Second-Order Type Optimization Methods For Data Analysis

Zaiwen Wen

Beijing International Center For Mathematical Research
Peking University
Overview

- Chao Ma, Xin Liu, Zaiwen Wen, Globally Convergent Levenberg-Marquardt Method For Phase Retrieval
Outline

1. Basic Concepts of Semi-smooth Newton method
2. Semi-smooth Newton method for SDP
3. Stochastic Semi-smooth Newton Method
4. Regularized Newton Method for Optimization on Manifold
5. Modified Levenberg-Marquardt Method For Phase Retrieval
Consider the following composite convex program

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

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
An SDP From Electronic Structure Calculation

system: BeO

(a) ADMM, CPU: 2003

(b) Semi-smooth Newton, CPU: 635
Forward-backward splitting (FBS)

- **proximal mapping**:
  \[
  \text{prox}_{th}(x) := \arg\min_{u \in \mathbb{R}^n} \{ h(u) + \frac{1}{2t} \| u - x \|_2^2 \}.
  \]

- FBS is the iteration
  \[
  x^{k+1} = \text{prox}_{th}(x^k - t \nabla f(x^k)), \quad k = 0, 1, \ldots,
  \]
  \[
  = \arg\min_x \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2t} \| x - x^k \|_2^2 + h(x)
  \]

- Equivalent to a fixed-point iteration
  \[
  x^{k+1} = T_{\text{FBS}}(x^k).
  \]

where
\[
T_{\text{FBS}} := \text{prox}_{th} \circ (I - t \nabla f).
\]
Douglas-Rachford splitting (DRS)/ADMM

- DRS is the following update:

\[
\begin{align*}
    x^{k+1} &= \text{prox}_{th}(z^k), \\
    y^{k+1} &= \text{prox}_{tf}(2x^{k+1} - z^k), \\
    z^{k+1} &= z^k + y^{k+1} - x^{k+1}.
\end{align*}
\]

- Equivalent to a fixed-point iteration

\[
    z^{k+1} = T_{\text{DRS}}(z^k),
\]

where

\[
    T_{\text{DRS}} := I + \text{prox}_{tf} \circ (2\text{prox}_{th} - I) - \text{prox}_{th}.
\]

- The ADMM to the primal is equivalent to the DRS to the dual
**Semi-smoothness**

- Solving the system
  
  \[ F(z) = 0, \]

  where \( F(z) = T(z) - z \) and \( T(z) \) is a fixed-point mapping.

- \( F(z) \) fails to be differentiable in many interesting applications.

- but \( F(z) \) is (strongly) semi-smooth and monotone.
  - (a) \( F \) is directionally differentiable at \( x \); and
  - (b) for any \( d \in \mathbb{R}^n \) and \( J \in \partial F(x + d) \),

  \[
  \| F(x + d) - F(x) - Jd \|_2 = o(\|d\|_2) \quad \text{as } d \to 0.
  \]
A regularized semi-smooth Newton method

- The Jacobian $J_k \in \partial_B F(z^k)$ is positive semidefinite
- Let $\mu_k = \lambda_k \|F^k\|_2$. Construct a Newton system:
  \[(J_k + \mu_k I)d = -F^k,\]
- Solving the Newton system inexactley:
  \[r^k := (J_k + \mu_k I)d^k + F^k.\]
- We seek a step $d^k$ approximately such that
  \[\|r^k\|_2 \leq \tau \min\{1, \lambda_k \|F^k\|_2 \|d^k\|_2\}, \quad \text{where } 0 < \tau < 1\]
- Newton Step: $z^{k+1} = z^k + d^k$
- Faster local convergence is ensured
Outline

1. Basic Concepts of Semi-smooth Newton method
2. Semi-smooth Newton method for SDP
3. Stochastic Semi-smooth Newton Method
4. Regularized Newton Method for Optimization on Manifold
5. Modified Levenberg-Marquardt Method For Phase Retrieval
**Semidefinite Programming**

Consider the SDP

\[
\min \langle C, X \rangle, \text{ s.t. } AX = b, X \succeq 0
\]

- \( f(X) = \langle C, X \rangle + 1_{\{AX=b\}}(X) \).
- \( h(X) = 1_K(X) \), where \( K = \{ X : X \succeq 0 \} \).
- Proximal Operator: \( \prox_{th}(Z) = \arg \min_X \frac{1}{2} \| X - Z \|_F^2 + th(X) \)
- Let \( Z = Q\Sigma Q^T \) be the spectral decomposition
  \[
  \prox_{tf}(Y) = (Y + tC) - A^*(AY + tAC - b),
  \prox_{th}(Z) = Q\alpha \Sigma \alpha Q^T,
  \]
- Fixed-point mapping from DRS:
  \[
  F(Z) = \prox_{th}(Z) - \prox_{tf}(2\prox_{th}(Z) - Z) = 0.
  \]
Semi-smooth Newton System

- assumption: $\mathcal{A}^* = I$

- The SMW theorem yields the inverse matrix

$$
(J_k + \mu_k I)^{-1} = H^{-1} + H^{-1} A^T (I - AWH^{-1}A^T)^{-1} AWH^{-1} = \frac{1}{\mu(\mu + 1)}(\mu I + T)(I + A^T \left( \frac{\mu^2}{2\mu + 1} I + ATA^T \right)^{-1} A \left( \frac{\mu}{2\mu + 1} I - T \right)).
$$

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

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

and $E_{\alpha\alpha}$ is a matrix of ones and $l_{ij} = \frac{\mu k_{ij}}{\mu + 1 - k_{ij}}$

