Semi-smooth Newton Type Methods for Composite Convex Programs

Zaiwen Wen

Beijing International Center For Mathematical Research
Peking University
wenzw@pku.edu.cn

Outline

1. composite convex programs
2. Semi-smoothness of proximal mapping
3. semi-smooth Newton methods based on the primal
   - Approach
   - Numerical Results
4. Semi-smooth Newton method based on the dual (SDPNAL)

Composite convex program

Consider the following composite convex program

$$\min_{x \in \mathbb{R}^n} \quad f(x) + h(x),$$

where f and h are convex, f is differentiable but h may not

Many applications:

- Sparse and low rank optimization: $h(x) = \|x\|_1$ or $\|X\|_*$ and many other forms.
- Regularized risk minimization: $f(x) = \sum_i f_i(x)$ is a loss function of some misfit and h is a regularization term.
- Constrained program: h is an indicator function of a convex set.

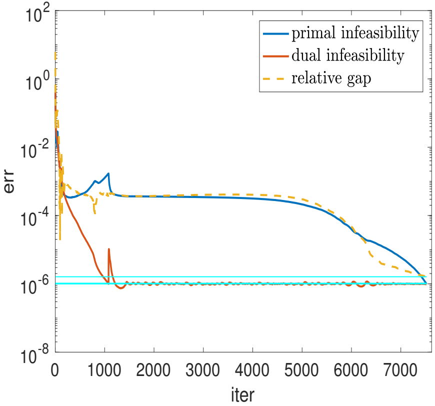
A General Recipe

Goal: study approaches to bridge the gap between first-order and second-order type methods for composite convex programs.

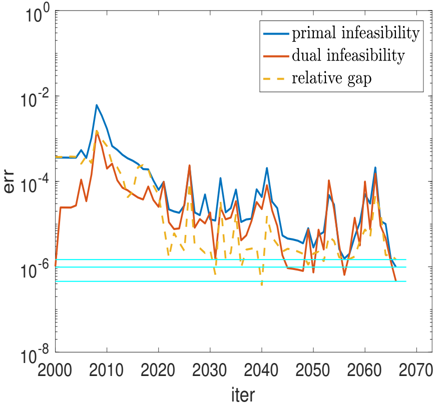
key observations:

- Many popular first-order methods can be equivalent to some fixed-point iterations: $x^{k+1} = T(x^k)$;
  - **Advantages:** easy to implement; converge fast to a solution with moderate accuracy.
  - **Disadvantages:** slow tail convergence.
- The original problem is equivalent to the system $F(x) := (I - T)(x) = 0$.
- Newton-type method since $F(x)$ is semi-smooth in many cases
- Computational costs can be controlled reasonably well

system: BeO



(a) ADMM, CPU: 2003s



(b) Semi-smooth Newton, CPU: 635s

Operator splitting and fixed-point algorithm

Examples:

- forward-backward splitting(FBS).
- Douglas-Rachford splitting(DRS).
- Peaceman-Rachford splitting(PRS).
- alternating direction method of multipliers(ADMM).

Advantages:

- easy to implement;
- converge fast to a solution with moderate accuracy.

Disadvantages:

- slow tail convergence.

Forward-backward splitting (FBS)

- Consider $\min_{x \in \mathbb{R}^n} \quad f(x) + h(x)$
- the *proximal mapping* of f is defined by

$$\text{prox}_{tf}(x) := \operatorname*{argmin}_{u \in \mathbb{R}^n} \{f(u) + \frac{1}{2t}\|u - x\|_2^2\}.$$

- Proximal gradient method or the FBS is the iteration

$$x^{k+1} = \text{prox}_{tf}(x^k - t\nabla h(x^k)), k = 0, 1, \cdots ,$$

- Equivalent to a fixed-point iteration

$$x^{k+1} = T_{\text{FBS}}(x^k).$$

where

$$T_{\text{FBS}} := \text{prox}_{tf} \circ (I - t\nabla h).$$

Douglas-Rachford splitting (DRS)

- DRS is the following update:

$$
\begin{aligned}
x^{k+1} &= \mathrm{prox}_{th}(z^k), \\
y^{k+1} &= \mathrm{prox}_{tf}(2x^{k+1} - z^k), \\
z^{k+1} &= z^k + y^{k+1} - x^{k+1}.
\end{aligned}
$$

- Equivalent to a fixed-point iteration

$$
z^{k+1} = T_{\mathrm{DRS}}(z^k),
$$

where

$$
T_{\mathrm{DRS}} := I + \mathrm{prox}_{tf} \circ (2\mathrm{prox}_{th} - I) - \mathrm{prox}_{th}.
$$

Alternating direction method of multipliers (ADMM)

- Consider a linear constrained program

$$\begin{aligned} \min_{x_1\in\mathbb{R}^{n_1},x_2\in\mathbb{R}^{n_2}} \quad & f_1(x_1)+f_2(x_2) \\ \text{s.t.} \quad & A_1x_1+A_2x_2=b, \end{aligned}$$

- The dual problem is

$$\min_{w\in\mathbb{R}^m} \quad d_1(w)+d_2(w),$$

where $d_1(w) := f_1^*(A_1^T w), \quad d_2(w) := f_2^*(A_2^T w) - b^T w.$

- The ADMM to the primal is equivalent to the DRS to the dual

Outline

Semi-smooth Newton-type method

- Solving the system
$$F(z) = 0,$$
where $F(z) = T(z) - z$ and $T(z)$ is a fixed-point mapping.
- Fixed-point algorithms suffer from slow tail convergence and may not be suitable for high accuracy applications.
- $F(z)$ fails to be differentiable in many interesting applications.
- but $F(z)$ is (strongly) semi-smooth and monotone.
- semi-smooth Newton type method

Semi-smoothness

- $F : \mathcal{O} \to \mathbb{R}^m$ be locally Lipschitz continuous.
- The B-subdifferential of F at x is defined by

$$\partial_B F(x) := \left\{ \lim_{k\to\infty} F'(x^k) | x^k \in D_F, x^k \to x \right\}.$$

  The set

$$\partial F(x) = \mathrm{co}(\partial_B F(x))$$

  is called Clarke's generalized Jacobian
- We say that F is semismooth at $x \in \mathcal{O}$ if
  - F is directionally differentiable at x;
  - for any $d \in \mathcal{O}$ and $J \in \partial F(x+d)$,

$$\|F(x+d) - F(x) - J(d)\| = o(\|d\|) \quad \text{as } d \to 0.$$

- F is said to be strongly semi-smooth at $x \in \mathcal{O}$ if F is semi-smooth and for any $d \in \mathcal{O}$ and $J \in \partial F(x+d)$,

$$\|F(x+d) - F(x) - J(d)\| = O(\|d\|^2) \quad \text{as } d \to 0.$$

Semi-smoothness

- (Strongly) semi-smoothness is closed under scalar multiplication, summation and composition.
- A vector-valued function is (strongly) semi-smooth if and only if each of its component functions is (strongly) semi-smooth.
- Examples:
  - semi-smooth
    - the smooth functions
    - all convex functions (thus norm)
    - the piecewise differentiable functions
  - strongly semi-smooth
    - Differentiable functions with Lipschitz gradients
    - For every $p \in [1, \infty]$, the norm $\| \cdot \|_p$
    - Piecewise affine functions

Semi-smoothness of proximal mappings

- Many commonly seen proximal mappings are semi-smooth
- Examples:
  - The proximal mapping of ℓ_1-norm $\|x\|_1$ (or ℓ_∞-norm $\|x\|_\infty$) is strongly semi-smooth.
  - The projection[1] over a polyhedral set is piecewise linear and hence strongly semi-smooth.
  - The projections over symmetric cones are proved to be strongly semi-smooth.
  - In many applications, the proximal mapping is shown to be piecewise $\mathcal{C}^1$ and hence semi-smooth.

