On solving saddle-point problems and non-linear monotone equations

Oleg Burdakov

Linköping University, Sweden

Joint work with:
Yu-Hong Dai and Na Huang

Table: Numerical results for test set 1.

n		MINRES	SCG	SWI(2)	SWI(5)	SWI(8)
768	Iter	563	191	237	221	219
	CPU	0.0254	0.0697	0.0161	0.0205	0.0327
3 072	Iter	2 001	378	429	427	384
	CPU	0.3808	0.8316	0.0838	0.1403	0.1486
12 288	Iter	7 367	735	827	818	741
	CPU	4.4968	11.0272	0.5950	0.9618	1.2020
27 648	Iter	16 088	1 091	1 217	1 211	1 096
	CPU	21.0000	49.7991	1.8228	3.1293	4.8511
49 152	Iter	27 974	1 435	1 609	1 601	1 462
	CPU	64.1672	150.0516	4.4440	10.0289	13.8535

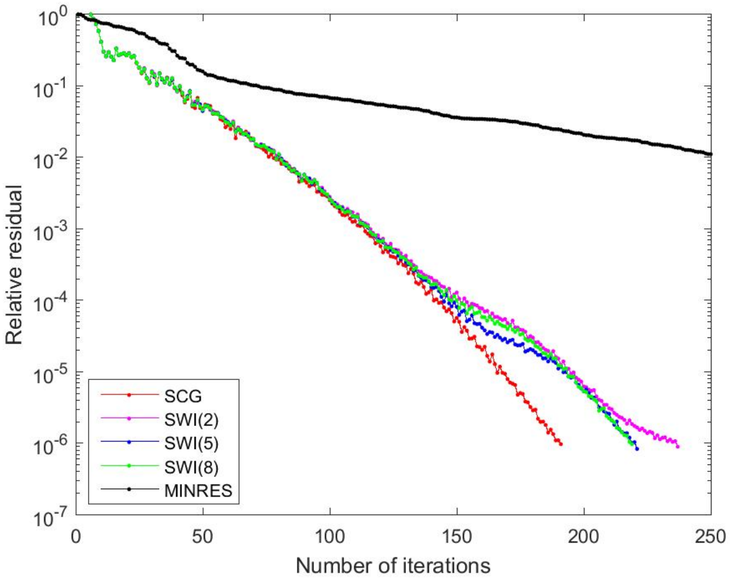

Figure: Relative residual vs. k for test set 1 ($n = 3072$).

Saddle-point problem

Given a function $f : R^{n_x \times n_y} \to R^1$, find a saddle point $z^* = [x^*, y^*] \in R^n$, where $x^* \in R^{n_x}$, $y^* \in R^{n_y}$ and $n = n_x + n_y$, such that

$$f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*), \quad \forall x \in R^{n_x}, y \in R^{n_y}.$$

Assumption.

Function $f(x, y)$ is strongly convex in x and strongly concave in y.

$\Rightarrow$ There exists a saddle point z^*, and it is unique.

Saddle-point problem

Given a function $f : R^{n_x \times n_y} \to R^1$, find a saddle point $z^* = [x^*, y^*] \in R^n$, where $x^* \in R^{n_x}$, $y^* \in R^{n_y}$ and $n = n_x + n_y$, such that

$$f(x^*, y) \le f(x^*, y^*) \le f(x, y^*), \quad \forall x \in R^{n_x}, y \in R^{n_y}.$$

Assumption.

Function $f(x, y)$ is strongly convex in x and strongly concave in y.

$\Rightarrow$ There exists a saddle point z^*, and it is unique.

Saddle-point problem

Given a function $f : R^{n_x \times n_y} \rightarrow R^1$, find a saddle point $z^* = [x^*, y^*] \in R^n$, where $x^* \in R^{n_x}$, $y^* \in R^{n_y}$ and $n = n_x + n_y$, such that

$$f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*), \quad \forall x \in R^{n_x}, y \in R^{n_y}.$$

Assumption.

Function $f(x, y)$ is strongly convex in x and strongly concave in y.

$\Rightarrow$ There exists a saddle point z^*, and it is unique.

Relation to unconstrained minimization

Observation.

When $n_y = 0$, the variable y vanishes in f, and then the saddle point problem is reduced to minimizing $f(x)$ in $x \in R^{n_x}$.

Aim.

Develop saddle-point search algorithms which, in the case of $n_y = 0$, would reduce to known unconstrained minimization algorithms.

Publications.

Burdakov O.P. Conjugate direction methods for solving systems of equations and finding saddle points. I
Engineering Cybernetics (1982) **20**, No. 3, pp. 13–19.

Burdakov O.P. Conjugate direction methods for solving systems of equations and finding saddle points. II
Engineering Cybernetics (1982) **20**, No. 4, pp. 23–32.

Relation to unconstrained minimization

Observation.

When $n_y = 0$, the variable y vanishes in f, and then the saddle point problem is reduced to minimizing $f(x)$ in $x \in R^{n_x}$.

Aim.

Develop saddle-point search algorithms which, in the case of $n_y = 0$, would reduce to known unconstrained minimization algorithms.

Publications.

Burdakov O.P. Conjugate direction methods for solving systems of equations and finding saddle points. I
Engineering Cybernetics (1982) **20**, No. 3, pp. 13–19.

Burdakov O.P. Conjugate direction methods for solving systems of equations and finding saddle points. II
Engineering Cybernetics (1982) **20**, No. 4, pp. 23–32.

Relation to unconstrained minimization

Observation.

When $n_y = 0$, the variable y vanishes in f, and then the saddle point problem is reduced to minimizing $f(x)$ in $x \in R^{n_x}$.

Aim.

Develop saddle-point search algorithms which, in the case of $n_y = 0$, would reduce to known unconstrained minimization algorithms.

Publications.

Burdakov O.P. Conjugate direction methods for solving systems of equations and finding saddle points. I
Engineering Cybernetics (1982) **20**, No. 3, pp. 13–19.

Burdakov O.P. Conjugate direction methods for solving systems of equations and finding saddle points. II
Engineering Cybernetics (1982) **20**, No. 4, pp. 23–32.