- computational cost $O(|\alpha|n^2)$
Semi-smooth Newton method

- Select $0 < \nu < 1$, $0 < \eta_1 \leq \eta_2 < 1$ and $1 < \gamma_1 \leq \gamma_2$. $\lambda > 0$
- A trial point $U^k = Z^k + S^k$
- Define a ratio
  \[ \rho_k = \frac{- \langle F(U^k), S^k \rangle}{\|S^k\|_F^2}. \]
- Update the point
  \[ Z^{k+1} = \begin{cases} 
  U^k, & \text{if } \|F(U^k)\|_F \leq \nu \max_{\max(1,k-\zeta+1) \leq j \leq k} \|F(Z^j)\|_F, \quad \text{[Newton]} \\
  Z^k, & \text{otherwise.} \end{cases} \]
- Update the regularization parameter
  \[ \lambda_{k+1} \in \begin{cases} 
  (\lambda, \lambda_k), & \text{if } \rho_k \geq \eta_2, \\
  [\lambda_k, \gamma_1 \lambda_k], & \text{if } \eta_1 \leq \rho_k < \eta_2, \\
  (\gamma_1 \lambda_k, \gamma_2 \lambda_k], & \text{otherwise.} \end{cases} \]
Switching between the ADMM and Newton steps

the reduced ratios of primal and dual infeasibilities

\[ \omega_{k_{\eta_p}} = \frac{\text{mean}_{k-5 \leq j \leq k} \eta^j_p}{\text{mean}_{k-25 \leq j \leq k-20} \eta^j_p} \quad \text{and} \quad \omega_{k_{\eta_q}} = \frac{\text{mean}_{k-5 \leq j \leq k} \eta^j_q}{\text{mean}_{k-25 \leq j \leq k-20} \eta^j_q} . \]

Repeat:

- **Semi-smooth Newton steps (doSSN == 1)**
  Compute \( U^k = Z^k + S^k \). Then update \( Z^{k+1} \) and \( \lambda_{k+1} \).
  If Newton step is failed, set \( N_f = N_f + 1 \).
  If \( N_f \geq \bar{N}_f \) or the Newton step performs bad
    Set doSSN = 0 and parameters for the ADMM steps

- **ADMM steps (doSSN == 0)**
  Perform an ADMM step.
  If the ADMM step performs bad
    Set doSSN = 1, \( N_f = 0 \) and parameters of the Newton steps
Global Convergence

Theorem

Suppose that \(\{Z^k\} \) is a sequence generated by the semismooth Newton method. Then the residuals of \(\{Z^k\} \) converge to 0, i.e.,

\[
\lim_{k \to \infty} \|F(Z^k)\| = 0.
\]

- If \(\{Z^k\} \) is bounded, then any accumulation point of \(\{Z^k\} \) converges to some point \(\tilde{Z}\) such that \(F(\tilde{Z}) = 0\).
- This algorithm can solve the general composite optimization.
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
    \[ \eta_p < 3 \times 10^{-6} \text{ and } \eta_d < 3 \times 10^{-7}. \]

- all experiments were performed on a computing cluster with an Intel Xeon 2.40GHz CPU that processes 28 cores and 256GB RAM.

- main criteria:
  \[
  \begin{align*}
  \eta_p &= \frac{\| A(X) - b \|_2}{\max(1, \| b \|_2)} \\
  \eta_d &= \frac{\| A^* y - C - S \|_F}{\max(1, \| C \|_F)} \\
  \eta_g &= \frac{| b^T y - \text{tr}(C^T X) |}{\max(1, \text{tr}(C^T X))} \\
  \text{err} &= b^T y - \text{energy}_{\text{fullCI}}
  \end{align*}
  \]
Computational Results: C2

(c) ADMM, CPU: 41694

(d) Semi-smooth Newton, CPU: 14074
Comparison on electronic structure calculation

\begin{align*}
\text{(e) } & \max \{ \eta_p, \eta_d, \eta_g \} \\
\text{(f) } & \text{cpu time}
\end{align*}
success: $\max\{\eta_p, \eta_d\} \leq 10^{-6}$

<table>
<thead>
<tr>
<th>case</th>
<th>SSNSDP</th>
<th>SDPNAL</th>
<th>SDPNAL+</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>number</td>
<td>percentage</td>
<td>number</td>
</tr>
<tr>
<td>success</td>
<td>276</td>
<td>100%</td>
<td>53</td>
</tr>
<tr>
<td>fastest</td>
<td>205</td>
<td>74.3%</td>
<td>30</td>
</tr>
<tr>
<td>fastest under success</td>
<td>232</td>
<td>84.1%</td>
<td>3</td>
</tr>
<tr>
<td>not slower 1.2 times</td>
<td>236</td>
<td>85.5%</td>
<td>71</td>
</tr>
<tr>
<td>not slower 1.2 times under success</td>
<td>251</td>
<td>90.9%</td>
<td>5</td>
</tr>
</tbody>
</table>

**Figure**: Comparison between SDPNAL, SDPNAL+ and SSNSDP
Outline

1. Basic Concepts of Semi-smooth Newton method
2. Semi-smooth Newton method for SDP
3. Stochastic Semi-smooth Newton Method
4. Regularized Newton Method for Optimization on Manifold
5. Modified Levenberg-Marquardt Method For Phase Retrieval
Examples and Applications

Consider
\[ \min_x f(x) + h(x) =: \psi(x) \]

Expected and Empirical Risk Minimization:
\[ f(x) := \mathbb{E}[F(x, \xi)] = \int_{\Omega} F(x, \xi(\omega)) \, dP(\omega), \quad f(x) := \frac{1}{N} \sum_{i=1}^{N} f_i(x) \]

Applications and Typical Situation:

- Large-scale machine learning problems, LASSO, sparse and bilinear logistic regression, low-rank matrix completion, sparse dictionary learning, ...
- \( P \) is not known (completely) or \( N \) is very large.

