Adaptive regularized Newton method for optimization on Riemannian manifold

Jiang Hu

Peking University

Joint work with Andre Milzarek, Zaiwen Wen and Yaxiang Yuan

Optimization on Manifold

Problem definition

$$\min_{x \in \mathcal{M}} \quad f(x),$$

where f is a smooth function and $\mathcal{M}$ is a Riemannian submanifold of an Euclidean space.

- Common matrix manifolds
  - Stiefel Manifold: $St(p, n) \triangleq \{X \in \mathbb{R}^{n \times p} \,|\, X^\top X = I_p\}$
  - Oblique manifold: $\{X \in \mathbb{R}^{n \times p} \mid \text{diag}(X^T X) = I_p\}$
  - Rank-p manifold: $\{X \in \mathbb{R}^{m \times n} \mid \text{rank}(X) = p\}$

Applications

- Low rank nearest correlation matrix estimation (D. Simon et al. 2010, Y. Gao et al. 2010)
- Kohn-Sham total energy minimization (C. Yang et al. 2009)
- Bose-Einstein condensates (X. Wu et al. 2015)
- Low-rank matrix completion (B. Vandereycken 2013)
- Other applications

Tangent vector and Riemannian gradient

Definition (Tangent vector)

A tangent vector ξ_x to a manifold $\mathcal{M}$ at a point x is a mapping from any real-valued function g defined on $\mathcal{M}$ to $\mathbb{R}$ such that there exists a curve γ on $\mathcal{M}$ with $\gamma(0) = x$, satisfying

$$\xi_x g = \frac{\mathrm{d}(g(\gamma(t)))}{\mathrm{d}t} \Big|_{t=0} .$$

Definition (Gradient)

Given a smooth real-valued function f on a Riemannian manifold $\mathcal{M}$, define its gradient $\mathrm{grad}\, f$ as the unique element of $T_x\mathcal{M}$ satisfying

$$\langle \mathrm{grad}\, f(x), \xi \rangle_x = \mathcal{D}f(x)[\xi], \quad \forall \xi \in T_x\mathcal{M}.$$

known as the steepest ascent direction.

Connection between Riemannian metric and gradient

- Sphere: $S^{n-1} = \{x \in R^n \mid \|x\|_2 = 1\}$.
- tangent space at x: $\{z \in \mathbb{R}^n | \, x^\top z = 0\}$.
- Riemannian metric: $\langle \xi, \eta \rangle_x := \xi^\top \eta, \quad \forall \xi, \eta \in T_x S^{n-1}$.
- Riemannian gradient:

$$\nabla^{(R)} f(X) = (I - xx^\top) \nabla^{(E)} f(x),$$

which is determined by the Riemannian metric.

Algorithm

Definition (Retraction)

A retraction R_x on a manifold $\mathcal{M}$ at a point x is a mapping from tangent space $T_x\mathcal{M}$ at x onto $\mathcal{M}$ satisfying

- $R_x(0_x) = x$, where 0_x denotes the zero tangent vector of $T_x\mathcal{M}$.
- $\mathcal{D}R_x(0_x) = \mathrm{id}_{T_x\mathcal{M}}$, where $\mathrm{id}_{T_x\mathcal{M}}$ denotes the identity mapping on $T_x\mathcal{M}$.

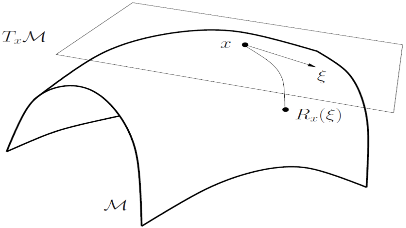

Figure 4.1 Retraction.

Figure : Absil et al. 2008

Retraction

Examples of Retraction

Stiefel manifold: $St(p,n) = \{X \in R^{n\times p} \,|\, X^\top X = I_p\}$. D is a tangent vector at X.