Monotone equations

Let $f(z)$ be sufficiently smooth. Denote

$$F(z) = E\nabla f(z),$$

where

$$E = \begin{bmatrix} I_{n_x} & 0 \\ 0 & -I_{n_y} \end{bmatrix}.$$

$$Ef''(z) = \begin{bmatrix} f''_{xx}(z) & f''_{xy}(z) \\ -f''_{yx}(z) & -f''_{yy}(z) \end{bmatrix}.$$

The saddle point problem is equivalent to solving the system of nonlinear monotone equations

$$F(z) = 0.$$

Monotone equations

Let $f(z)$ be sufficiently smooth. Denote

$$F(z) = E\nabla f(z),$$

where

$$E = \begin{bmatrix} I_{n_x} & 0 \\ 0 & -I_{n_y} \end{bmatrix}.$$

$$Ef''(z) = \begin{bmatrix} f''_{xx}(z) & f''_{xy}(z) \\ -f''_{yx}(z) & -f''_{yy}(z) \end{bmatrix}.$$

The saddle point problem is equivalent to solving the system of nonlinear monotone equations

$$F(z) = 0.$$

Monotone equations

Let $f(z)$ be sufficiently smooth. Denote

$$F(z) = E\nabla f(z),$$

where

$$E = \begin{bmatrix} I_{n_x} & 0 \\ 0 & -I_{n_y} \end{bmatrix}.$$

$$Ef''(z) = \begin{bmatrix} f''_{xx}(z) & f''_{xy}(z) \\ -f''_{yx}(z) & -f''_{yy}(z) \end{bmatrix}.$$

The saddle point problem is equivalent to solving the system of nonlinear monotone equations

$$F(z) = 0.$$

Properties of $F(z)$

$f(z)$ is strongly convex-concave

$$\Downarrow$$

There exists a scalar $c > 0$ such that, for all $z \in R^n$,

$$\langle f''_{xx}(z)p_x, p_x\rangle \geq c\|p_x\|^2, \quad \forall p_x \in R^{n_x},$$
$$\langle f''_{yy}(z)p_y, p_y\rangle \leq -c\|p_y\|^2, \quad \forall p_y \in R^{n_y}.$$

$$\Downarrow$$

$$\begin{aligned}\langle Ef''(z)p, p\rangle &= \langle f''_{xx}(z)p_x, p_x\rangle - \langle f''_{yy}(z)p_y, p_y\rangle \\ &\geq c\|p\|^2, \quad \forall p = [p_x, p_y] \in R^n,\end{aligned}$$

i.e. the matrix $Ef''(z) = F'(z)$ is positively definite.

$$\Downarrow$$

The mapping F is strongly monotone

$$\langle F(u) - F(v), u - v\rangle \geq c\|u - v\|^2, \quad \forall u, v \in R^n.$$

Properties of $F(z)$

$f(z)$ is strongly convex-concave

$$\Downarrow$$

There exists a scalar $c > 0$ such that, for all $z \in R^n$,

$$\langle f''_{xx}(z)p_x, p_x\rangle \geq c\|p_x\|^2, \quad \forall p_x \in R^{n_x},$$
$$\langle f''_{yy}(z)p_y, p_y\rangle \leq -c\|p_y\|^2, \quad \forall p_y \in R^{n_y}.$$

$$\Downarrow$$

$$\begin{aligned} \langle Ef''(z)p, p\rangle &= \langle f''_{xx}(z)p_x, p_x\rangle - \langle f''_{yy}(z)p_y, p_y\rangle \\ &\geq c\|p\|^2, \quad \forall p = [p_x, p_y] \in R^n, \end{aligned}$$

i.e. the matrix $Ef''(z) = F'(z)$ is positively definite.

$$\Downarrow$$

The mapping F is strongly monotone

$$\langle F(u) - F(v), u - v\rangle \geq c\|u - v\|^2, \quad \forall u, v \in R^n.$$

Properties of $F(z)$

$f(z)$ is strongly convex-concave

$$\Downarrow$$

There exists a scalar $c > 0$ such that, for all $z \in R^n$,

$$\langle f''_{xx}(z)p_x, p_x \rangle \geq c\|p_x\|^2, \quad \forall p_x \in R^{n_x},$$
$$\langle f''_{yy}(z)p_y, p_y \rangle \leq -c\|p_y\|^2, \quad \forall p_y \in R^{n_y}.$$

$$\Downarrow$$

$$\begin{aligned} \langle Ef''(z)p, p \rangle &= \langle f''_{xx}(z)p_x, p_x \rangle - \langle f''_{yy}(z)p_y, p_y \rangle \\ &\geq c\|p\|^2, \quad \forall p = [p_x, p_y] \in R^n, \end{aligned}$$

i.e. the matrix $Ef''(z) = F'(z)$ is positively definite.

$$\Downarrow$$

The mapping F is strongly monotone

$$\langle F(u) - F(v), u - v \rangle \geq c\|u - v\|^2, \quad \forall u, v \in R^n.$$

Line search for saddle point problem

$$z_{k+1} = z_k + \alpha_k p_k$$

Orthogonality-based line search:

$$\langle E\nabla f(z_k + \alpha p_k), p_k \rangle = 0.$$

- Since $f(x, y)$ is strongly convex-concave, the solution α_k to this equation exists and unique for any nonzero p_k.
- When $n_y = 0$, the line search reduces to minimization of $f(x)$ along p_k.

Line search for saddle point problem

$$z_{k+1} = z_k + \alpha_k p_k$$

Orthogonality-based line search:

$$\langle E \nabla f(z_k + \alpha p_k), p_k \rangle = 0.$$

- Since $f(x, y)$ is strongly convex-concave, the solution α_k to this equation exists and unique for any nonzero p_k.
- When $n_y = 0$, the line search reduces to minimization of $f(x)$ along p_k.

Line search for saddle point problem

$$z_{k+1} = z_k + \alpha_k p_k$$

Orthogonality-based line search:

$$\langle E \nabla f(z_k + \alpha p_k), p_k \rangle = 0.$$

- Since $f(x, y)$ is strongly convex-concave, the solution α_k to this equation exists and unique for any nonzero p_k.
- When $n_y = 0$, the line search reduces to minimization of $f(x)$ along p_k.