\( \Rightarrow \) Full evaluation of \( f \) and \( \nabla f \) is impractical or even not possible.

\( \Rightarrow \) Use stoch. optimization techniques and sampling strategies!
Algorithmic Idea

Basic idea based on $x^{k+1} = T_{\text{FBS}}(x^k) = \text{prox}^h(x^k - t\nabla f(x^k))$.

- The proximity operator [Moreau ’65]

$$\text{prox}^h(y) := \arg\min_z h(z) + \frac{1}{2}\|y - z\|_h^2.$$ 

- We incorporate second order information and use stochastic Hessian oracles (SSO)

$$H(x^k; t^k) \approx \nabla^2 f(x^k)$$

to estimate the Hessian $\nabla^2 f$ and compute the Newton step.

- The sample collections $s^k$ and $t^k$ are chosen independently of each other and of the other batches $s^\ell$, $t^\ell$, $\ell \in \mathcal{N}_0 \setminus \{k\}$.

- Let $G : \mathbb{R}^n \times \Xi \to \mathbb{R}^n$ and $H : \mathbb{R}^n \times \Xi \to \mathbb{S}^n$ be Carathéodory functions. We work with the following SFO and SSO:

$$G_{s^k}(x) := \frac{1}{n^g_k} \sum_{i=1}^{n^g_k} G(x; s^k_i) \quad \text{and} \quad H_{t^k}(x) := \frac{1}{n^h_k} \sum_{j=1}^{n^h_k} H(x; t^k_j).$$
Stochastic Semi-smooth Newton Method: Idea

To accelerate the stochastic proximal gradient method, we want to augment it by a stochastic Newton-type step, obtained from the (sub-sampled) optimality condition:

\[ F_s^\Lambda(x) = x - \text{prox}_h^\Lambda(x - \Lambda^{-1}G_s(x)) \approx 0. \]

The semi-smooth Newton step is given by

\[ M_k d^k = -F_s^\Lambda(x^k), \quad x^{k+1} = x^k + d^k, \]

with sample batches \( s^k, t^k \) and \( M_k \in \mathcal{M}_{s^k,t^k}^\Lambda(x^k) \),

\[ \mathcal{M}_{s,t}^\Lambda(x) := \{ M = I - D + D\Lambda^{-1}H_t(x) : D \in \partial \text{prox}_h^\Lambda(u_s^\Lambda(x)) \} \]

and \( u_s^\Lambda(x) := x - \Lambda^{-1}G_s(x) \).

\[ \implies \text{Aim: Utilize fast local convergence to stationary points!} \]
Stochastic Semismooth Newton Method

Sub-sampled Semi-smooth Newton Method (S4N)

0. Choose $x^0 \in \text{dom } h$, batch sizes $(n^g_k), (n^h_k)$, matrices $(\Lambda_k)$, and step sizes $(\alpha_k)$. Select ind. batches $s^0, t^0$. Set $k := 0$.

While “not converged” do:

1. Compute $F_{s_k}^\Lambda(x^k)$ and choose $M_k \in M_{s_k,t_k}^\Lambda(x^k)$. Select new sample batches $s^{k+1}, t^{k+1}$.

2. Compute the semismooth Newton step via

$$M_k d^k = -F_{s_k}^\Lambda(x^k).$$

If this is not possible, go to step 4.

3. Set $z^k := x^k + d^k$. If $z^k \in \text{dom } h$ and $z^k$ satisfies the growth conditions ($\star$), set $x^{k+1} := z^k$ and go to step 5.

4. Compute a proximal gradient step $x^{k+1} := x^k - \alpha_k F_{s_k}^\Lambda(x^k)$.

5. Increment $k$ and go to step 1.
Algorithmic Framework (Cont’)

We use the following growth conditions (∗) in step 3:

\[
\|F_{s_{k+1}}^{\Lambda_{k+1}}(z^k)\| \leq (\eta + \nu_k) \cdot \theta_k + \epsilon_{k}^{1}, \quad (G.1)
\]

\[
\psi(z^k) \leq \psi(x^k) + \beta \cdot \theta_k^{1/2} \|F_{s_{k+1}}^{\Lambda_{k+1}}(z^k)\|^{1/2} + \epsilon_{2}^{k}, \quad (G.2)
\]

where \(\eta \in (0, 1), \beta > 0,\) and \((\nu_k), (\epsilon_{2}^{k}) \in \ell_+^1, (\epsilon_{k}^{1}) \in \ell_+^{1/2} \).

We set \(\theta_{k+1}\) to \(\|F_{s_{k+1}}^{\Lambda_{k+1}}(x^{k+1})\|\) if \(x^{k+1}\) was obtained in step 3.

Remark:

- Calculating \(F_{s_{k+1}}^{\Lambda_{k+1}}(z^k)\) requires evaluation of \(G_{s_{k+1}}(z^k)\). This information can be reused in the next iteration if \(z^k \mapsto x^{k+1}\) is accepted as new iterate.
Global Convergence: Assumptions

Basic Assumptions:

(A.1) \( \nabla f \) is Lipschitz continuous on \( \mathbb{R}^n \) with constant \( L \).
(A.2) The matrices \( (\Lambda_k) \subset S^{++}_n \) satisfy \( \lambda_M I \preceq \Lambda_k \geq \lambda_m I \) for all \( k \).
(A.3) \( \psi \) is bounded from below on \( \text{dom } h \).