[1] The proximal mapping of an indicator function onto a closed set is the metric projection over this set.

Some concepts on monotonicity

- A mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is said to be monotone, if

$$\langle x - y, F(x) - F(y) \rangle \geq 0, \quad \text{for all } x, y \in \mathbb{R}^n.$$

- A mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is called strongly monotone with modulus $c > 0$ if

$$\langle x - y, F(x) - F(y) \rangle \geq c\|x - y\|_2^2, \quad \text{for all } x, y \in \mathbb{R}^n.$$

- It is said that F is cocoercive with modulus $\beta > 0$ if

$$\langle x - y, F(x) - F(y) \rangle \geq \beta\|F(x) - F(y)\|_2^2, \quad \text{for all } x, y \in \mathbb{R}^n.$$

Monotone mapping

monotone properties of $F_{\text{FBS}} = I - T_{\text{FBS}}$ and $F_{\text{DRS}} = I - T_{\text{DRS}}$:

(i) Suppose that ∇h is cocoercive with $\beta > 0$, then F_{FBS} is monotone if $0 < t \leq 2\beta$.

(ii) Suppose that ∇h is strongly monotone with $c > 0$ and Lipschitz with $L > 0$, then F_{FBS} is strongly monotone if $0 < t < 2c/L^2$.

(iii) Suppose that $h \in C^2$, $H(x) := \nabla^2 h(x)$ is positive semidefinite for any $x \in \mathbb{R}^n$ and $\bar{\lambda} = \max_x \lambda_{\max}(H(x)) < \infty$. Then, F_{FBS} is monotone if $0 < t \leq 2/\bar{\lambda}$.

(iv) The fixed-point mapping F_{DRS} is monotone.

(v) For a monotone and Lipschitz continuous mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ and any $x \in \mathbb{R}^n$, each element of $\partial_B F(x)$ is positive semidefinite.

Outline

Semi-smooth Newton system

- $J_k \in \partial_B F(z^k)$: positively semidefinite.
- regularized Newton's method

$$(J_k + \mu_k I)d = -F_k,$$

where $F_k = F(z^k)$, $\mu_k = \lambda_k \|F_k\|$ and $\lambda_k > 0$ is a regularization parameter.

- solve the linear system inexactly.

$$r_k := (J_k + \mu_k I)d^k + F_k.$$

- seek to step d^k by solving the system approximately such that

$$\|r_k\| \leq \tau \min\{1, \lambda_k \|F_k\| \cdot \|d^k\|\},$$

where $0 < \tau < 1$ is some positive constant.

Semi-smooth Newton method

- Select $0 < v < 1$, $0 < \eta_1 \leq \eta_2 < 1$ and $1 < \gamma_1 \leq \gamma_2$. $\underline{\lambda} > 0$
- A trial point $u^k = z^k + d^k$
- Define a ratio

$$\rho_k = \frac{-\left\langle F(u^k), d^k \right\rangle}{\|d^k\|_F^2}.$$

- Update the point

$$z^{k+1} = \begin{cases} u^k, \ \text{if } \|F(u^k)\|_F \leq \nu \max\limits_{\max(1,k-\zeta+1) \leq j \leq k} \|F(z^j)\|_F, & \text{[Newton]} \\ z^k, \text{otherwise.} & \text{[failed]} \end{cases}$$

- Update the regularization prameter

$$\lambda_{k+1} \in \begin{cases} (\underline{\lambda}, \lambda_k), & \text{if } \rho_k \geq \eta_2, \\ [\lambda_k, \gamma_1 \lambda_k], & \text{if } \eta_1 \leq \rho_k < \eta_2, \\ (\gamma_1 \lambda_k, \gamma_2 \lambda_k], & \text{otherwise,.} \end{cases}$$

Ensuring global convergence I

- If the residual F is not reduced sufficiently or certain other conditions are not met, switching to first order methods. Note that F itself is a first order methods
- construct another point from the Newton step?
- X. Xiao, Y. Li, Z. Wen, L, Zhang, A Regularized Semi-Smooth Newton Method with Projection Steps for Composite Convex Programs, Journal of Scientfic Computing, 2018, Vol 76, No. 1, pp 364-389
- Y. Li, Z. Wen, C. Yang, Y. Yuan, A Semi-smooth Newton Method For semidefinite programs and its applications in electronic structure calculations, SIAM Journal on Scientific Computing, Vol 40, No. 6, 2018, A4131A4157

Ensuring global convergence II: projection step

- $d^k = 0$, then x_k is the optimal solution.
- A trial point
$$u^k = z^k + d^k.$$
- d_k is small enough,
$$\langle F(u^k), z^k - u^k \rangle = -\langle F(u^k), d^k \rangle > 0.$$
- By monotonicity of F, for any optimal solution z^*
$$\langle F(u^k), z^* - u^k \rangle \leq 0.$$
- Therefore the hyperplane
$$H_k := \{z \in \mathbb{R}^n | \langle F(u^k), z - u^k \rangle = 0\}$$
strictly separates z^k from the solution set Z^*.

Ensuring global convergence II: projection step

- Define a ratio
$$\rho_k = \frac{-\left\langle F(u^k), d^k \right\rangle}{\|d^k\|^2}.$$
- If ρ_k is big enough,
$$z^{k+1} = z^k - \frac{\left\langle F(u^k), z^k - u^k \right\rangle}{\|F(u^k)\|^2} F(u^k),$$
which is the projection onto the hyperplane H_k.
- If ρ_k is too small, $z^{k+1} = z^k$ and increase the parameter.

Ensuring global convergence II: projection step

- Select some parameters $0 < \eta_1 \leq \eta_2 < 1$ and $1 < \gamma_1 \leq \gamma_2$. $\underline{\lambda} > 0$ is a small positive constant.
- Update the point

$$z^{k+1} = \begin{cases} z^k - \frac{\langle F(u^k), z^k - u^k \rangle}{\|F(u^k)\|^2} F(u^k), & \text{if } \rho_k \geq \eta_1, \\ z^k, & \text{otherwise.} \end{cases}$$

- Update the regularization prameter

$$\lambda_{k+1} \in \begin{cases} (\underline{\lambda}, \lambda_k), & \text{if } \rho_k \geq \eta_2, \\ [\lambda_k, \gamma_1 \lambda_k], & \text{if } \eta_1 \leq \rho_k < \eta_2, \\ (\gamma_1 \lambda_k, \gamma_2 \lambda_k], & \text{otherwise,.} \end{cases}$$

- For any $z^* \in Z^*$ and any successful iteration

$$\|z^{k+1} - z^*\|^2 \leq \|z^k - z^*\|^2 - \|z^{k+1} - z^k\|^2.$$

Global convergence

Assumption:

- Assume that $F : \mathbb{R}^n \to \mathbb{R}^n$ is strongly semi-smooth and monotone.
- Suppose that there exists a constant $c_1 > 0$ such that $\|J_k\| \leq c_1$ for any $k \geq 0$ and any $J_k \in \partial_B F(z^k)$.