Line search for saddle point problem

$$z_{k+1} = z_k + \alpha_k p_k$$

Orthogonality-based line search:

$$\langle E \nabla f(z_k + \alpha p_k), p_k \rangle = 0.$$

- Since $f(x, y)$ is strongly convex-concave, the solution α_k to this equation exists and unique for any nonzero p_k.
- When $n_y = 0$, the line search reduces to minimization of $f(x)$ along p_k.

A trade-off provided by the line search

Partitioning:

$p_k = [p_x, p_y] \quad \text{and} \quad \nabla_z f(z_{k+1}) = [\nabla_x f, \nabla_y f]$

Assumption:

$\langle \nabla_x f, p_x \rangle \neq 0 \quad (\Rightarrow \langle \nabla_y f, p_y \rangle \neq 0, \text{because } E\nabla f(z_{k+1}) \perp p_k)$

Given a sufficiently small $\varepsilon > 0$, consider

$f_x^* = \min_{t\in[-\varepsilon,\varepsilon]} f(x_{k+1} + tp_x, y_{k+1}), \quad f_y^* = \max_{t\in[-\varepsilon,\varepsilon]} f(x_{k+1}, y_{k+1} + tp_y)$

$\Downarrow$

$t_x^* = \pm\varepsilon, \;\; t_y^* = -t_x^*$

$f_x^* = f(z_{k+1}) - \varepsilon|\langle p_x, \nabla_x f\rangle| + o(\varepsilon^2), \quad f_y^* = f(z_{k+1}) + \varepsilon|\langle p_y, \nabla_y f\rangle| + o(\varepsilon^2)$

Thus, the gain in minimizing $f(x, y_{k+1})$ along p_x is equal to the gain in maximizing $f(x_{k+1}, y)$ along p_y to the first-order approximation. This means that the orthogonality-based line search provides in the resulting point z_{k+1} a kind of 'equal opportunities' for a local minimization over p_x and a local maximization over p_y.

A trade-off provided by the line search

Partitioning:

$p_k = [p_x, p_y] \quad \text{and} \quad \nabla_z f(z_{k+1}) = [\nabla_x f, \nabla_y f]$

Assumption:

$\langle \nabla_x f, p_x \rangle \neq 0 \quad (\Rightarrow \langle \nabla_y f, p_y \rangle \neq 0, \text{because } E\nabla f(z_{k+1}) \perp p_k)$

Given a sufficiently small $\varepsilon > 0$, consider

$f_x^* = \min_{t\in[-\varepsilon,\varepsilon]} f(x_{k+1} + tp_x, y_{k+1}), \quad f_y^* = \max_{t\in[-\varepsilon,\varepsilon]} f(x_{k+1}, y_{k+1} + tp_y)$

$\Downarrow$

$t_x^* = \pm\varepsilon, \;\; t_y^* = -t_x^*$

$f_x^* = f(z_{k+1}) - \varepsilon|\langle p_x, \nabla_x f\rangle| + o(\varepsilon^2), \quad f_y^* = f(z_{k+1}) + \varepsilon|\langle p_y, \nabla_y f\rangle| + o(\varepsilon^2)$

Thus, the gain in minimizing $f(x, y_{k+1})$ along p_x is equal to the gain in maximizing $f(x_{k+1}, y)$ along p_y to the first-order approximation. This means that the orthogonality-based line search provides in the resulting point z_{k+1} a kind of 'equal opportunities' for a local minimization over p_x and a local maximization over p_y.

A trade-off provided by the line search

Partitioning:

$p_k = [p_x, p_y] \quad \text{and} \quad \nabla_z f(z_{k+1}) = [\nabla_x f, \nabla_y f]$

Assumption:

$\langle \nabla_x f, p_x \rangle \neq 0 \quad (\Rightarrow \langle \nabla_y f, p_y \rangle \neq 0, \text{because } E\nabla f(z_{k+1}) \perp p_k)$

Given a sufficiently small $\varepsilon > 0$, consider

$f_x^* = \min_{t \in [-\varepsilon, \varepsilon]} f(x_{k+1} + tp_x, y_{k+1}), \quad f_y^* = \max_{t \in [-\varepsilon, \varepsilon]} f(x_{k+1}, y_{k+1} + tp_y)$

$\Downarrow$

$t_x^* = \pm\varepsilon, \quad t_y^* = -t_x^*$

$f_x^* = f(z_{k+1}) - \varepsilon |\langle p_x, \nabla_x f \rangle| + o(\varepsilon^2), \quad f_y^* = f(z_{k+1}) + \varepsilon |\langle p_y, \nabla_y f \rangle| + o(\varepsilon^2)$

Thus, the gain in minimizing $f(x, y_{k+1})$ along p_x is equal to the gain in maximizing $f(x_{k+1}, y)$ along p_y to the first-order approximation. This means that the orthogonality-based line search provides in the resulting point z_{k+1} a kind of 'equal opportunities' for a local minimization over p_x and a local maximization over p_y.

A trade-off provided by the line search

Partitioning:

$p_k = [p_x, p_y] \quad \text{and} \quad \nabla_z f(z_{k+1}) = [\nabla_x f, \nabla_y f]$

Assumption:

$\langle \nabla_x f, p_x \rangle \neq 0 \quad (\Rightarrow \langle \nabla_y f, p_y \rangle \neq 0, \text{because } E\nabla f(z_{k+1}) \perp p_k)$

Given a sufficiently small $\varepsilon > 0$, consider

$f_x^* = \min_{t \in [-\varepsilon, \varepsilon]} f(x_{k+1} + tp_x, y_{k+1}), \quad f_y^* = \max_{t \in [-\varepsilon, \varepsilon]} f(x_{k+1}, y_{k+1} + tp_y)$

$\Downarrow$

$t_x^* = \pm\varepsilon, \;\; t_y^* = -t_x^*$

$f_x^* = f(z_{k+1}) - \varepsilon |\langle p_x, \nabla_x f \rangle| + o(\varepsilon^2), \quad f_y^* = f(z_{k+1}) + \varepsilon |\langle p_y, \nabla_y f \rangle| + o(\varepsilon^2)$

Thus, the gain in minimizing $f(x, y_{k+1})$ along p_x is equal to the gain in maximizing $f(x_{k+1}, y)$ along p_y to the first-order approximation. This means that the orthogonality-based line search provides in the resulting point z_{k+1} a kind of 'equal opportunities' for a local minimization over p_x and a local maximization over p_y.