Stochastic Assumptions:

(S.1) For all \( k \in \mathcal{N} \), there exists \( \sigma_k \geq 0 \) such that
\[
\mathbb{E}[\|\nabla f(x^k) - G_{s_k}(x^k)\|^2] \leq \sigma_k^2.
\]
(S.2) The matrices \( M_k \), chosen in step 1, are random operators.
Global Convergence

**Theorem: Global Convergence [MXCW, ’17]**

Suppose that (A.1)–(A.3) and (S.1)–(S.2) are fulfilled. Then, under the additional conditions, $\alpha_k \leq \alpha := \min\{1, \lambda_m/L\}$,

$$(\alpha_k) \text{ is nonincreasing, } \sum \alpha_k = \infty, \sum \alpha_k \sigma_k^2 < \infty$$

it holds $\liminf_{k \to \infty} \mathbb{E}[\|F^\Lambda(x^k)\|^2] = 0$ and $\liminf_{k \to \infty} F^\Lambda(x^k) = 0$ a.s. for any $\Lambda \in \mathbb{S}^n_{++}$.

- Verify that $(x^k)$ actually defines an adapted stochastic process.
- The batch $s^k$ and the iterate $x^k$ are not independent.
- Derive approximate and uniform descent estimates for the terms $\psi(x^k) - \psi(x^{k+1})$.

**For strongly convex case:** $\lim_{k \to \infty} \mathbb{E}[\|F^\Lambda(x^k)\|^2] = 0$ and $\lim_{k \to \infty} F^\Lambda(x^k) = 0$ a.s. for any $\Lambda \in \mathbb{S}^n_{++}$. 
Numerical Results: Sparse Logistic Regression

We consider the following $\ell_1$-regularized logistic regression problem

$$
\min_x \frac{1}{N} \sum_{i=1}^N f_i(x) + \mu \|x\|_1, \quad f_i(x) := \log(1 + \exp(-b_i \cdot a_i^T x))
$$

where $a_i^T \in \mathbb{R}^n$ denotes the $i$th row of the data matrix $A \in \mathbb{R}^{N \times n}$ and $b \in \{-1, 1\}^N$ is a binary vector.

Specifications of the test framework:

<table>
<thead>
<tr>
<th>dataset</th>
<th>data points $N$</th>
<th>features $n$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>covtype</td>
<td>581 012</td>
<td>54</td>
<td>5e-3</td>
</tr>
<tr>
<td>gisette</td>
<td>6 000</td>
<td>5 000</td>
<td>5e-2</td>
</tr>
<tr>
<td>rcv1</td>
<td>20 242</td>
<td>47 236</td>
<td>1e-3</td>
</tr>
</tbody>
</table>

\(^1\)LIBSVM - www.csie.ntu.edu.tw/ cjlin/libsvm/
Numerical Comparisons - covtype, Epochs

![Graph showing numerical comparisons for covtype dataset across different epochs. The graph plots relative sub-optimality against epochs, with various algorithms such as SVRG, Adagrad, S4N-HG [10%), S4N-HG [50%), S4N-HG [100%), S4N-H, and S4N-VR.]
Numerical Comparisons - covtype, Time

![Graph showing numerical comparisons over time for different methods including SVRG, Adagrad, S4N-HG [10%, 50%, 100%], S4N-H, S4N-VR.]
Numerical Comparisons - gisette, Epochs
Numerical Comparisons - gisette, Time

The graph shows the relative sub-optimality over time elapsed (sec) for different algorithms and hyperparameter settings.

- SVRG
- Adagrad
- S4N-HG [10%]
- S4N-HG [50%]
- S4N-HG [100%]
- S4N-H
- S4N-VR
Outline

1. Basic Concepts of Semi-smooth Newton method
2. Semi-smooth Newton method for SDP
3. Stochastic Semi-smooth Newton Method
4. Regularized Newton Method for Optimization on Manifold
5. Modified Levenberg-Marquardt Method For Phase Retrieval
Optimization on Riemannian Manifold

Problem definition

\[
\min_{x \in \mathcal{M}} f(x),
\]

where \( \mathcal{M} \) is a Riemannian manifold.

- Common matrix manifolds
  
  - **Stiefel Manifold**: \( St(p, n) \) \( \equiv \) \( \{ X \in \mathbb{R}^{n \times p} | X^T X = I_p \} \)
  
  - **Grassmann manifold**: \( Grass(p, n) \) denote the set of all \( p \)-dimensional subspaces of \( \mathbb{R}^n \)
  
  - **Oblique manifold**: \( \{ X \in \mathbb{R}^{n \times p} | \text{diag}(X^T X) = I_p \} \)
  
  - **Rank-\( p \) manifold**: \( \{ X \in \mathbb{R}^{m \times n} : \text{rank}(X) = p \} \)
Electronic Structure Calculation

- Total energy minimization:
  \[
  \min_{X^*X = I} E_{kinetic}(X) + E_{ion}(X) + E_{Hartree}(X) + E_{xc}(X) + E_{fock}(X),
  \]
  where
  \[
  \begin{align*}
  E_{kinetic}(X) &= \frac{1}{2} \text{tr}(X^* L X) \\
  E_{ion}(X) &= \text{tr}(X^* V_{ion} X) \\
  E_{Hartree}(X) &= \frac{1}{2} \rho(X)^\top L^\dagger \rho(X) \\
  E_{xc}(X) &= \rho(X)^\top \mu_{xc}(\rho(X)) \\
  \rho(X) &= \text{diag}(D(X)), \quad D(X) = XX^* \\
  E_{fock}(X) &= \langle V(D)X, X \rangle
  \end{align*}
  \]

- Nonlinear eigenvalue problem (looks like the KKT conditions):
  \[
  H(X)X = X \Lambda \\
  X^* X = I
  \]
Bose-Einstein condensates