- Geodesic-like update scheme:
  $Y_{geol}(\tau, X) = [X, Q] \exp\left(\tau \begin{bmatrix} -X^T D & -R^T \\ R & 0 \end{bmatrix}\right)\begin{pmatrix} I_p \\ 0 \end{pmatrix}$, where
  $QR = -(I_n - XX^T)D$ is the QR factorization of $-(I_n - XX^T)D$.

- Projection-like update scheme:
  - $Y_p(\tau, X) = \mathbb{P}_{St(p,n)}(X - \tau D)$ where $\mathbb{P}_{St(p,n)}(Z) = Z(Z^\top Z)^{-1/2}$.
  - $Y_{qr}(\tau, X) = \mathrm{qr}(X - \tau D)$ where $qr(Z)$ is the Q factor of QR decomposition $Z = QR$.

Curvilinear search on Riemannian manifold

Curvilinear search updating formula

$$x_{k+1} = R_{x_k}(t_k \eta_k).$$

- R_{x_k} is a retraction at x_k.
- η_k is chosen as descent direction, i.e., $\langle \operatorname{grad} f(x_k), \eta_k \rangle_{x_k} < 0$.
- t_k as the step size is chosen properly

  Non-monotone Armijio rule: Given $\rho, \delta \in (0, 1)$, find the smallest integer h satisfying:

  $$f(R_{x_k}(t_k \eta_k)) \leq C_k + \rho t_k \left\langle \operatorname{grad} f(x_k), \eta_k \right\rangle_{x_k},$$

  where $t_k = \gamma_k \delta^h$ and γ_k is the initial step size (BB step size is a good choice). $C_{k+1} = (\eta Q_k C_k + f(x_{k+1}))/Q_{k+1}$, where $C_0 = f(x_0)$, $Q_{k+1} = \eta Q_k + 1$ and $Q_0 = 1$.

First-order algorithm

- Riemannian gradient method:

$$x_{k+1} = R_{x_k}(-\alpha \mathrm{grad} f(x_k)),$$

  where $\mathrm{grad} f(x_k) \in T_{x_k}\mathcal{M}$ is the Riemannian gradient and $\alpha > 0$ is a step size.

- Riemannian Adaptive gradient method:

$$\begin{cases} G_k = G_{k-1} + \mathrm{grad} f(x_k) \odot \mathrm{grad} f(x_k) \\ x_k = \mathbb{P}_{\mathcal{M}}(x_k - \alpha \mathrm{grad} f(x_k) \oslash \sqrt{G_k + \epsilon}) \end{cases}$$

  where $\mathbb{P}_{\mathcal{M}}$ is the projection operator onto M and $\alpha > 0$ is a step size.

Second-order algorithm

- Classical Riemannian trust-region (RTR) method (Absil et al. 2007):

$$\begin{cases} \min_{\xi \in T_{x_k}\mathcal{M}} & m_k(\xi) := f(x_k) + \langle \operatorname{grad} f(x_k), \xi \rangle_{x_k} + \frac{1}{2} \langle \operatorname{Hess} f(x_k)[\xi], \xi \rangle_{x_k}, \\ \text{s.t.} & \|\xi\|_{x_k} \leq \Delta_k, \end{cases}$$

  where $\operatorname{grad} f(x_k)$ is the Riemannian gradient and $\operatorname{Hess} f(x_k)$ is the Riemannian Hessian.

- Our new adaptive regularized Newton (ARNT) method:

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

  where $\nabla f(x_k)$ is the Euclidean gradient and H_k is the Euclidean Hessian or some approximation.

Algorithmic framework

Subproblem

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

Regularized parameter update (trust-region-like strategy):

- ratio: $\rho_k = \frac{f(z_k) - f(x_k)}{m_k(z_k)}$.
- regularization parameter σ_k:

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

where $0 < \eta_1 \le \eta_2 < 1$ and $1 < \gamma_1 \le \gamma_2$.