Global Convergence

The sequence $\{z^k\}$ generated by our algorithm converges to some point $\bar{z}$ such that $F(\bar{z}) = 0$ from any initial point.

Local Quadratic convergence

Assumption:

- The mapping F is BD-regular at z^*, that is, all elements in $\partial_B F(z^*)$ are nonsingular.

Local Quadratic convergence

For any Newton step and $z^k \in N(z^*, \varepsilon_1)$ with some $\varepsilon_1 > 0$, we have

$$\|z^{k+1} - z^*\|_2 \leq c_2 \|z^k - z^*\|_2^2,$$

where c_2 is some positive constant.

- If z^k is close enough to z^*, the condition $\|F(u^k)\|_2 \leq \nu \|F(z^k)\|_2$ is always satisfied.
- Our algorithm turns to a second-order Newton method in a neighborhood of z^*.

Applications to the FBS Method

- Consider the ℓ_1-regularized optimization problem of the form

$$\min \ \mu\|x\|_1 + h(x), \quad h(x) = \frac{1}{2}\|Ax - b\|_2^2$$

- Let $f(x) = \mu\|x\|_1$. The system of nonlinear equations is

$$F(x) = x - \mathrm{prox}_{tf}(x - t\nabla h(x)) = 0.$$

- The generalized Jacobian matrix of $F(x)$ is

$$J(x) = I - M(x)(I - t\partial^2 h(x)),$$

where $M(x) \in \partial \mathrm{prox}_{tf}(x - t\nabla h(x))$ and $\partial^2 h(x)$ is the generalized Hessian matrix of $h(x)$.

- $M(z)$ is diagonal matrix whose diagonal entries are

$$(M(z))_{ii} = \begin{cases} 1, & \text{if } |z_i| > \mu t, \\ 0, & \text{otherwise.} \end{cases}$$

l1-regularized optimization problems

- introduce the index sets

$$\mathcal{I}(x) := \{i : |(x - t\nabla h(x))_i| > t\mu\} = \{i : (M(x))_{ii} = 1\},$$
$$\mathcal{O}(x) := \{i : |(x - t\nabla h(x))_i| \le t\mu\} = \{i : (M(x))_{ii} = 0\}.$$

- The Jacobian matrix can be represented by

$$J(x) = \begin{pmatrix} t(\partial^2 h(x))_{\mathcal{I}(x)\mathcal{I}(x)} & t(\partial^2 h(x))_{\mathcal{I}(x)\mathcal{O}(x)} \\ 0 & I \end{pmatrix}.$$

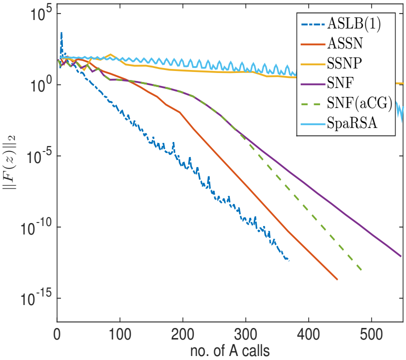
- Let $\mathcal{I} = \mathcal{I}(x^k)$ and $\mathcal{O} = \mathcal{O}(x^k)$. Then one can reduce the Newton system to a small system.

$$s^k_{\mathcal{O}} = -\frac{1}{1+\mu_k} F_{k,\mathcal{O}},$$
$$(t(\partial^2 h(x))_{\mathcal{I}\mathcal{I}} + \mu I)s^k_{\mathcal{I}} = -F_{k,\mathcal{I}} - t(\partial^2 h(x))_{\mathcal{I}\mathcal{O}} s^k_{\mathcal{O}}.$$

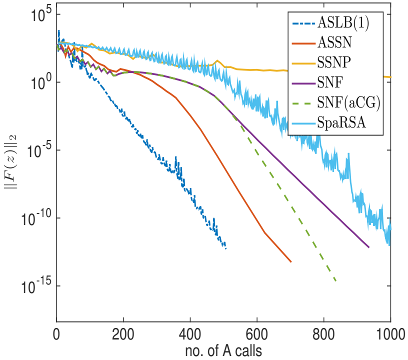
l1-regularized optimization problems

Table: Total number of A- and A^T- calls N_A and CPU time (in seconds) averaged over 10 independent runs with dynamic range 20 dB

method	$\epsilon : 10^{-0}$		$\epsilon : 10^{-2}$		$\epsilon : 10^{-4}$		$\epsilon : 10^{-6}$	
	time	N_A	time	N_A	time	N_A	time	N_A
SNF	1.12	84.6	3.19	254.2	3.87	307	4.5	351
SNF(aCG)	1.11	84.6	3.19	254.2	4.19	331.2	4.3	351.2
ASSN	1.15	89.8	2.2	173	3.15	246.4	3.76	298.2
SSNP	2.52	199	8.05	649.4	20.7	1679.8	29.2	2369.6
ASLB(2)	0.803	57	1.66	121	2.79	202.4	3.63	264.6
ASLB(1)	0.586	42.2	1.29	92	2.54	181.4	3.85	275
FPC-AS	1.45	109.8	7.08	510.4	10	719.8	10.3	743.6
SpaRSA	5.46	517.2	5.9	539.8	6.75	627	9.05	844.4



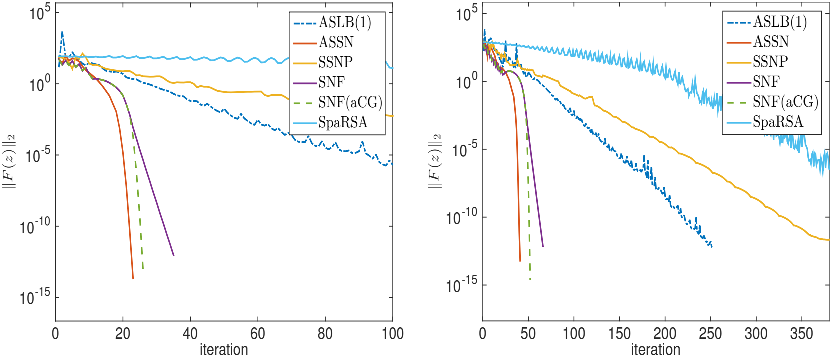
(c) 20dB



(d) 40dB

Figure: residual history with respect to total number of A- and A^T- calls N_A

l1-regularized optimization problems



(a) 20dB (b) 40dB

Figure: residual history with respect to total number of iterations

Applications to the FBS Method

- The fixed-point mapping

$$F(x) = \text{prox}_{tf}(x - t\nabla h(x)) - x.$$

- The generalized Jacobian matrix of $F(x)$ is

$$J(x) = M(x)(I - t\partial^2 h(x)) - I,$$

where $M(x) \in \partial \text{prox}_{tf}(x - t\nabla h(x))$ and $\partial^2 h(x)$ is the generalized Hessian matrix of $h(x)$.