Minimization problem

$$\min_{\phi \in S} E(\phi),$$

$$E(\phi) = \int_{\mathbb{R}^d} \left[ \frac{1}{2} |\nabla \phi(x)|^2 + V_d(x) |\phi(x)|^2 + \frac{\beta_d}{2} |\phi(x)|^4 - \Omega \phi(x) L_z \phi(x) \right] dx$$

$$S = \left\{ \phi \mid E(\phi) < \infty, \int_{\mathbb{R}^d} |\phi(x)|^2 dx = 1 \right\}.$$

Discretized problem

$$\min_{X \in \mathbb{C}^N} \mathcal{F}(X) := \frac{1}{2} X^* A X + \alpha \sum_{i=1}^{N} |X_i|^4, \text{ s.t. } \|X\|_F = 1.$$
Cryo-electron microscopy reconstruction

Find 3D structure given samples of 2D projections images
Thanks: Amit Singer
Regularized Newton Method

- Our new adaptively regularized Newton (ARNT) method:

\[
\begin{aligned}
\min \quad m_k(x) := & \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle H_k[x - x_k], x - x_k \rangle + \frac{\sigma_k}{2} \|x - x_k\|^2, \\
s.t. \quad & x \in M,
\end{aligned}
\]

where \(\nabla f(x_k)\) and \(H_k\) are the Euclidean gradient Hessian.

- Regularized parameter update (trust-region-like strategy):
  - ratio: \(\rho_k = \frac{f(z_k) - f(x_k)}{m_k(z_k)}\).
  - regularization parameter \(\sigma_k\):

\[
\sigma_{k+1} \in \begin{cases} 
(0, \sigma_k] & \text{if } \rho_k > \eta_2, \Rightarrow x_{k+1} = Z_k \\
[\sigma_k, \gamma_1 \sigma_k] & \text{if } \eta_1 \leq \rho_k \leq \eta_2, \Rightarrow x_{k+1} = Z_k \\
[\gamma_1 \sigma_k, \gamma_2 \sigma_k] & \text{otherwise.} \Rightarrow x_{k+1} = x_k
\end{cases}
\]

where \(0 < \eta_1 \leq \eta_2 < 1\) and \(1 < \gamma_1 \leq \gamma_2\).
Solvers for subproblem

- The subproblem implicitly preserve the Lagrangian multipliers

\[
\text{Hess} m_k(x_k)[\xi] = P_{x_k}(H_k[\xi] - U_{\text{sym}}((x_k)^*\nabla f(x_k))) + \tau_k \xi,
\]

- Riemannian Gradient method with BB step size.

- Newton system for the subproblem

\[
\text{grad } m_k(x_k) + \text{Hess} m_k(x_k)[\xi] = 0.
\]

- Modified CG method

\[
\xi_k = \begin{cases} 
  s_k + \tau_k d_k & \text{if } d_k \neq 0, \\
  s_k & \text{if } d_k = 0,
\end{cases}
\]

with \( \tau_k := \frac{\langle d_k, \text{grad } m_k(x_k) \rangle_{x_k}}{\langle d_k, \text{Hess} m_k(x_k)[d_k] \rangle_{x_k}} \)

- \( d_k \) represents and transports the negative curvature information
- \( s^k \) corresponds to the “usual” output of the CG method.
### Hartree-Fock total energy minimization

<table>
<thead>
<tr>
<th>Solver</th>
<th>fval</th>
<th>nrmG</th>
<th>its</th>
<th>time</th>
<th>fval</th>
<th>nrmG</th>
<th>its</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ctube661</td>
<td></td>
<td></td>
<td></td>
<td>glutamine</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACE</td>
<td>-1.43e+2</td>
<td>9.2e-7</td>
<td>8(2.8)</td>
<td>795</td>
<td>-1.04e+2</td>
<td>3.9e-7</td>
<td>10(3.0)</td>
<td>229</td>
</tr>
<tr>
<td>GBBN</td>
<td>-1.43e+2</td>
<td>6.5e-7</td>
<td>10(26.3)</td>
<td>1399</td>
<td>-1.04e+2</td>
<td>8.4e-7</td>
<td>11(13.3)</td>
<td>256</td>
</tr>
<tr>
<td>ARN</td>
<td>-1.43e+2</td>
<td>6.0e-7</td>
<td>9(14.1)</td>
<td>832</td>
<td>-1.04e+2</td>
<td>8.8e-7</td>
<td>10(9.5)</td>
<td>209</td>
</tr>
<tr>
<td>ARQN</td>
<td>-1.43e+2</td>
<td>2.0e-7</td>
<td>8(13.2)</td>
<td>777</td>
<td>-1.04e+2</td>
<td>1.5e-7</td>
<td>8(10.1)</td>
<td>182</td>
</tr>
<tr>
<td>AKQN</td>
<td>-1.43e+2</td>
<td>6.1e-7</td>
<td>17(10.3)</td>
<td>1502</td>
<td>-1.04e+2</td>
<td>9.1e-7</td>
<td>25(6.0)</td>
<td>515</td>
</tr>
<tr>
<td>RQN</td>
<td>-1.43e+2</td>
<td>7.2e-6</td>
<td>59</td>
<td>6509.0</td>
<td>-1.04e+2</td>
<td>2.9e-6</td>
<td>57</td>
<td>1532</td>
</tr>
<tr>
<td></td>
<td>graphene16</td>
<td></td>
<td></td>
<td></td>
<td>graphene30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACE</td>
<td>-1.01e+2</td>
<td>7.6e-7</td>
<td>13(3.4)</td>
<td>367</td>
<td>-1.87e+2</td>
<td>8.6e-7</td>
<td>58(4.2)</td>
<td>14992</td>
</tr>
<tr>
<td>GBBN</td>
<td>-1.01e+2</td>
<td>4.2e-7</td>
<td>14(42.1)</td>
<td>659</td>
<td>-1.87e+2</td>
<td>8.9e-7</td>
<td>29(72.2)</td>
<td>19701</td>
</tr>
<tr>
<td>ARN</td>
<td>-1.01e+2</td>
<td>4.5e-7</td>
<td>14(23.0)</td>
<td>403</td>
<td>-1.87e+2</td>
<td>9.0e-7</td>
<td>45(35.6)</td>
<td>14860</td>
</tr>
<tr>
<td>ARQN</td>
<td>-1.01e+2</td>
<td>4.9e-7</td>
<td>11(20.2)</td>
<td>357</td>
<td>-1.87e+2</td>
<td>7.6e-7</td>
<td>15(26.5)</td>
<td>6183</td>
</tr>
<tr>
<td>AKQN</td>
<td>-1.01e+2</td>
<td>7.9e-7</td>
<td>49(15.1)</td>
<td>1011</td>
<td>-1.87e+2</td>
<td>8.0e-7</td>
<td>39(12.3)</td>
<td>9770</td>
</tr>
<tr>
<td>RQN</td>
<td>-1.01e+2</td>
<td>1.0e-3</td>
<td>74</td>
<td>2978</td>
<td>-1.87e+2</td>
<td>1.5e-5</td>
<td>110</td>
<td>39091</td>
</tr>
</tbody>
</table>
Bose-Einstein condensates