Solvers for subproblem

- Fast local convergence rate can be guaranteed if subproblem is solved accurately enough.
- Riemannian Gradient method with BB step size (GBB).
  - It can be faster than directly using GBB to solve the original problem because of the simpler objective function.
  - It may converge slowly when the iteration is close to the optimal solution.
- Modified CG method
  - We perform one Riemannian Newton step and use a modified CG method to solve the Newton equation inexactly.
  - Negative curvature directions are used to construct new search directions.

An adaptive regularized Newton method

Modified CG method

- Set $\xi_0 = 0,\ p_0 = -\operatorname{grad} m(x_k)$ and $i = 0$.
- If $\langle p_i, \mathrm{Hess} m_k(x_k)[p_i]\rangle / \|p_i\|_{x_k}^2 \le -\epsilon$, then

$$\xi_k = \xi_{k-1} + \left\langle \operatorname{grad} m_k(x_k), p_j \right\rangle_{x_k} / \left\langle p_j, \mathrm{Hess} m_k(x_k)[p_j] \right\rangle_{x_k} p_i,$$

  and return. Otherwise, do the normal CG update to obtain the direction ξ_k.

An Adaptive Regularized Newton Method

- Use modified CG method to obtain the search direction ξ_k
- Do Armijo search to obtain a new trial point z_k
- Compute ratio $\rho_k = \frac{f(z_k) - f(x_k)}{m_k(z_k)}$
- Update x_{k+1} and regularization parameter σ_k according to (1)

Algorithm 2: A Modified CG Method for Solving Subproblem (3.4)

S0 Set $T > 0$, $\theta > 1$, $\epsilon \geq 0$, $\eta_0 = 0$, $r_0 = \operatorname{grad} m_k(x_k)$, $p_0 = -r_0$, and $i = 0$.

while $i \leq n - 1$ **do**

- **S1** Compute $\pi_i = \langle p_i, \operatorname{Hess} m_k(x_k)[p_i] \rangle_{x_k}$.
- **S2** **if** $\pi_i \,/\, \langle p_i, p_i \rangle_{x_k} \leq \epsilon$ **then**
  - **if** $i = 0$ **then** set $s_k = -p_0$, $d_k = 0$;
  - **else** set $s_k = \eta_i$,
  - **if** $\pi_i \,/\, \langle p_i, p_i \rangle_{x_k} \leq -\epsilon$ **then** $d_k = p_i$, set $\sigma_{est} = |\pi_i| \,/\, \langle p_i, p_i \rangle_{x_k}$;
  - **else** $d_k = 0$;
  - break;
- **S3** Set $\alpha_i = \langle r_i, r_i \rangle_{x_k} \,/\, \pi_i$, $\eta_{i+1} = \eta_i + \alpha_i p_i$, and $r_{i+1} = r_i + \alpha_i \operatorname{Hess} m_k(x_k)[p_i]$.
- **S4** **if** $\|r_{i+1}\|_{x_k} \leq \min\{\|r_0\|_{x_k}^{\theta}, T\}$ **then**
  - choose $s_k = \eta_{i+1}, d_k = 0$; break;
- **S5** Set $\beta_{i+1} = \langle r_{i+1}, r_{i+1} \rangle_{x_k} \,/\, \langle r_i, r_i \rangle_{x_k}$ and $p_{i+1} = -r_{i+1} + \beta_{i+1} p_i$.
- $i \leftarrow i + 1$.

S6 Update ξ_k according to (3.7).

Global convergence

Assumptions

(A.1) The gradient ∇f is Lipschitz continuous on the convex hull of the manifold $\mathcal{M}$ – denoted by conv($\mathcal{M}$), i.e., there exists $L_f > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \le L_f \|x - y\|, \quad \forall\, x, y \in \text{conv}(\mathcal{M}).$$

(A.2) There exists $\kappa_g > 0$ such that $\|\nabla f(x_k)\| \le \kappa_g$ for all $k \in \mathcal{N}$.

(A.3) There exists $\kappa_H > 0$ such that $\|H_k\| \le \kappa_H$ for all $k \in \mathcal{N}$.