A trade-off provided by the line search

Partitioning:

$p_k = [p_x, p_y] \quad \text{and} \quad \nabla_z f(z_{k+1}) = [\nabla_x f, \nabla_y f]$

Assumption:

$\langle \nabla_x f, p_x \rangle \neq 0 \quad (\Rightarrow \langle \nabla_y f, p_y \rangle \neq 0, \text{because } E\nabla f(z_{k+1}) \perp p_k)$

Given a sufficiently small $\varepsilon > 0$, consider

$f_x^* = \min_{t \in [-\varepsilon, \varepsilon]} f(x_{k+1} + tp_x, y_{k+1}), \quad f_y^* = \max_{t \in [-\varepsilon, \varepsilon]} f(x_{k+1}, y_{k+1} + tp_y)$

$\Downarrow$

$t_x^* = \pm\varepsilon, \;\; t_y^* = -t_x^*$

$f_x^* = f(z_{k+1}) - \varepsilon |\langle p_x, \nabla_x f \rangle| + o(\varepsilon^2), \quad f_y^* = f(z_{k+1}) + \varepsilon |\langle p_y, \nabla_y f \rangle| + o(\varepsilon^2)$

Thus, the gain in minimizing $f(x, y_{k+1})$ along p_x is equal to the gain in maximizing $f(x_{k+1}, y)$ along p_y to the first-order approximation. This means that the orthogonality-based line search provides in the resulting point z_{k+1} a kind of 'equal opportunities' for a local minimization over p_x and a local maximization over p_y.

A trade-off provided by the line search

Partitioning:

$p_k = [p_x, p_y] \quad \text{and} \quad \nabla_z f(z_{k+1}) = [\nabla_x f, \nabla_y f]$

Assumption:

$\langle \nabla_x f, p_x \rangle \neq 0 \quad (\Rightarrow \langle \nabla_y f, p_y \rangle \neq 0, \text{because } E\nabla f(z_{k+1}) \perp p_k)$

Given a sufficiently small $\varepsilon > 0$, consider

$f_x^* = \min_{t \in [-\varepsilon, \varepsilon]} f(x_{k+1} + tp_x, y_{k+1}), \quad f_y^* = \max_{t \in [-\varepsilon, \varepsilon]} f(x_{k+1}, y_{k+1} + tp_y)$

$\Downarrow$

$t_x^* = \pm\varepsilon, \;\; t_y^* = -t_x^*$

$f_x^* = f(z_{k+1}) - \varepsilon |\langle p_x, \nabla_x f \rangle| + o(\varepsilon^2), \quad f_y^* = f(z_{k+1}) + \varepsilon |\langle p_y, \nabla_y f \rangle| + o(\varepsilon^2)$

Thus, the gain in minimizing $f(x, y_{k+1})$ along p_x is equal to the gain in maximizing $f(x_{k+1}, y)$ along p_y to the first-order approximation. This means that the orthogonality-based line search provides in the resulting point z_{k+1} a kind of 'equal opportunities' for a local minimization over p_x and a local maximization over p_y.

Newton's method

Newton's search direction: $p_k = -(f_k'')^{-1} \nabla f_k = -(F_k')^{-1} F_k$

Properties of the orthogonality-based line search:

- $\alpha_k \to 1, \quad k \to \infty$
- $z_k \to z^*$ superlinearly / quadratically

Newton's method

Newton's search direction: $p_k = -(f_k'')^{-1}\nabla f_k = -(F_k')^{-1}F_k$

Properties of the orthogonality-based line search:

- $\alpha_k \to 1, \quad k \to \infty$
- $z_k \to z^*$ superlinearly / quadratically

Newton's method

Newton's search direction: $p_k = -(f_k'')^{-1}\nabla f_k = -(F_k')^{-1}F_k$

Properties of the orthogonality-based line search:

- $\alpha_k \to 1, \quad k \to \infty$
- $z_k \to z^*$ superlinearly / quadratically

Conjugate direction methods for unconstrained optimization

Let f(x) be a strictly convex quadratic function in R^n with $f'' = A$. Given a system of conjugate directions $\{p_i\}_{i=0}^{n-1}$:

$$\langle Ap_i, p_j \rangle = 0, \quad \forall 0 \leq i, j \leq n-1, i \neq j.$$

Then, for any starting point x_0, the exact-line-search-based iterates

$$x_{k+1} = x_k + \alpha_k p_k$$

converges to x^* in at most n iterations, because

$$\langle \nabla f(x_{k+1}), p_i \rangle = 0, \quad \forall 0 \leq i \leq k$$

Q: How to build a sequence of conjugate directions?

Example: the conjugate gradient method

Conjugate direction methods for unconstrained optimization

Let f(x) be a strictly convex quadratic function in R^n with $f'' = A$. Given a system of conjugate directions $\{p_i\}_{i=0}^{n-1}$:

$$\langle Ap_i, p_j \rangle = 0, \quad \forall 0 \leq i, j \leq n-1, i \neq j.$$

Then, for any starting point x_0, the exact-line-search-based iterates

$$x_{k+1} = x_k + \alpha_k p_k$$

converges to x^* in at most n iterations, because

$$\langle \nabla f(x_{k+1}), p_i \rangle = 0, \quad \forall 0 \leq i \leq k$$

Q: How to build a sequence of conjugate directions?

Example: the conjugate gradient method

Conjugate direction methods for unconstrained optimization

Let f(x) be a strictly convex quadratic function in R^n with $f'' = A$. Given a system of conjugate directions $\{p_i\}_{i=0}^{n-1}$:

$$\langle Ap_i, p_j \rangle = 0, \quad \forall 0 \le i, j \le n-1, i \ne j.$$

Then, for any starting point x_0, the exact-line-search-based iterates

$$x_{k+1} = x_k + \alpha_k p_k$$

converges to x^* in at most n iterations, because

$$\langle \nabla f(x_{k+1}), p_i \rangle = 0, \quad \forall 0 \le i \le k$$

Q: How to build a sequence of conjugate directions?