LASSO Regression

- The Lasso regression problem

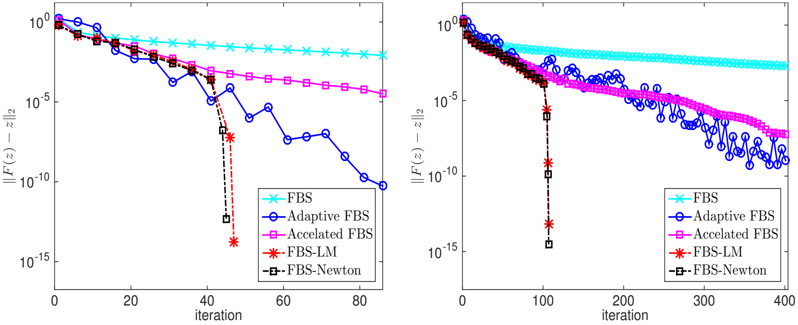
$$\min \frac{1}{2}\|Ax-b\|_2^2 \text{ s.t. } \|x\|_1 \le \lambda,$$

  where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $\lambda \ge 0$ are given.
- $h(x) = \frac{1}{2}\|Ax-b\|_2^2$ and $f(x) = 1_\Omega(x)$, where $\Omega = \{x \mid \|x\|_1 \le \lambda\}$.
- For a given $z \in \mathbb{R}^n$, let $|z_{[1]}| \ge |z_{[2]}| \ge \ldots \ge |z_{[n]}|$, the Jacobian matrix $M(z)$

$$M(z)_{ij} = \begin{cases} 1 & \text{if } \alpha < 0, j = i \\ 1 - \alpha \text{sign}(z_i)\text{sign}(z_j)/p, & \text{if } |z_i| \ge \alpha \text{ and } \alpha > 0, j = [1], \ldots, [p]. \end{cases}$$

  where α be the largest value of $\left(\sum_{i=1}^k |z_{[i]}| - \lambda\right)/k$, $k = 1, \ldots, n$, and p be the corresponding k of α.

LASSO Regression



(a) $k = 50$ (b) $k = 150$

Figure: residual history of LASSO on $n = 1000$, $m = 500$ and $\mu = 0.9\|x\|_1$

Logistic Regression

- Sparse logistic regression problem

$$\min \ \mu\|x\|_1 + h(x),$$

where $\sum_{i=1}^m \log(e^{A_i x} + 1) - b_i^T A_i x$.

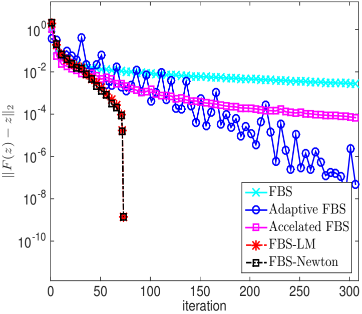
- The proximal mapping corresponding to $f(x) = \mu\|x\|_1$

$$\left(\mathrm{prox}_{tf}(z)\right)_i = \mathrm{sign}(z_i)\max(|z_i| - \mu t, 0).$$

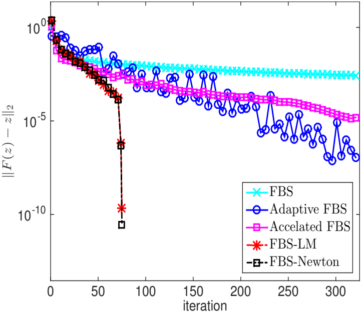
- the Jacobian matrix $M(z)$ is diagonal matrix whose diagonal entries are

$$(M(z))_{ii} = \begin{cases} 1, & \text{if } |z_i| > \mu t, \\ 0, & \text{otherwise.} \end{cases}$$

Logistic Regression



(a) $k = 200$



(b) $k = 600$

Figure: residual history of the logistic regression problem on $n = 2000$, $m = 1000$ and $\mu = 1$

General Quadratic Programming

- The general quadratic programming

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + c^T x, \text{ s.t. } Ax \leq b,$$

  where $Q \in R^{n \times n}$ is symmetric positive definite, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

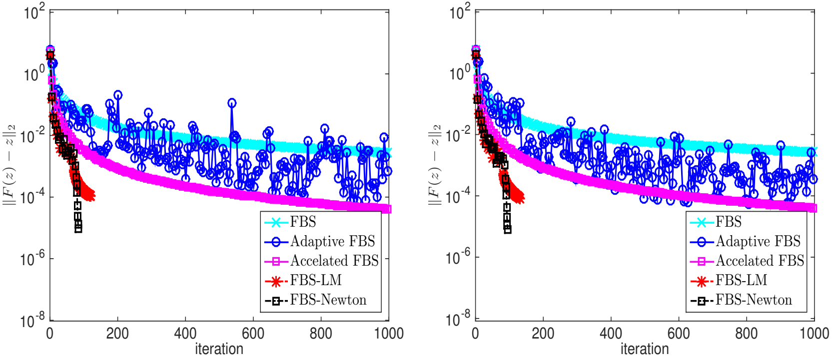
- The dual problem is

$$\max_{y \geq 0} \min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + c^T x + y^T (Ax - b),$$

  which is equivalent to

$$\min_{y \geq 0} \frac{1}{2} y^T (A Q^{-1} A^T) y + (A Q^{-1} c + b)^T y.$$

General Quadratic Programming



(a) LISWET1 (b) LISWET2

Figure: residual history of quadratic programming

Applications to the DRS Method

- Optimization problems

$$\min f(x), \text{ s.t. } Ax = b,$$

where $A \in \mathbb{R}^{m\times n}$ is of full row rank and $b \in \mathbb{R}^m$.

- $h(x) = 1_\Omega(x)$, where $\Omega = \{x \mid Ax = b\}$.
- The proximal mapping with respect to $h(x)$ is

$$\text{prox}_{th}(x) = \mathcal{P}_\Omega(x) = (I - \mathcal{P}_{A^T})x + (A^T(AA^T)^{-1})b,$$

where $\mathcal{P}_{A^T} = A^T(AA^T)^{-1}A$.

Applications to the DRS Method

- The DRS fixed-point mapping reduces to

$$F(z) = \mathrm{prox}_{tf}((2D - I)z + 2\beta) - Dz - \beta,$$

where

$$D = I - \mathcal{P}_{A^T} \quad \text{and} \quad \beta = (A^T(AA^T)^{-1})b.$$

- The generalized Jacobian matrix of $F(z)$ is in the form of

$$J(z) = M(z)(2D - I) - D = \Psi(z) - \Phi(z)\mathcal{P}_{A^T},$$

where $M(z) \in \partial\mathrm{prox}_{tf}((2D - I)z + 2\beta)$, $\Psi(z) = M(z) - I$ and $\Phi(z) = 2M(z) - I$.

Applications to the DRS Method

- The ℓ_1 minimization problem:

$$\min_{x\in\mathbb{R}^n} \quad \|x\|_1, \text{ s.t.} \quad Ax = b.$$

- Let $f(x) = 1_\Omega(Ax - b)$ and $h(x) = \|x\|_1$, where the set $\Omega = \{0\}$. The system of nonlinear equations is

$$F(z) = \text{prox}_{th}(z) - \text{prox}_{tf}(2\text{prox}_{th}(z) - z) = 0.$$

- Hence, a generalized Jacobian matrix of $F(z)$ is in the form of

$$J(z) = M(z) + D(I - 2M(z)).$$

- A generalized Jacobian matrix $M(z) \in \partial\text{prox}_{th}(z)$ is a diagonal matrix with diagonal entries

$$M_{ii}(z) = \begin{cases} 1, & |(z)_i| > t, \\ 0, & \text{otherwise.} \end{cases}$$