(A.4) The Euclidean and the Riemannian Hessian are bounded, i.e., there exist κ_F and $\kappa_R \ge 1$ such that

$$\|\nabla^2 f(x_k)\| \le \kappa_F \quad \text{and} \quad \|\mathrm{Hess} f(x_k)\|_{x_k} \le \kappa_R, \quad \forall\, k \in \mathcal{N}.$$

Global convergence

- Descent direction: The direction ξ_k obtained from modified CG is a descent direction, i.e., there exists some constant β such that

$$\frac{\langle \xi_k, \operatorname{grad} f(x_k)\rangle_{x_k}}{\|\operatorname{grad} m_k(x_k)\|_{x_k}\|\xi_k\|_{x_k}} \leq -\beta.$$

- Sufficient descent: Suppose that the assumptions (A.2)–(A.4) are satisfied. Let $\rho \in (0,1)$ be arbitrary and set $z_k(t) := R_{x_k}(t\xi_k)$. Then, we have

$$m_k(z_k(t)) \leq \rho t \langle \operatorname{grad} m_k(x_k), \xi_k\rangle_{x_k}, \quad \forall\, t \in [0, \zeta_k], \tag{2}$$

where

$$\zeta_k := \min\left\{\frac{\chi}{\|\xi_k\|_{x_k}}, \frac{2(1-\rho)\beta}{(\kappa_2\kappa_g + \kappa_1^2(\kappa_H + \sigma_k))}\frac{\|\operatorname{grad} f(x_k)\|_{x_k}}{\|\xi_k\|_{x_k}}\right\} \tag{3}$$

and κ_1, κ_2, χ are constants that do not depend on x_k.

Global convergence

Lemma: σ_k is bounded

Suppose that the assumptions (A.1)–(A.4) are satisfied and there exists $\tau > 0$ such that $\|\text{grad} f(x_k)\|_{x_k} \geq \tau$ for all $k \in \mathcal{N}$. Then, the sequence $\{\sigma_k\}$ is bounded, i.e., there exists $L_\tau \geq 0$ such that

$$\sigma_k \leq L_\tau, \quad \forall\, k \in \mathcal{N}.$$

Theorem

Suppose that the assumptions (A.1)–(A.4) *hold and let* $\{f(x_k)\}$ *be bounded from below. Then, either*

$$\text{grad} f(x_\ell) = 0 \;\; \textit{for some} \;\; \ell \geq 0 \quad \textit{or} \quad \lim_{\ell \to \infty} \|\text{grad} f(x_\ell)\|_{x_\ell} = 0.$$

Local convergence

Assumptions

(B.1) The sequence $\{x_k\}$ converges to x_*.

(B.2) The Euclidean Hessian $\nabla^2 f$ is continuous on $\mathrm{conv}(\mathcal{M})$.

(B.3) The Riemannian Hessian $\mathrm{Hess} f$ is positive definite at x_* and the constant ϵ in Algorithm is set to zero.

(B.4) H_k is a good approximation of the Euclidean Hessian $\nabla^2 f$, i.e., it holds

$$\|H_k - \nabla^2 f(x_k)\| \to 0, \quad \text{whenever} \quad \|\mathrm{grad} f(x_k)\|_{x_k} \to 0.$$

(B.5) Suppose that there exists $\beta_R, \delta_R > 0$ such that

$$\left\| \frac{\mathrm{d}}{\mathrm{d}t} R_x(t\xi) \right\|_x \le \beta_R$$

for all $x \in \mathcal{M}$, all $\xi \in T_x\mathcal{M}$ with $\|\xi\|_x = 1$ and all $t < \delta_R$.

Local convergence

Lemma: $\sigma_k \to 0$

Let the conditions $(\mathrm{A}.3)$ and $(\mathrm{B}.1)$–$(\mathrm{B}.4)$ be satisfied. Then, all iterations are eventually very successful.