Example: the conjugate gradient method

Derivative-free conjugate direction methods for unconstrained optimization

C.S. Smith (1962), M.J.D. Powell (1964):

- Given $a, b, p \in R^n$. Let x_a and x_b be the minimizers of $f(x)$ along p from a and b, respectively. Then

$$\langle A(x_b - x_a), p \rangle = 0$$

- Given $a, b \in R^n$ and a linear subspace $L \in R^m$. Let x_a and x_b be the minimizers of $f(x)$ in the linear manifolds $a + L$ and $b + L$, respectively. Then

$$\langle A(x_b - x_a), p \rangle = 0, \quad \forall p \in L$$

Derivative-free conjugate direction methods for unconstrained optimization

C.S. Smith (1962), M.J.D. Powell (1964):

- Given $a, b, p \in R^n$. Let x_a and x_b be the minimizers of $f(x)$ along p from a and b, respectively. Then

$$\langle A(x_b - x_a), p \rangle = 0$$

- Given $a, b \in R^n$ and a linear subspace $L \in R^m$. Let x_a and x_b be the minimizers of $f(x)$ in the linear manifolds $a + L$ and $b + L$, respectively. Then

$$\langle A(x_b - x_a), p \rangle = 0, \quad \forall p \in L$$

Saddle problem search case

OB (1980, 1982):
Let $f(x,y)$ be a strictly convex-concave quadratic function in R^n with $f'' = A$.

- Given $a, b, p \in R^n$. Let $x_a = a + \alpha_a p$ and $x_b = b + \alpha_b p$ be such that
$$\langle E\nabla f(x_a), p\rangle = 0 \text{ and } \langle E\nabla f(x_b), p\rangle = 0,$$
respectively. Then
$$\langle EA(x_b - x_a), p\rangle = 0$$

- Given $a, b \in R^n$ and a linear subspace $L \in R^m$. Let $x_a \in a + L$ and $x_b \in b + L$ be such that
$$\langle E\nabla f(x_a), p\rangle = 0 \text{ and } \langle E\nabla f(x_b), p\rangle = 0, \quad \forall p \in L,$$
respectively. Then
$$\langle EA(x_b - x_a), p\rangle = 0, \quad \forall p \in L$$

Saddle problem search case

OB (1980, 1982):
Let $f(x, y)$ be a strictly convex-concave quadratic function in R^n with $f'' = A$.

- Given $a, b, p \in R^n$. Let $x_a = a + \alpha_a p$ and $x_b = b + \alpha_b p$ be such that

  $$\langle E\nabla f(x_a), p\rangle = 0 \text{ and } \langle E\nabla f(x_b), p\rangle = 0,$$

  respectively. Then

  $$\langle EA(x_b - x_a), p\rangle = 0$$

- Given $a, b \in R^n$ and a linear subspace $L \in R^m$. Let $x_a \in a + L$ and $x_b \in b + L$ be such that

  $$\langle E\nabla f(x_a), p\rangle = 0 \text{ and } \langle E\nabla f(x_b), p\rangle = 0, \quad \forall p \in L,$$

  respectively. Then

  $$\langle EA(x_b - x_a), p\rangle = 0, \quad \forall p \in L$$

Semi-conjugate directions

G.W. Stewart (1973), V.V. Voevodin (1979), OB (1980, 1982):

Ordered vectors $p_0, p_1, \ldots, p_{n-1}$ in R^n are called semi-conjugate, if

$$\langle EAp_i, p_j \rangle = 0, \quad \forall 0 \le j < i \le n-1.$$

Semi-conjugate direction methods:

$$z_{k+1} = z_k + \alpha_k p_k,$$

where α_k is produced by the orthogonality-based line search.

Properties:

- $\langle E\nabla f(z_{k+1}), p_i \rangle = 0, \quad \forall 0 \le i \le k.$
- For any z_0, the sequence z_k converges to z^* in at most n iterations.
- When $n_y = 0$, the semi-conjugate direction methods reduce to the conjugate direction methods.

Semi-conjugate directions

G.W. Stewart (1973), V.V. Voevodin (1979), OB (1980, 1982):

Ordered vectors $p_0, p_1, \dots, p_{n-1}$ in R^n are called semi-conjugate, if

$$\langle EAp_i, p_j \rangle = 0, \quad \forall 0 \le j < i \le n-1.$$

Semi-conjugate direction methods:

$$z_{k+1} = z_k + \alpha_k p_k,$$

where α_k is produced by the orthogonality-based line search.

Properties:

- $\langle E\nabla f(z_{k+1}), p_i \rangle = 0, \quad \forall 0 \le i \le k.$
- For any z_0, the sequence z_k converges to z^* in at most n iterations.
- When $n_y = 0$, the semi-conjugate direction methods reduce to the conjugate direction methods.

Semi-conjugate directions

G.W. Stewart (1973), V.V. Voevodin (1979), OB (1980, 1982):

Ordered vectors $p_0, p_1, \dots, p_{n-1}$ in R^n are called semi-conjugate, if

$$\langle EAp_i, p_j \rangle = 0, \quad \forall 0 \le j < i \le n-1.$$

Semi-conjugate direction methods:

$$z_{k+1} = z_k + \alpha_k p_k,$$

where α_k is produced by the orthogonality-based line search.

Properties:

- $\langle E \nabla f(z_{k+1}), p_i \rangle = 0, \quad \forall 0 \le i \le k.$
- For any z_0, the sequence z_k converges to z^* in at most n iterations.
- When $n_y = 0$, the semi-conjugate direction methods reduce to the conjugate direction methods.

Semi-conjugate directions

G.W. Stewart (1973), V.V. Voevodin (1979), OB (1980, 1982):

Ordered vectors $p_0, p_1, \ldots, p_{n-1}$ in R^n are called semi-conjugate, if

$$\langle EAp_i, p_j \rangle = 0, \quad \forall 0 \leq j < i \leq n-1.$$

Semi-conjugate direction methods:

$$z_{k+1} = z_k + \alpha_k p_k,$$

where α_k is produced by the orthogonality-based line search.

Properties:

- $\langle E\nabla f(z_{k+1}), p_i \rangle = 0, \quad \forall 0 \leq i \leq k.$
- For any z_0, the sequence z_k converges to z^* in at most n iterations.
- When $n_y = 0$, the semi-conjugate direction methods reduce to the conjugate direction methods.

Semi-conjugate directions

G.W. Stewart (1973), V.V. Voevodin (1979), OB (1980, 1982):

Ordered vectors $p_0, p_1, \dots, p_{n-1}$ in R^n are called semi-conjugate, if

$$\langle EAp_i, p_j \rangle = 0, \quad \forall 0 \le j < i \le n-1.$$

Semi-conjugate direction methods:

$$z_{k+1} = z_k + \alpha_k p_k,$$

where α_k is produced by the orthogonality-based line search.

