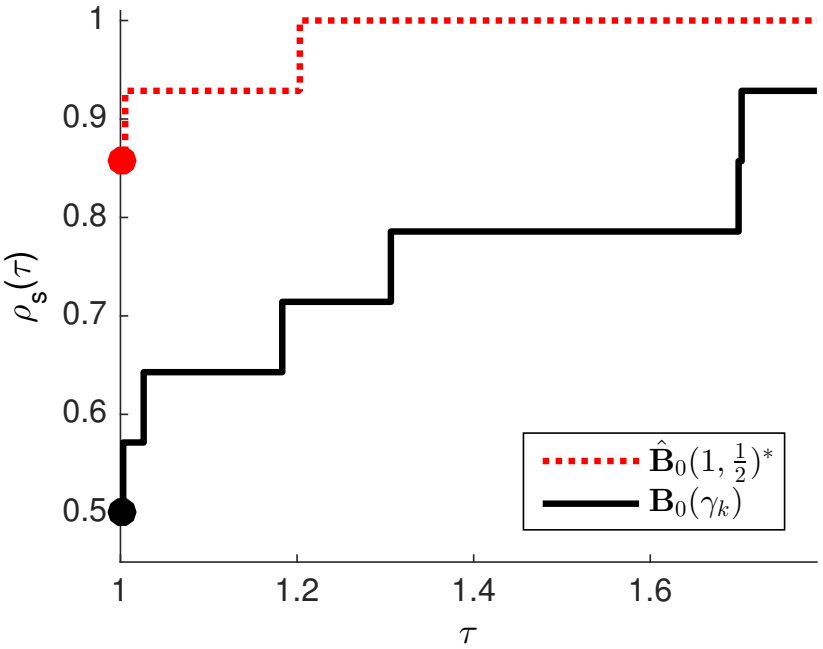
Numerical experiments: dense vs. diagonal initialization

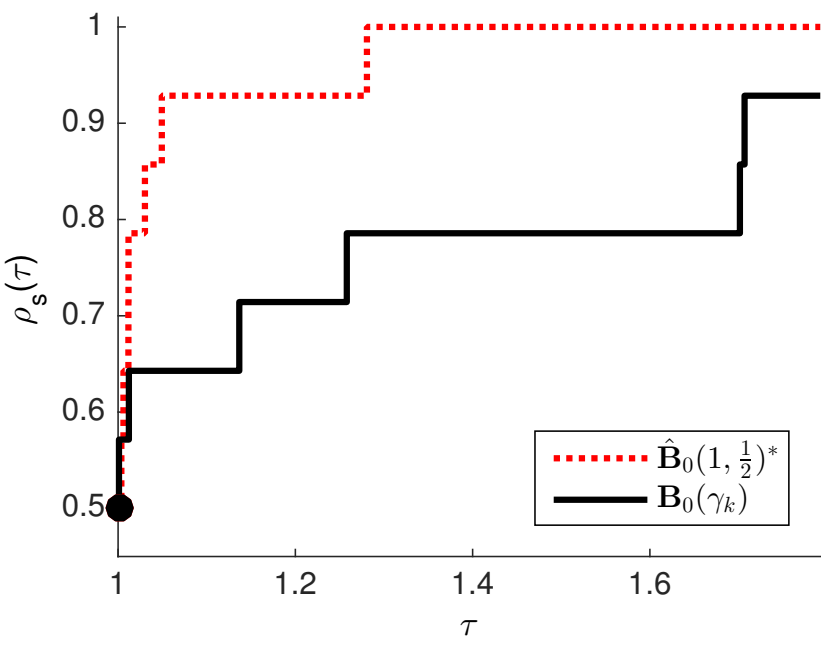




The dense initialization with $\gamma_k^{\perp}(1, \frac{1}{2})$ vs. the conventional initialization; number of iterations (left) and CPU time (right)

Dense vs. diagonal initialization (cont.)





The dense initialization with $\gamma_k^{\perp}(1, \frac{1}{2})$ vs. the conventional initializationon the subset of 14 problems in which the unconstrained minimizer is rejected at over 30% of the iterations; number of iterations (left) and CPU time (right)

Part 5. Future plans

- SR1 quasi-Newton updating formula
- Multipoint symmetric secant updates