Theorem

Suppose that the conditions $(\mathrm{B}.1)$–$(\mathrm{B}.5)$ *are satisfied and let* $\alpha_0 = 1$ *and* $\rho \in (0, \frac{1}{2})$ *in Armijo rule. Then, the sequence* $\{x_k\}$ *converges q-superlinearly to* x_*.

Numerical results

Solver:

- GBB: Riemannian gradient method with BB step size
- AdaGBB: Riemannian Adaptive gradient method (J. Duchi et al. 2011) with BB step size
- ARNT (our): Adaptive regularized Newton method
- RTR: Riemannian trust-region method

Stopping criterion:

- norm of Riemannian gradient is less that 10^{-6}
- Maximal number of iteration is reached. (10000 for GBB and AdaGBB, 500 for ARNT and RTR)

Setting:

- We use GBB to find an initial point (norm of Riemannian gradient is less than 10^{-3} or 2000 iterations is reached) for ARNT and RTR

Low rank matrix estimation (D. Simon et al. 2010, Y. Gao et al. 2010)

- Given a correlation matrix C and a nonnegative symmetric weight matrix H, the low rank nearest correlation matrix problem is formulated as

$$\min_{X \in \mathbb{R}^{n \times n}} \frac{1}{2} \|H \odot (X - C)\|_F^2, \text{ s.t. } X_{ii} = 1,\ i = 1, \ldots, n,\ \operatorname{rank}(X) \leq p,\ X \succeq 0,$$

where $p \leq n$. By expressing $X = V^T V$ with $V = [V_1, \ldots, V_n] \in \mathbb{R}^{p \times n}$, problem can be converted into:

$$\min_{V \in \mathbb{R}^{p \times n}} \frac{1}{2} \|H \odot (V^T V - C)\|_F^2, \text{ s.t. } \|V_i\|_2 = 1,\ i = 1, \ldots, n.$$

- The matrix C is based on $100,000$ ratings for 1682 movies by 943 users from the Movielens data sets. The weight matrix H is provided by T. Fushiki at Institute of Statistical Mathematics, Japan.

Low rank nearest correlation matrix estimation

Table : Numerical results on low rank nearest correlation

	GBB			AdaGBB			ARNT			RTR		
p	its	nrmG	time	its	nrmG	time	its	nrmG	time	its	nrmG	time
5	10000	1.7e+2	196.5	4178	6.9e-7	41.3	260(8)	9.4e-7	38.1	500(12)	8.8e-2	78.4
10	10000	3.0e-4	207.4	4973	8.2e-7	103.8	347(12)	8.4e-7	58.9	500(17)	9.3e-2	102.5
20	10000	1.5e-4	198.3	5089	7.1e-7	86.6	237(24)	8.3e-7	63.4	500(23)	9.7e-2	152.0
50	10000	9.1e-5	288.1	3675	1.0e-6	90.2	34(58)	2.0e-7	38.1	63(82)	7.7e-7	80.2
100	10000	3.6e-4	181.6	10000	2.5e-6	258.0	26(118)	7.1e-7	50.1	19(428)	7.1e-7	120.4
150	10000	3.5e-2	124.2	10000	4.4e-5	241.7	35(134)	3.0e-7	76.1	18(688)	9.0e-7	173.2
200	10000	3.5e-2	153.7	10000	7.2e-5	245.3	37(130)	5.5e-7	78.4	16(758)	8.3e-7	162.0

By using a suitable discretization scheme, we can formulate a finite dimensional approximation to the continuous KS minimization problem as

$$\min_{X \in \mathbf{C}^{n\times p}} \quad f(X), \text{ s.t. } \quad X^* X = I,$$

where

$$f(X) := \frac{1}{4}\mathrm{tr}(X^* L X) + \frac{1}{2}\mathrm{tr}(X^* V_{ion} X) + \frac{1}{2}\sum_i \sum_l |x_i^* \omega_l| + \frac{1}{4}\rho^\top L^\dagger \rho + \frac{1}{2} e^\top \epsilon_{xc}(\rho),$$

$X = [x_1, \cdots, x_p] \in \mathbf{C}^{n\times p}$, $\rho(X) := \mathrm{diag}(XX^*)$, L is a finite dimensional Laplacian operator, V_{ion} corresponds to the ionic pseudopotentials, w_l represents a discretized pseudopotential reference projection function and ϵ_{xc} is related to the exchange correlation energy.