Basis Pursuit

- Make the assumption that $AA^\top = I$. Then we can obtain

$$\text{prox}_{tf}(z) = z - A^\top(Az - b).$$

A generalized Jacobian matrix $D \in \partial\text{prox}_{tf}((2\text{prox}_{th}(z) - z))$ is taken as follows

$$D = I - A^\top A.$$

- Let $W = (I - 2M(z))$ and $H = W + M(z) + \mu I$. The diagonal entries of matrix W and H are

$$W_{ii}(z) = \begin{cases} -1, & |(z)_i| > t, \\ 1, & \text{otherwise} \end{cases} \quad \text{and} \quad H_{ii}(z) = \begin{cases} \mu, & |(z)_i| > t, \\ 1 + \mu, & \text{otherwise.} \end{cases}$$

- Using the binomial inverse theorem, we obtain the inverse matrix

$$\begin{aligned} (J(z) + \mu I)^{-1} &= (H - A^\top AW)^{-1} \\ &= H^{-1} + H^{-1}A^\top(I - AWH^{-1}A^\top)^{-1}AWH^{-1}. \end{aligned}$$

Basis Pursuit

- Then $WH^{-1} = \frac{1}{1+\mu}I - S$, where S is a diagonal matrix with diagonal entries

$$S_{ii}(z) = \begin{cases} \frac{1}{\mu} + \frac{1}{1+\mu}, & |(z)_i| > t, \\ 0, & \text{otherwise.} \end{cases}$$

- Hence, $I - AWH^{-1}A^\top = (1 - \frac{1}{1+\mu})I + ASA^\top$.
- Define the index sets

$$\mathcal{I}(x) := \{i : |(z)_i| > t\} = \{i : M_{ii}(x) = 1\},$$
$$\mathcal{O}(x) := \{i : |(z)_i| \leq t\} = \{i : M_{ii}(x) = 0\}$$

- $A_{\mathcal{I}(x)}$ denote the matrix containing the column $\mathcal{I}(x)$ of A, then we have

$$ASA^\top = (\frac{1}{\mu} + \frac{1}{1+\mu})A_{\mathcal{I}(x)}A_{\mathcal{I}(x)}^\top.$$

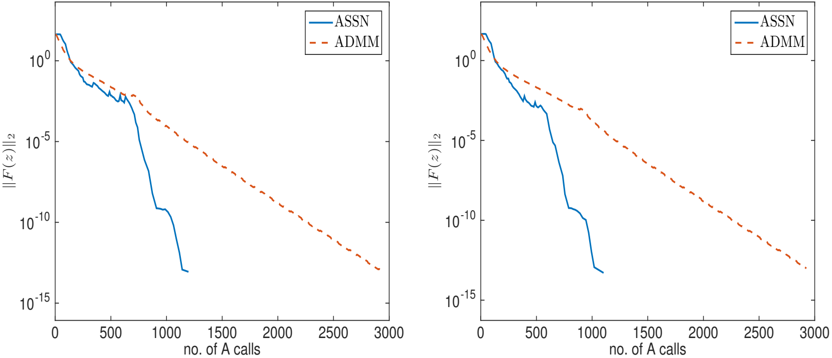
Basis Pursuit

Table: Total number of A- and A^T- calls N_A, CPU time (in seconds) and relative error with dynamic ranges 60dB and 80dB

method	$\epsilon : 10^{-2}$			$\epsilon : 10^{-4}$			$\epsilon : 10^{-6}$		
	time	N_A	rerr	time	N_A	rerr	time	N_A	rerr
ADMM	7.44	599	1.90e-03	13.5	980	2.50e-06	18.7	1403	2.91e-08
ASSN	5.48	449	1.32e-03	9.17	740	1.92e-06	10.2	802	1.93e-08
SPGL1	55.3	2367	5.02e-03	70.7	2978	5.02e-03	89.4	3711	5.02e-03

method	$\epsilon : 10^{-2}$			$\epsilon : 10^{-4}$			$\epsilon : 10^{-6}$		
	time	N_A	rerr	time	N_A	rerr	time	N_A	rerr
ADMM	7.8	592	5.38e-04	13.8	1040	2.48e-06	17.7	1405	2.35e-08
ASSN	4.15	344	5.19e-04	7.92	618	1.21e-06	8.74	702	5.62e-09
SPGL1	32.2	1368	4.86e-04	56.1	2396	4.86e-04	67.4	2840	4.86e-04

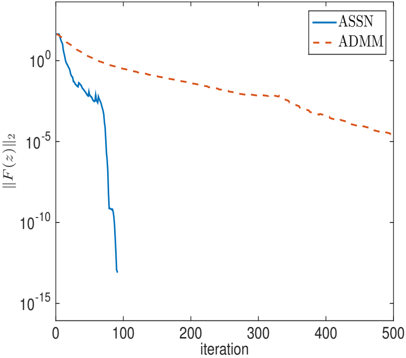
Basis Pursuit



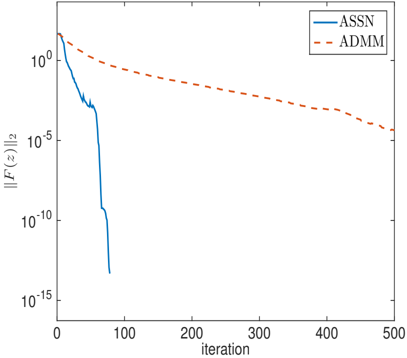
(a) 60dB (b) 80dB

Figure: residual history with respect to the total number of A- and A^T- calls N_A

Basis Pursuit



(a) 60dB



(b) 80dB

Figure: residual history with respect to the total number of iterations

SDP

- Consider the semi-definite programming(SDP)

$$\begin{aligned} \min \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \mathcal{A}X = b, \\ & X \succeq 0 \end{aligned}$$

- $f(X) = \langle C, X \rangle + 1_{\{\mathcal{A}X=b\}}(X)$.
- $h(X) = 1_K(X)$, where $K = \{X : X \succeq 0\}$.
- Proximal Operator:

$$\text{prox}_{th}(Z) = \arg\min_X \frac{1}{2}\|X - Z\|_F^2 + th(X)$$

- Let $Z = Q\Sigma Q^T$ be the spectral decomposition

$$\begin{aligned} \text{prox}_{tf}(Y) &= (Y + tC) - \mathcal{A}^*(\mathcal{A}Y + t\mathcal{A}C - b), \\ \text{prox}_{th}(Z) &= Q_\alpha \Sigma_\alpha Q_\alpha^T, \end{aligned}$$

Semi-smooth Newton System

- assumption: $\mathcal{A}\mathcal{A}^* = I$
- The binomial inverse theorem yields the inverse matrix

$$
\begin{aligned}
(J_k + \mu_k I)^{-1} &= (H - A^T A W)^{-1} \\
&= H^{-1} + H^{-1} A^T (I - AWH^{-1}A^T)^{-1} AWH^{-1}.
\end{aligned}
$$

- computational cost $O(n^2 \min\{r, |n-r|\})$, where r is the rank of primal variable.
- computational cost $O(\sum_i n_i^2 \min\{r_i, |n_i - r_i|\})$, if there is a block diagonal structure.