<table>
<thead>
<tr>
<th>solver</th>
<th>f</th>
<th>its</th>
<th>nrmG</th>
<th>time</th>
<th>f</th>
<th>its</th>
<th>nrmG</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( \beta = 500 \)

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( \Omega = 0.70 \)

<table>
<thead>
<tr>
<th>OptM</th>
<th>6.9731</th>
<th>340</th>
<th>1.0e-4</th>
<th>56.3</th>
<th>OptM</th>
<th>6.1016</th>
<th>386</th>
<th>1.0e-4</th>
<th>65.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRQH</td>
<td>6.9731</td>
<td>7 (55)</td>
<td>2.0e-4</td>
<td>61.6</td>
<td>TRQH</td>
<td>6.1016</td>
<td>5 (64)</td>
<td>2.0e-4</td>
<td>83.1</td>
</tr>
<tr>
<td>ARNT</td>
<td>6.9731</td>
<td>10 (99)</td>
<td>8.7e-5</td>
<td>44.4</td>
<td>ARNT</td>
<td>6.1016</td>
<td>10 (104)</td>
<td>8.7e-5</td>
<td>70.6</td>
</tr>
<tr>
<td>RTR</td>
<td>6.9731</td>
<td>99 (118)</td>
<td>9.3e-5</td>
<td>234.2</td>
<td>RTR</td>
<td>6.1016</td>
<td>18 (130)</td>
<td>7.7e-5</td>
<td>130.1</td>
</tr>
</tbody>
</table>

\( \Omega = 0.80 \)

<table>
<thead>
<tr>
<th>OptM</th>
<th>6.1016</th>
<th>386</th>
<th>1.0e-4</th>
<th>65.2</th>
<th>OptM</th>
<th>6.1016</th>
<th>386</th>
<th>1.0e-4</th>
<th>65.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRQH</td>
<td>6.1016</td>
<td>5 (64)</td>
<td>2.0e-4</td>
<td>83.1</td>
<td>TRQH</td>
<td>6.1016</td>
<td>5 (64)</td>
<td>2.0e-4</td>
<td>83.1</td>
</tr>
<tr>
<td>ARNT</td>
<td>6.1016</td>
<td>10 (104)</td>
<td>8.7e-5</td>
<td>70.6</td>
<td>ARNT</td>
<td>6.1016</td>
<td>10 (104)</td>
<td>8.7e-5</td>
<td>70.6</td>
</tr>
<tr>
<td>RTR</td>
<td>6.1016</td>
<td>18 (130)</td>
<td>7.7e-5</td>
<td>130.1</td>
<td>RTR</td>
<td>6.1016</td>
<td>18 (130)</td>
<td>7.7e-5</td>
<td>130.1</td>
</tr>
</tbody>
</table>

\( \Omega = 0.90 \)

<table>
<thead>
<tr>
<th>OptM</th>
<th>4.7784</th>
<th>10000</th>
<th>1.2e-3</th>
<th>243.6</th>
<th>OptM</th>
<th>3.7419</th>
<th>10000</th>
<th>7.4e-4</th>
<th>241.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRQH</td>
<td>4.7778</td>
<td>277 (176)</td>
<td>2.0e-4</td>
<td>1090.9</td>
<td>TRQH</td>
<td>3.7416</td>
<td>363 (181)</td>
<td>2.0e-4</td>
<td>1185.1</td>
</tr>
<tr>
<td>ARNT</td>
<td>4.7777</td>
<td>147 (132)</td>
<td>9.6e-5</td>
<td>413.3</td>
<td>ARNT</td>
<td>3.7414</td>
<td>500 (147)</td>
<td>2.6e-4</td>
<td>1204.0</td>
</tr>
<tr>
<td>RTR</td>
<td>4.7777</td>
<td>500 (147)</td>
<td>8.5e-4</td>
<td>1250.4</td>
<td>RTR</td>
<td>3.7415</td>
<td>500 (172)</td>
<td>9.7e-4</td>
<td>1419.0</td>
</tr>
</tbody>
</table>

\( \Omega = 0.95 \)