Properties:

- $\langle E\nabla f(z_{k+1}), p_i \rangle = 0, \quad \forall 0 \le i \le k.$
- For any z_0, the sequence z_k converges to z^* in at most n iterations.
- When $n_y = 0$, the semi-conjugate direction methods reduce to the conjugate direction methods.

Semi-conjugate direction methods: Non-quadratic case

Local quadratic rate of convergence $z_k \to z^*$, OB (1982).

Sketch of the proof

1. If the search directions are uniformly linearly independent, then $z_k \to z^*$ quadratically.
2. If, on the contrary, the convergence is not quadratic, then the search directions must be uniformly linearly independent, which implies that $z_k \to z^*$ quadratically.

Semi-conjugate direction methods: Non-quadratic case

Local quadratic rate of convergence $z_k \to z^*$, OB (1982).

Sketch of the proof

1. If the search directions are uniformly linearly independent, then $z_k \to z^*$ quadratically.
2. If, on the contrary, the convergence is not quadratic, then the search directions must be uniformly linearly independent, which implies that $z_k \to z^*$ quadratically.

Semi-conjugate direction methods: Non-quadratic case

Local quadratic rate of convergence $z_k \to z^*$, OB (1982).

Sketch of the proof

1. If the search directions are uniformly linearly independent, then $z_k \to z^*$ quadratically.
2. If, on the contrary, the convergence is not quadratic, then the search directions must be uniformly linearly independent, which implies that $z_k \to z^*$ quadratically.

Numerical experiments

Saddle point problem for the quadratic function

$$f(x, y) = \frac{1}{2} z^T \mathcal{A} z + \ell^T z,$$

where

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \in R^{(n_x+n_y)\times(n_x+n_y)},$$

$A, C \succ 0$.

SCG - semi-conjugate gradient algorithm.
SWI - limited memory (sliding window) version of SCG.

Stopping criteria:

$$\frac{\|\nabla f(z_k)\|_2}{\|\ell\|_2} \le 10^{-6}.$$

Numerical experiments

Saddle point problem for the quadratic function

$$f(x,y) = \frac{1}{2} z^T \mathcal{A} z + \ell^T z,$$

where

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \in R^{(n_x+n_y)\times(n_x+n_y)},$$

$A, C \succ 0$.

SCG - semi-conjugate gradient algorithm.
SWI - limited memory (sliding window) version of SCG.

Stopping criteria:

$$\frac{\|\nabla f(z_k)\|_2}{\|\ell\|_2} \leq 10^{-6}.$$

Numerical experiments

Saddle point problem for the quadratic function

$$f(x, y) = \frac{1}{2} z^T \mathcal{A} z + \ell^T z,$$

where

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \in R^{(n_x+n_y)\times(n_x+n_y)},$$

$A, C \succ 0$.

SCG - semi-conjugate gradient algorithm.
SWI - limited memory (sliding window) version of SCG.

Stopping criteria:

$$\frac{\|\nabla f(z_k)\|_2}{\|\ell\|_2} \leq 10^{-6}.$$

Test set 1 (Navier-Stokes equation)

$\ell = (1, \cdots, 1)^T$ and the matrices A, B and C are defined as follows:

$$A = \begin{pmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{pmatrix} \in R^{2p^2 \times 2p^2},$$

$$B = \begin{pmatrix} I \otimes F \\ F \otimes I \end{pmatrix} \in R^{2p^2 \times p^2}, \qquad C = \mathrm{diag}(1, 2, \cdots, p^2) \in R^{p^2 \times p^2}.$$

Here

$$T = \frac{1}{h^2} \cdot \mathrm{tridiag}(-1, 2, -1) \in R^{p \times p}, \qquad F = \frac{1}{h} \cdot \mathrm{tridiag}(-1, 1, 0) \in R^{p \times p},$$

with $\otimes$ being the Kronecker product symbol and $h = \frac{1}{p+1}$ the discretization meshsize.
The problem size is $n = 3p^2$, where $p = 16, 32, 64, 96\,128$ was considered.

Table: Numerical results for test set 1.

n		MINRES	SCG	SWI(2)	SWI(5)	SWI(8)
768	Iter	563	191	237	221	219
	CPU	0.0254	0.0697	0.0161	0.0205	0.0327
3 072	Iter	2 001	378	429	427	384
	CPU	0.3808	0.8316	0.0838	0.1403	0.1486
12 288	Iter	7 367	735	827	818	741
	CPU	4.4968	11.0272	0.5950	0.9618	1.2020
27 648	Iter	16 088	1 091	1 217	1 211	1 096
	CPU	21.0000	49.7991	1.8228	3.1293	4.8511
49 152	Iter	27 974	1 435	1 609	1 601	1 462
	CPU	64.1672	150.0516	4.4440	10.0289	13.8535

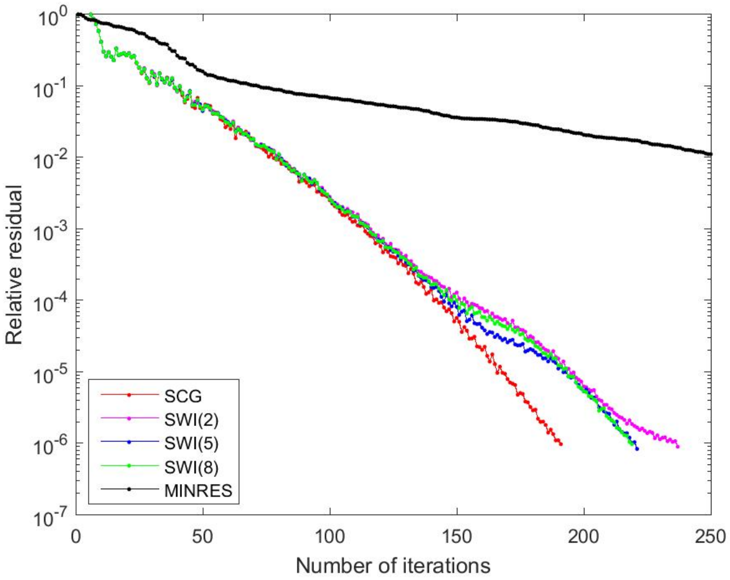

Figure: Relative residual vs. k for test set 1 ($n = 3072$).