Semi-smooth Newton method

- Define $T = \tilde{Q}L\tilde{Q}^T$, where L is a diagonal matrix with diagonal entries

$$L_{ii}(z) = \begin{cases} 1, & (\Lambda)_{ii} = 1, \\ \frac{\omega\mu}{\mu+1-\omega}, & (\Lambda)_{ii} = \omega, \\ 0, & (\Lambda)_{ii} = 0. \end{cases}$$

- Then $H^{-1} = \frac{1}{\mu+1}I + \frac{1}{\mu(\mu+1)}T$ and $WH^{-1} = \frac{1}{1+\mu}I - (\frac{1}{\mu} + \frac{1}{\mu+1})T$.
- Hence,

$$\begin{aligned} & (J(Z) + \mu I)^{-1} \\ = \quad & \frac{1}{\mu(\mu+1)}(\mu I + T)(I + A^\top(\frac{\mu^2}{2\mu+1}I + ATA^\top)^{-1}A(\frac{\mu}{2\mu+1}I - T)). \end{aligned}$$

- $ATA^\top d = \mathcal{A}Q(\Omega_0 \circ (Q^T(D)Q))Q^T$, where $D = \mathcal{A}^* d$,

$$\Omega_0 = \begin{bmatrix} E_{\alpha\alpha} & l_{\alpha\bar{\alpha}} \\ l_{\alpha\bar{\alpha}}^T & 0 \end{bmatrix},$$

  and $E_{\alpha\alpha}$ is a matrix of ones and $l_{ij} = \frac{\mu k_{ij}}{\mu+1-k_{ij}}$
- computational cost $O(|\alpha|n^2)$

Switching between the ADMM and Newton steps

the reduced ratios of primal and dual infeasibilities

$$\omega_{\eta_p}^k = \frac{\mathrm{mean}_{k-5\le j\le k}\eta_p^j}{\mathrm{mean}_{k-25\le j\le k-20}\eta_p^j} \text{ and } \omega_{\eta_q}^k = \frac{\mathrm{mean}_{k-5\le j\le k}\eta_q^j}{\mathrm{mean}_{k-25\le j\le k-20}\eta_q^j}.$$

Repeat:

- Semi-smooth Newton steps (doSSN == 1)
  Select $J_k \in \partial_B F(Z^k)$ and solve the Newton system approximately.
  Compute $U^k = Z^k + S^k$. Then update Z^{k+1} and λ_{k+1}.
  If Newton step is failed, set $N_f = N_f + 1$.
  If $N_f \ge \bar{N}_f$ or the Newton step performs bad
  Set doSSN = 0 and parameters for the ADMM steps
- ADMM steps (doSSN == 0)
  Perform an ADMM step. Equivalently, it defines
  $Z^{k+1} = Z^k - F(Z^k)$.
  If the ADMM step performs bad
  Set doSSN = 1, $N_f = 0$ and parameters of the Newton steps

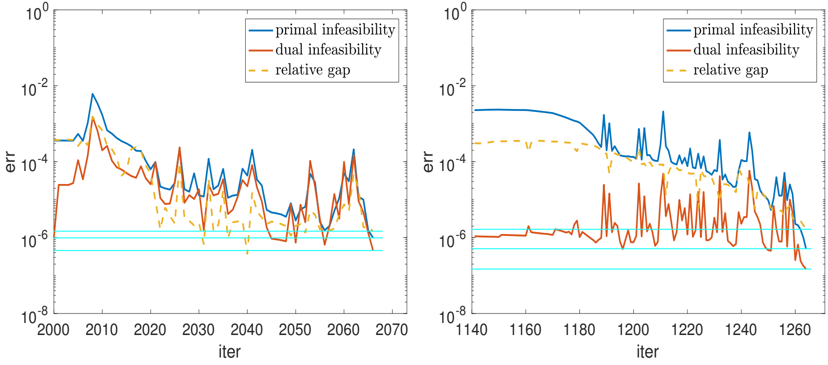
Comparison on electronic structure calculation

- The data set are used in the paper of Nakata, et al. Thanks Prof. Nakata Maho and Prof. Mituhiro Fukuta for sharing all data sets on 2RDM
- solver:
  - SDPNAL: Newton-CG Augmented Lagrangian Method proposed by Zhao, Sun and Toh
  - SDPNAL+: Enhanced version of SDPNAL by Yang, Sun and Toh
  - SSNSDP: the semi-smooth Newton method using stop rules $\eta_p < 3 \times 10^{-6}$ and $\eta_d < 3 \times 10^{-7}$.
- all experiments were performed on a computing cluster with an Intel Xeon 2.40GHz CPU that processes 28 cores and 256GB RAM.
- main criteria:

$$\eta_p = \frac{\|\mathcal{A}(X) - b\|_2}{\max(1, \|b\|_2)} \quad \eta_d = \frac{\|\mathcal{A}^* y - C - S\|_F}{\max(1, \|C\|_F)}$$

$$\eta_g = \frac{|b^T y - \mathrm{tr}(C^T X)|}{\max(1, \mathrm{tr}(C^T X))} \quad \mathrm{err} = b^T y - \mathrm{energy}_{\mathrm{fullCI}}$$

Comparison on electronic structure calculation



(a) BeO (b) C2

Figure: SSNSDP: Relative gap, primal infeasibility and dual infeasibility

Comparison on electronic structure calculation

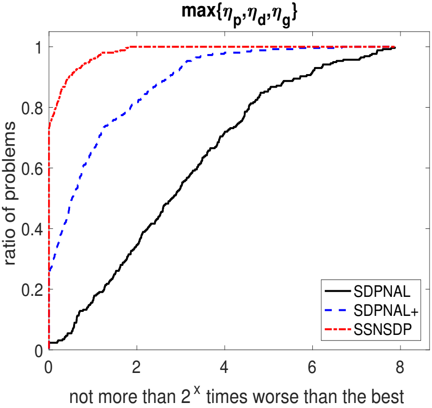
		ADMM						SSNSDP					
system	condition	err	η_p	η_d	η_g	it	t	err	η_p	η_d	η_g	it	t
BH_3O	PQGT1T2'	-1.4-3	9.9-7	8.8-7	2.5-6	2148	3954	-1.4-3	1.0-6	9.1-7	2.3-6	2138	3918
BeO	PQGT1T2	-1.9-3	8.8-7	1.1-6	4.4-7	10261	2003	-2.0-3	4.2-7	3.6-7	1.0-6	1487	635
BeO	PQGT1T2'	-2.0-3	1.0-6	1.0-6	1.6-6	7521	1492	-2.0-3	9.8-7	4.6-7	1.5-6	2066	593
C_2	PQGT1T2	1.7-2	9.5-3	1.9-6	1.6-3	20000	41694	-8.0-3	7.3-7	9.3-7	2.8-5	1165	14074
C_2	PQGT1T2'	-4.0-3	9.5-7	1.3-6	4.9-6	13363	28505	-3.7-3	7.0-7	2.2-7	2.3-6	1440	11849
CH	PQGT1T2	-2.0-3	9.9-7	1.1-6	3.9-6	12723	6292	-1.9-3	3.7-7	6.8-7	7.2-7	1625	1583
CH	PQGT1T2'	-7.5-4	1.0-6	1.1-6	5.1-6	3975	2140	-6.3-4	7.7-7	5.5-7	3.5-6	1597	1432

Figure: Comparison between ADMM and SSNSDP

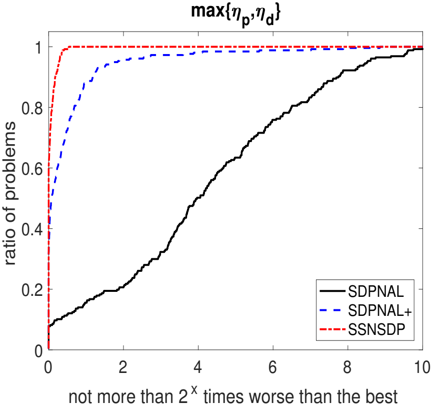
Comparison on electronic structure calculation