<table>
<thead>
<tr>
<th>OptM</th>
<th>4.7777</th>
<th>500 (147)</th>
<th>8.5e-4</th>
<th>1250.4</th>
<th>OptM</th>
<th>3.7415</th>
<th>500 (172)</th>
<th>9.7e-4</th>
<th>1419.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRQH</td>
<td>4.7777</td>
<td>500 (147)</td>
<td>8.5e-4</td>
<td>1250.4</td>
<td>TRQH</td>
<td>3.7415</td>
<td>500 (172)</td>
<td>9.7e-4</td>
<td>1419.0</td>
</tr>
<tr>
<td>ARNT</td>
<td>4.7777</td>
<td>500 (147)</td>
<td>8.5e-4</td>
<td>1250.4</td>
<td>ARNT</td>
<td>3.7415</td>
<td>500 (172)</td>
<td>9.7e-4</td>
<td>1419.0</td>
</tr>
<tr>
<td>RTR</td>
<td>4.7777</td>
<td>500 (147)</td>
<td>8.5e-4</td>
<td>1250.4</td>
<td>RTR</td>
<td>3.7415</td>
<td>500 (172)</td>
<td>9.7e-4</td>
<td>1419.0</td>
</tr>
</tbody>
</table>

\( \beta = 1000 \)

<table>
<thead>
<tr>
<th>OptM</th>
<th>9.5283</th>
<th>990</th>
<th>1.0e-4</th>
<th>63.7</th>
<th>OptM</th>
<th>8.2627</th>
<th>10000</th>
<th>5.5e-4</th>
<th>231.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRQH</td>
<td>9.5301</td>
<td>102 (156)</td>
<td>2.0e-4</td>
<td>404.1</td>
<td>TRQH</td>
<td>8.2610</td>
<td>453 (177)</td>
<td>2.0e-4</td>
<td>1427.0</td>
</tr>
<tr>
<td>ARNT</td>
<td>9.5301</td>
<td>60 (81)</td>
<td>9.3e-5</td>
<td>140.4</td>
<td>ARNT</td>
<td>8.2610</td>
<td>202 (105)</td>
<td>6.7e-5</td>
<td>412.7</td>
</tr>
<tr>
<td>RTR</td>
<td>9.5301</td>
<td>293 (91)</td>
<td>8.6e-5</td>
<td>478.8</td>
<td>RTR</td>
<td>8.2610</td>
<td>500 (113)</td>
<td>5.5e-4</td>
<td>972.7</td>
</tr>
</tbody>
</table>

\( \Omega = 0.70 \)

<table>
<thead>
<tr>
<th>OptM</th>
<th>9.5301</th>
<th>102 (156)</th>
<th>2.0e-4</th>
<th>404.1</th>
<th>OptM</th>
<th>8.2610</th>
<th>453 (177)</th>
<th>2.0e-4</th>
<th>1427.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRQH</td>
<td>9.5301</td>
<td>60 (81)</td>
<td>9.3e-5</td>
<td>140.4</td>
<td>TRQH</td>
<td>8.2610</td>
<td>202 (105)</td>
<td>6.7e-5</td>
<td>412.7</td>
</tr>
<tr>
<td>ARNT</td>
<td>9.5301</td>
<td>293 (91)</td>
<td>8.6e-5</td>
<td>478.8</td>
<td>ARNT</td>
<td>8.2610</td>
<td>500 (113)</td>
<td>5.5e-4</td>
<td>972.7</td>
</tr>
<tr>
<td>RTR</td>
<td>9.5301</td>
<td>293 (91)</td>
<td>8.6e-5</td>
<td>478.8</td>
<td>RTR</td>
<td>8.2610</td>
<td>500 (113)</td>
<td>5.5e-4</td>
<td>972.7</td>
</tr>
</tbody>
</table>

\( \Omega = 0.95 \)

<table>
<thead>
<tr>
<th>OptM</th>
<th>9.5301</th>
<th>60 (81)</th>
<th>9.3e-5</th>
<th>140.4</th>
<th>OptM</th>
<th>8.2610</th>
<th>202 (105)</th>
<th>6.7e-5</th>
<th>412.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRQH</td>
<td>9.5301</td>
<td>293 (91)</td>
<td>8.6e-5</td>
<td>478.8</td>
<td>TRQH</td>
<td>8.2610</td>
<td>500 (113)</td>
<td>5.5e-4</td>
<td>972.7</td>
</tr>
<tr>
<td>ARNT</td>
<td>9.5301</td>
<td>293 (91)</td>
<td>8.6e-5</td>
<td>478.8</td>
<td>ARNT</td>
<td>8.2610</td>
<td>500 (113)</td>
<td>5.5e-4</td>
<td>972.7</td>
</tr>
<tr>
<td>RTR</td>
<td>9.5301</td>
<td>293 (91)</td>
<td>8.6e-5</td>
<td>478.8</td>
<td>RTR</td>
<td>8.2610</td>
<td>500 (113)</td>
<td>5.5e-4</td>
<td>972.7</td>
</tr>
</tbody>
</table>
Outline

1. Basic Concepts of Semi-smooth Newton method
2. Semi-smooth Newton method for SDP
3. Stochastic Semi-smooth Newton Method
4. Regularized Newton Method for Optimization on Manifold
5. Modified Levenberg-Marquardt Method For Phase Retrieval
Phase retrieval

Detectors record **intensities** of diffracted rays $\implies$ phaseless data only!