Test set 2

$$A = \mathrm{diag}(1, 2, \cdots, n_x), \quad C = \mathrm{diag}(n_y, n_y - 1, \cdots, 1),$$
$$B = [I_{n_y}, \mathrm{rand}(n_y, n_x - n_y)], \quad n_y = 0.8n_x, \quad \ell = (1, \cdots, 1)^T$$

Table: Numerical results for test set 2.

n		MINRES	SCG	SWI(3)	SWI(6)	SWI(9)
1 800	Iter	2 680	182	235	216	194
	CPU	1.6227	0.2014	0.1336	0.1586	0.1515
3 600	Iter	5 466	255	335	304	269
	CPU	11.6519	0.8854	0.6879	0.6685	0.6147
7 200	Iter	11 049	358	482	427	377
	CPU	80.7257	3.7414	3.2795	3.0208	2.7506
14 400	Iter	22 238	501	696	595	531
	CPU	622.1941	18.3640	18.7429	16.3657	15.0092
28 800	Iter	44 647	702	1 016	774	800
	CPU	4976.7783	96.3985	111.2124	85.6512	90.5496

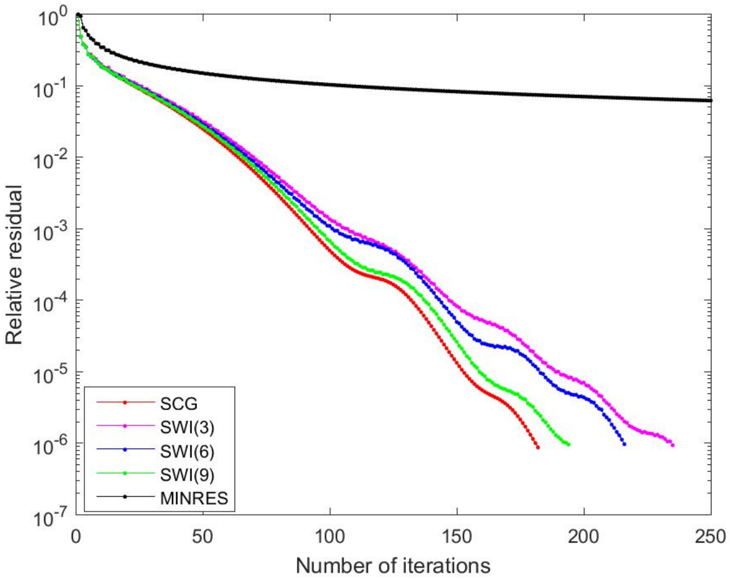

Figure: Relative residual vs. k for test set 2 ($n = 1800$).

Test set 3

$A = \hat{A}^T \hat{A} + \frac{1}{n_x} W_{n_x}$, $B = \mathrm{randn}(n_y, n_x)$, and $C = \hat{C}^T \hat{C} + W_{n_y}$, where $\hat{A} = \mathrm{randn}(n_x)$, $\hat{C} = \mathrm{randn}(n_y)$ and $W_r = \mathrm{diag}(1, 2, \cdots, r)$.
$n_y = 0.8 n_x$, $\ell = (1, 2, \ldots, r)^T$.

Table: Numerical results for test set 3.

n		MINRES	SCG	SWI(3)	SWI(6)	SWI(9)
3 600	Iter	4 072	504	550	530	548
	CPU	40.5615	6.7709	5.4802	5.4223	5.6892
7 200	Iter	5 753	696	750	747	725
	CPU	190.9926	28.2574	24.5692	24.1950	24.6226
10 800	Iter	7 157	864	912	896	907
	CPU	640.1808	90.8115	80.0332	78.7759	80.3283
14 400	Iter	8 209	997	1 034	1 027	1 031
	CPU	1259.3554	177.2851	157.7605	160.0064	161.6104
18 000	Iter	9 158	1 108	1 156	1 142	1 152
	CPU	2278.7554	301.9440	273.5980	272.0086	275.4546

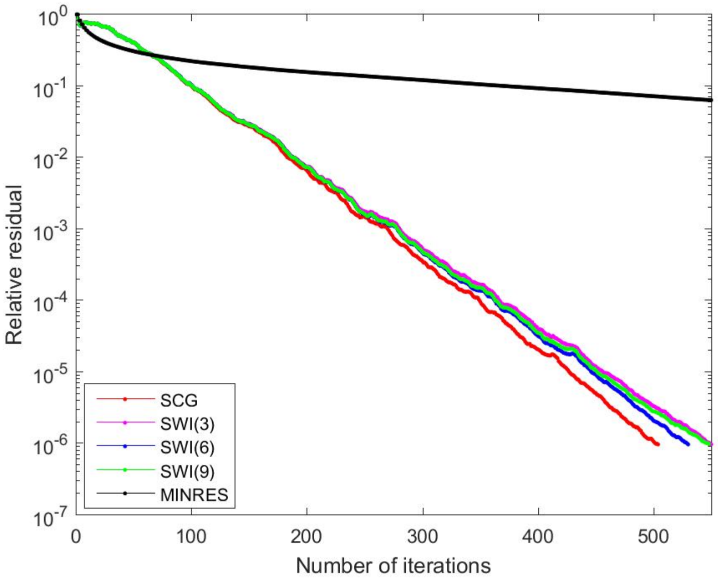

Figure: Relative residual vs. k for test set 3 ($n = 3600$).