	SSNSDP						SDPNAL						SDPNAL+					
system	err	η_p	η_d	η_g	it	t	err	η_p	η_d	η_g	it	t	err	η_p	η_d	η_g	it	t
H_2O	-1.3-3	4.2-7	2.7-7	5.1-6	1605	3788	-1.9-3	7.4-5	4.9-7	1.1-4	266	5595	-2.0-3	7.4-7	8.9-7	1.1-5	3420	9253
H_3	-2.6-5	9.9-7	8.4-7	6.2-6	1511	42	-3.3-5	8.5-7	9.7-7	7.5-6	163	56	-1.6-5	1.1-6	9.4-7	1.9-6	1026	39
HF	-1.8-3	1.0-6	9.3-7	6.7-6	2589	1500	-2.3-3	5.6-5	6.9-7	7.5-5	236	1716	-2.3-3	8.4-7	9.6-7	1.2-5	3062	3208
HLi_2	-8.0-5	8.2-7	4.9-7	2.3-6	1624	791	-2.8-4	1.9-5	7.7-7	2.2-5	260	1105	-9.7-5	1.2-13	1.0-6	2.0-7	3820	1941
HN_2^+	-2.0-3	9.9-7	6.6-7	1.4-6	1742	645	-2.2-3	8.4-6	7.8-7	1.1-5	187	703	-1.9-3	9.1-7	8.3-7	1.2-6	1532	886
HNO	-1.2-3	4.7-7	3.8-7	8.2-7	2065	1984	-1.5-3	1.4-5	7.1-7	2.3-5	213	1530	-1.2-3	8.6-7	9.3-7	1.8-6	1286	1753
Li	-1.9-5	1.9-7	8.3-7	2.1-6	410	36	-1.7-5	2.1-7	6.8-7	1.8-6	145	23	-2.4-5	5.1-7	1.1-6	2.5-6	1123	14
Li_2	-7.3-5	4.3-7	4.4-7	2.2-6	1636	363	-2.0-4	2.5-5	6.9-7	2.5-5	262	497	-2.4-4	3.3-8	1.0-6	8.9-6	5826	1319
LiF	-5.9-4	7.4-7	1.0-6	2.0-6	2813	598	-6.6-4	9.6-6	6.2-7	1.1-5	217	547	-3.5-4	1.0-6	9.0-7	1.5-6	1830	833
LiH(1)	-3.0-5	4.6-7	2.4-7	1.8-6	1715	2273	-1.2-4	2.7-5	7.4-7	1.8-5	253	1765	-2.8-4	5.6-14	5.9-5	4.6-5	17840	10001
LiH(2)	-2.3-5	9.9-7	8.5-7	2.0-6	2154	42	-5.9-5	8.5-6	6.9-7	4.9-6	232	105	-7.1-5	4.3-7	9.8-7	4.5-6	1455	56
LiOH	-9.7-4	1.0-6	9.7-7	2.4-6	2340	809	-1.0-3	1.0-5	5.4-7	1.5-5	203	835	-6.7-4	7.9-7	7.0-7	7.0-7	2098	1499
N	-2.2-4	4.2-7	3.6-7	2.1-6	1608	385	-5.0-4	6.8-5	5.0-7	6.6-5	229	347	-1.1-3	3.8-7	1.4-6	1.1-5	20144	2663
N_2^+	-2.6-3	1.0-6	9.2-7	1.1-6	2434	328	-2.8-3	5.6-6	7.6-7	1.1-5	187	304	-2.8-3	7.6-7	1.0-6	2.0-7	3939	820
N_2	-1.6-3	8.0-7	5.7-7	2.4-6	1036	177	-1.5-3	8.6-6	4.4-7	8.2-6	180	287	-2.0-3	5.1-14	1.5-6	3.3-6	20058	2513
NH(1)	-9.0-4	7.4-7	6.1-7	4.0-6	1599	1468	-1.3-3	4.5-5	5.1-7	7.3-5	264	1998	-1.5-3	1.6-13	1.0-6	8.3-6	3434	3595
NH(2)	-5.1-4	2.9-7	2.2-7	3.0-6	1607	1614	-9.7-4	1.1-4	5.2-7	1.6-4	253	1726	-7.8-4	6.9-13	9.5-7	2.8-6	4046	3909
NH_2^-(1)	-1.3-3	8.0-7	3.1-7	6.1-6	1402	3283	-1.8-3	7.0-5	5.0-7	1.3-4	255	5370	-1.9-3	9.4-7	1.0-6	7.7-6	2602	8306
NH_2^-(2)	-1.9-4	4.9-7	9.2-7	1.3-6	906	44	-1.6-4	1.5-6	4.8-7	1.8-6	171	94	-1.9-4	2.0-7	9.8-7	1.1-6	817	46
NH_3^+	-3.5-4	6.6-7	5.0-7	1.0-6	1096	93	-3.7-4	1.3-6	5.6-7	1.7-6	195	177	-2.7-4	8.9-7	8.0-7	4.3-7	854	133
NH_3	-8.7-4	6.4-7	1.4-7	4.3-6	1307	11463	-1.6-3	9.6-6	5.8-7	1.7-5	259	13131	-2.1-3	9.3-6	4.2-6	1.3-6	298	10220
NH_4^+	-5.2-4	1.0-6	6.5-7	1.3-6	1603	173	-6.1-4	1.9-6	6.3-7	1.8-6	182	190	-6.8-4	4.6-7	9.8-7	2.4-6	1228	196
Na	-4.4-4	1.0-6	8.3-7	1.2-6	1575	127	-5.2-4	4.4-6	6.4-7	3.9-6	184	173	-4.2-4	1.2-7	9.5-7	1.1-6	724	105
NaH	-6.1-4	1.0-6	8.5-7	1.6-6	1782	371	-7.9-4	5.4-6	7.2-7	6.2-6	199	485	-3.9-4	6.3-7	8.4-7	2.8-7	1161	513
Ne	-2.1-3	1.0-6	8.4-7	8.6-6	1967	264	-2.5-3	2.0-5	7.7-7	3.4-5	208	319	-2.6-3	8.6-7	9.2-7	1.2-5	2370	495
O(1)	-1.5-3	9.7-7	1.7-7	2.0-6	1587	332	-2.0-3	2.1-5	4.5-7	2.9-5	216	332	-2.6-3	5.1-10	1.0-6	1.0-5	2651	736
O(2)	-9.1-4	9.5-7	9.8-7	4.6-6	2599	326	-1.2-3	7.4-5	5.6-7	9.1-5	217	328	-1.6-3	5.9-7	9.5-7	1.0-5	1661	542
O(3)	-1.9-3	8.8-7	1.4-7	2.0-6	1575	333	-2.5-3	1.8-5	5.3-7	2.0-5	235	347	-3.0-3	8.2-7	1.0-6	1.0-5	2696	707
O_2^+	-2.3-3	9.9-7	7.8-7	1.6-7	1729	232	-2.4-3	4.4-6	5.6-7	6.5-6	172	289	-2.5-3	3.6-7	9.7-7	1.6-6	939	246
P	-2.8-4	3.3-7	7.3-7	1.7-7	1675	1484	-1.1-3	7.0-6	6.3-7	7.0-6	208	1126	-6.3-4	3.5-13	1.0-6	7.2-7	640	2188
SiH_4	-1.1-3	1.0-6	7.3-7	2.1-6	1657	1471	-1.0-3	5.6-6	5.1-7	4.6-6	185	1715	-3.1-4	3.5-13	1.0-6	1.8-6	817	2322