- Recover $x$ from phaseless measurements about $x \in \mathbb{C}^n$
  
  $$\text{find } x, \text{ s.t. } |Ax| = b.$$  

- An equivalent model
  
  $$\min_{x \in \mathbb{C}^n, y \in \mathbb{R}^m} \frac{1}{2} ||Ax - y||_2^2, \text{ s.t. } |y| = b.$$  

Applications: Hubble Space Telescope, X-ray crystallography
Phase retrieval by non-convex optimization

Solve the system of quadratic equations:

\[ y_r = |\langle a_r, x \rangle|^2, \quad r = 1, 2, \ldots, m. \]

- Gaussian model:
  \[ a_r \in \mathbb{C}^n \ i.i.d. \mathcal{N}(0, I/2) + i\mathcal{N}(0, I/2). \]

Nonlinear least square problem

\[
\min_{z \in \mathbb{C}^n} f(z) = \frac{1}{4m} \sum_{k=1}^{m} (y_k - |\langle a_k, z \rangle|^2)^2
\]

\( f \) is nonconvex, many local minima
Wirtinger flow: Candes, Li and Soltanolkotabi (’14)

**Spectral Initialization:**

1. Input measurements \( \{a_r\} \) and observation \( \{y_r\} (r = 1, 2, ..., m) \).
2. Calculate \( z_0 \) to be the leading eigenvector of \( Y = \frac{1}{m} \sum_{r=1}^{m} y_r a_r a_r^* \).
3. Normalize \( z_0 \) such that \( ||z_0||^2 = n \frac{\sum_{r=1}^{m} y_r}{\sum_{r=1}^{m} ||a_r||^2} \).

**Iteration via Wirtinger derivatives:** for \( \tau = 0, 1, \ldots \)

\[
Z_{\tau+1} = Z_{\tau} - \frac{\mu_{\tau+1}}{||Z_{\tau}||^2} \nabla f(Z_{\tau})
\]
The Modified LM method for Phase Retrieval

Levenberg-Marquardt Iteration:

\[ z_{k+1} = z_k - (\psi(z_k) + \mu_k I)^{-1} g(z_k) \]

Algorithm

1. **Input:** Measurements \( \{a_r\} \), observations \( \{y_r\} \). Set \( \epsilon \geq 0 \).
2. Construct \( z_0 \) using the spectral initialization algorithms.
3. **While** \( \|g(z_k)\| \geq \epsilon \) **do**
   - Compute \( s_k \) by solving equation
     \[ \psi_{z_k}^{\mu_k} s_k = (\psi(z_k) + \mu_k I) s_k = -g(z_k). \]
     until \( \|\psi_{z_k}^{\mu_k} s_k + g(z_k)\| \leq \eta_k \|g(z_k)\| \).
   - Set \( z_{k+1} = z_k + s_k \) and \( k := k + 1 \).
3. **Output:** \( z_k \).
Convergence of the Gaussian Model

Theorem

If the measurements follow the Gaussian model, the LM equation is solved accurately ($\eta_k = 0$ for all $k$), and the following conditions hold:

- $m \geq cn \log n$, where $c$ is sufficiently large;
- If $f(z_k) \geq \frac{\|z_k\|^2}{900n}$, let $\mu_k = 70000n \sqrt{nf(z_k)}$; if else, let $\mu_k = \sqrt{f(z_k)}$.

Then, with probability at least $1 - 15e^{-\gamma n} - 8/n^2 - me^{-1.5n}$, we have $\text{dist}(z_0, x) \leq (1/8)\|x\|$, and

$$\text{dist}(z_{k+1}, x) \leq c_1 \text{dist}(z_k, x),$$

Meanwhile, once $f(z_s) < \frac{\|z_s\|^2}{900n}$, for any $k \geq s$ we have

$$\text{dist}(z_{k+1}, x) < c_2 \text{dist}(z_k, x)^2.$$
Key to proof

**Lower bound of GN matrix’s second smallest eigenvalue**

For any $y, z \in \mathbb{C}^n$, $\text{Im}(y^*z) = 0$, we have:

$$y^*\Psi(z)y \geq \|y\|^2\|z\|^2,$$

holds with high probability.

$$\text{Im}(y^*z) = 0 \Rightarrow \|(\Psi_z^\mu)^{-1}y\| \leq \frac{2}{\|z\|^2 + \mu}\|y\|.$$
Key to proof

### Local error bound property

\[
\frac{1}{4} \text{dist}(z, x)^2 \leq f(z) \leq 8.04 \text{dist}(z, x)^2 + 6.06n \text{dist}(z, x)^4,
\]

holds for any \( z \) satisfying \( \text{dist}(z, x) \leq \frac{1}{8} \).

### Regularity condition

\[
\mu(z) h^* \left( \psi_z^\mu \right)^{-1} g(z) \geq \frac{1}{16} \|h\|^2 + \frac{1}{64100n\|h\|} \|g(z)\|^2
\]

holds for any \( z = x + h, \|h\| \leq \frac{1}{8} \), and \( f(z) \geq \frac{\|z\|^2}{900n} \).
Numerical Result: Natural Image

Figure: The Milky Way Galaxy. Image size is 1080×1920 pixels. For the ALM method, the CPU time is 20240.64s, with a final relative error to be $2.44 \times 10^{-16}$; for the ILM algorithm, the CPU time is 4733.43s, and the final relative error is $2.42 \times 10^{-16}$; for WF algorithm, the CPU time is 5211.35s, while the final relative error is $4.91 \times 10^{-16}$
Numerical Result: Natural Image

Figure: Relation between relative error and CPU time used for natural images recovery.
Contact Information

Many Thanks For Your Attention!

- Looking for Ph.D students and Postdoc
- Competitive salary as U.S and Europe
- http://bicmr.pku.edu.cn/~wenzw
- E-mail: wenzw@pku.edu.cn
- Office phone: 86-10-62744125