Test set 4

System of linear monotone equations:

$$\mathcal{A}z = \ell$$

$\mathcal{A} = A^T A + c(B - B')$
$A = \mathrm{rand}(n)$
$B = \mathrm{rand}(n)$
$\ell = (1, 1, \cdots, 1)^T \in \mathbb{R}^n$
$c = 0.1, 1, 10$

Table: Numerical results for test set 4 with $c = 0.1$.

n		GMRES	SCG	SWI(40)	SWI(50)	SWI(60)
3000	Iter	1530	653	774	736	726
	CPU	20.9385	43.0067	10.2724	10.7132	10.8520
6000	Iter	2244	766	1118	931	876
	CPU	122.9305	141.5028	57.6757	49.0868	46.8783
9000	Iter	2814	884	1036	989	944
	CPU	347.1497	323.4416	129.6058	117.4073	114.0603
12000	Iter	3011	947	1126	1181	1022
	CPU	640.8991	556.3637	225.6072	248.1383	208.3967
15000	Iter	3592	1028	1171	1144	1131
	CPU	1273.3257	892.4290	387.7717	381.3977	379.2692

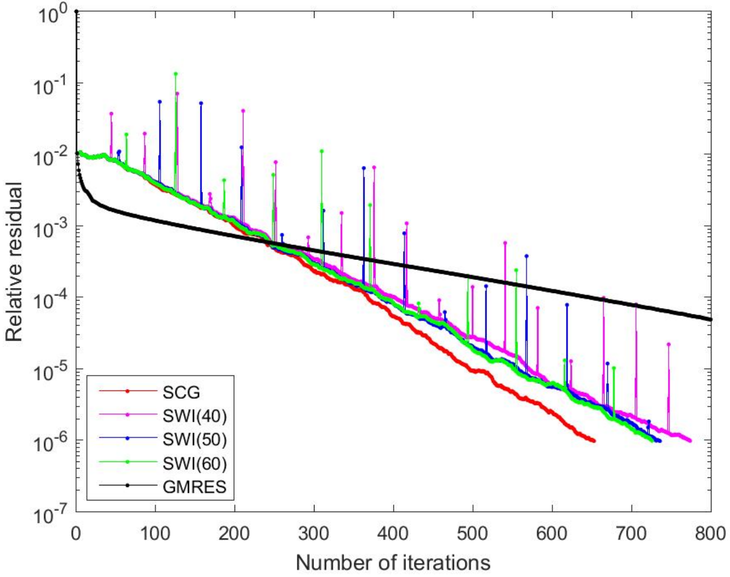

Figure: Relative residual vs. k for test set 4 ($n = 3000$, $c = 0.1$).

Table: Numerical results for test set 4 with $c = 1$.

n		GMRES	SCG	SWI(10)	SWI(30)	SWI(50)
3000	Iter	215	166	189	174	171
	CPU	3.0300	4.2487	2.4171	2.2814	2.5199
6000	Iter	276	200	239	211	206
	CPU	14.9939	17.2486	12.0735	11.1798	11.2197
9000	Iter	336	222	245	235	228
	CPU	41.2403	40.3821	27.3158	26.9847	26.6925
12000	Iter	355	239	332	249	245
	CPU	78.2420	77.2898	66.6002	50.7532	50.8405
15000	Iter	371	253	321	264	257
	CPU	135.6248	130.9967	105.1391	87.9882	87.9408

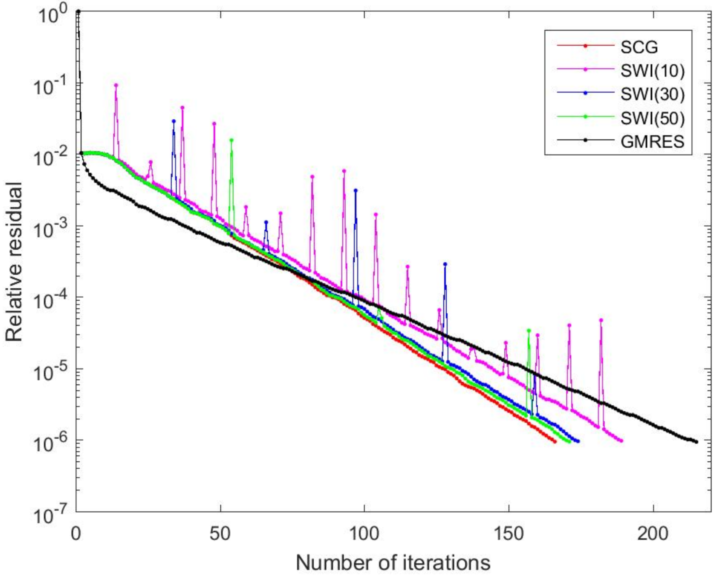

Figure: Relative residual vs. k for test set 4 ($n = 3000$, $c = 1$).

Table: Numerical results for test set 4 with $c = 10$.

n		GMRES	SCG	SWI(20)	SWI(30)	SWI(40)
3000	Iter	78	75	79	78	77
	CPU	1.0892	1.2266	0.9830	0.9885	0.9855
6000	Iter	78	77	80	79	78
	CPU	4.2441	4.7223	3.9465	3.9935	3.9919
9000	Iter	80	78	81	80	79
	CPU	9.8569	11.1940	9.3261	9.4335	9.2561
12000	Iter	84	80	84	83	81
	CPU	18.4798	20.8381	16.8601	17.1527	16.8352
15000	Iter	85	81	84	83	82
	CPU	31.1470	33.5072	27.8080	27.3828	27.3301

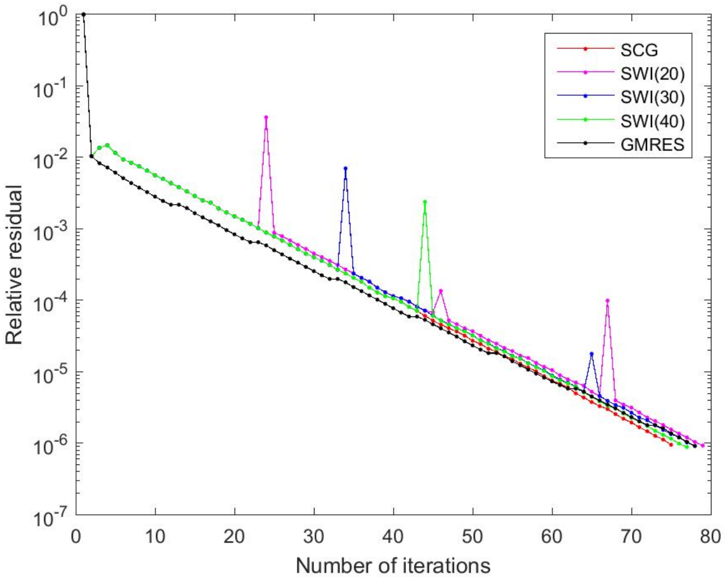

Figure: Relative residual vs. k for test set 4 ($n = 3000$, $c = 10$).