Comparison on electronic structure calculation

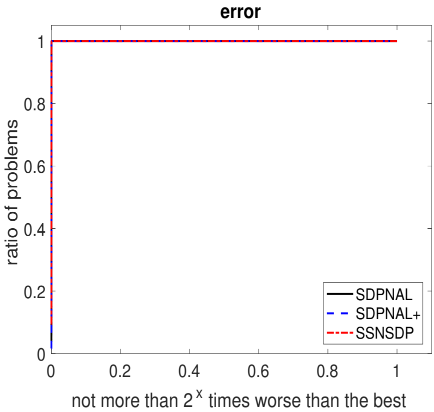


(a) max(η_p,η_d,η_g)

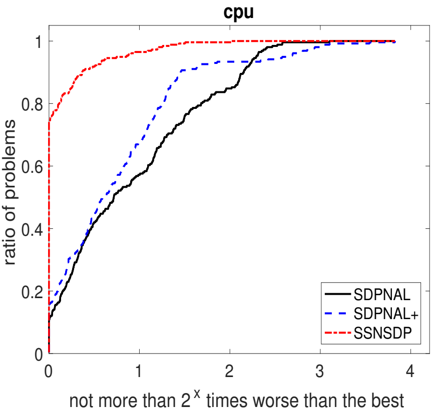


(b) η_d

Comparison on electronic structure calculation



(c) error



(d) cpu time

Comparison on electronic structure calculation

success: $\max\{\eta_p, \eta_d\} \leq 10^{-6}$

case	SSNSDP		SDPNAL		SDPNAL+	
	number	percentage	number	percentage	number	percentage
success	276	100%	53	19.2%	265	96%
fastest	205	74.3%	30	10.9%	41	14.9%
fastest under success	232	84.1%	3	1.09%	41	14.9%
not slower 1.2 times	236	85.5%	71	25.7%	87	31.5%
not slower 1.2 times under success	251	90.9%	5	1.81%	87	31.5%

Figure: Comparison between SDPNAL, SDPNAL+ and SSNSDP

Linear Programming

- The classic linear programming problem

$$\min_{x\in\mathbb{R}^n} \quad c^T x, \text{ s.t.} \quad Ax = b,\ x \geq 0.$$

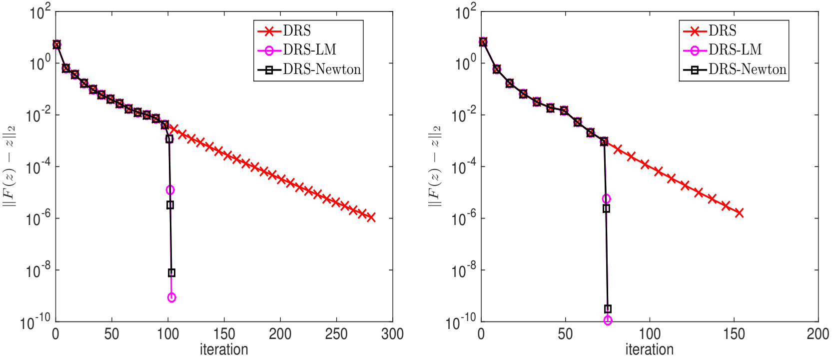
- Let $f(x) = c^T x + 1_K(x)$ where $K := \{x \mid x \geq 0\}$.
- Every element of the generalized Jacobian $\partial\mathcal{P}_K$ at $(2D - I)z + \beta$ is a diagonal matrix with diagonal entries

$$M_{ii}(z) \begin{cases} = 1, & ((2D - I)z + \beta)_i > 0, \\ = 0, & ((2D - I)z + \beta)_i < 0, \\ \in [0, 1], & ((2D - I)z + \beta)_i = 0. \end{cases}$$

- Choose $M(z)$ such that $M_{ii}(z) = 1$ when $((2D - I)z + \beta)_i = 0$.
- we have

$$\begin{cases} \Psi_{ii}(z) = 0, \quad \Phi_{ii}(z) = 1, & ((2D - I)z + \beta)_i \geq 0, \\ \Psi_{ii}(z) = -1, \quad \Phi_{ii}(z) = -1, & ((2D - I)z + \beta)_i < 0. \end{cases}$$

Linear Programming



(a) $m = 300$ (b) $m = 400$

Figure: residual history of the LP problem on $n = 1000$

Outline

SDP

A reference is: Zhao, Xin-Yuan, Defeng Sun, and Kim-Chuan Toh. "A Newton-CG augmented Lagrangian method for semidefinite programming." SIAM Journal on Optimization 20.4 (2010): 1737-1765.
`http://epubs.siam.org/doi/abs/10.1137/080718206`.

- Consider the semi-definite programming (P)

$$\begin{aligned} \min \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \mathcal{A}X = b, \\ & X \succeq 0 \end{aligned}$$

- The dual problem (D) is

$$\begin{aligned} \max \quad & b^\top y \\ \text{s.t.} \quad & \mathcal{A}^* y + S = C, \\ & S \succeq 0 \end{aligned}$$

- the augmented Lagrangian function:

$$L_\sigma(y, S, X^k) = -b^\top y + \langle X, S - \mathcal{A}^* y + C\rangle + \frac{\sigma}{2}\|S - \mathcal{A}^* y + C\|_F^2$$

- Starting from X^0, the augmented Lagrangian method solves the dual problem (D) by

$$(y^{k+1}, S^{k+1}) = \arg\min_{S\succeq 0, y\in\mathbb{R}^m} L_\sigma(y, S, X^k),$$

$$X^{k+1} = X^k + \sigma(S^{k+1} - \mathcal{A}^* y^{k+1} + C),$$

- The variable S is eliminated as $S^{k+1} = \Pi_{\mathcal{S}^n_+}(\mathcal{A}^* y^{k+1} - C - X^k/\sigma)$, where $\Pi_{\mathcal{S}^n_+}$ is the projection on semidefinite matrix cone. Consequently, SDPNAL solves an equivalent form

$$y^{k+1} = \arg\min \tilde{L}_{\sigma^k}(y, X^k) \quad (1)$$

$$X^{k+1} = \Pi_{\mathcal{S}^n_+}(X^k - \sigma(\mathcal{A}^* y^{k+1} - C)), \quad (2)$$

where $\tilde{L}_\sigma(y, X) = b^T y + \frac{1}{2\sigma}(||\Pi_{\mathcal{S}^n_+}(X - \sigma(\mathcal{A}^* y - C))||_F^2 - ||X||_F^2)$.

- Then the subproblem (1) is minimized by using a semismooth Newton method to certain accuracy. The gradient and an alternative element of the generalized Hessian of $\tilde{L}_\sigma(y, X)$ with respect to y is

$$
\begin{aligned}
\nabla_y \tilde{L}_\sigma(y, X) &= b - \mathcal{A}\Pi_{\mathcal{S}^n_+}(X - \sigma(\mathcal{A}^* y - C)), && (3) \\
V &\in \sigma \mathcal{A} \partial \Pi_{\mathcal{S}^n_+}(X - \sigma(\mathcal{A}^* y - C))\mathcal{A}^*. && (4)
\end{aligned}
$$

- For fixed y and X, the corresponding semi-smooth Newton step is

$$(V + \epsilon I)d = \nabla_y L_\sigma(y, X), \quad (5)$$

where ϵ is a small constant.