Table : Numerical results on KS total energy minimization.

solver	f	its	nrmG	time	f	its	nrmG	time
	ctube661				graphene16			
SCF	-1.3464e+02	16	3.1e-7	88.5	-9.4028e+01	101	5.8e-4	160.0
OptM	-1.3464e+02	101	7.2e-7	93.0	-9.4046e+01	187	8.5e-7	40.8
TRQH	-1.3464e+02	6(19)	3.2e-7	138.5	-9.4046e+01	8(19)	9.5e-7	70.3
ARNT	-1.3464e+02	3(11)	4.9e-7	78.3	-9.4046e+01	3(19)	8.6e-7	40.3
RTR	-1.3464e+02	3(11)	4.2e-7	78.2	-9.4046e+01	3(19)	7.3e-7	40.7
	graphene30				h2o			
SCF	-1.7358e+02	101	2.2e-3	860.6	-1.6441e+01	9	1.4e-7	1.8
OptM	-1.7360e+02	378	6.5e-7	517.0	-1.6441e+01	58	8.9e-7	2.0
TRQH	-1.7360e+02	12(38)	8.6e-7	783.9	-1.6441e+01	5(38)	8.4e-7	2.9
ARNT	-1.7360e+02	4(33)	2.5e-7	446.8	-1.6441e+01	3(11)	3.9e-7	1.8
RTR	-1.7360e+02	100(4)	2.3e-5	828.8	-1.6441e+01	3(11)	3.1e-7	2.1
	ptnio				qdot			
SCF	-2.2679e+02	66	7.7e-7	146.2	2.7702e+01	101	3.4e-2	22.3
OptM	-2.2679e+02	495	5.3e-7	145.6	2.7695e+01	2000	3.3e-6	70.8
TRQH	-2.2679e+02	23(39)	9.3e-7	286.0	2.7695e+01	91(39)	9.9e-7	115.8
ARNT	-2.2679e+02	4(52)	6.9e-7	132.4	2.7695e+01	27(65)	7.1e-7	64.5
RTR	-2.2679e+02	4(46)	8.5e-7	122.5	2.7695e+01	37(68)	4.0e-7	83.3

Bose-Einstein condensates (X. Wu et al. 2015)

The total energy in BEC is given as

$$E(\psi) = \int_{\mathbb{R}^d} \left[\frac{1}{2} |\nabla \psi(\mathbf{x})|^2 + V(\mathbf{x}) |\psi(\mathbf{x})|^2 + \frac{\beta}{2} |\psi(\mathbf{x})|^4 - \Omega \bar{\psi}(\mathbf{x}) L_z(\mathbf{x}) \right] d\mathbf{x},$$

where $\mathbf{x} \in \mathbb{R}^d$ is the spatial coordinate vector, $\bar{\psi}$ denotes the complex conjugate of ψ, $L_z = -i(x\partial - y\partial x)$, $V(x)$ is an external trapping potential. By using some proper discretizations, e.g., the finite difference, sine pseudospectral and Fourier pseudospectral (FP) method, we can formulate the BEC problem as

$$\begin{cases} \min\limits_{x \in \mathbf{C}^M} & f(x) := \frac{1}{2} x^* A x + \beta \sum\limits_{j=1}^{M} |x_j|^4, \\ \text{s.t.} & \|x\|_2 = 1, \end{cases}$$

where M is a positive integer, β is a given real constant, $A \in \mathbf{C}^{M \times M}$ is a Hermitian matrix.

Table : Numerical results on BEC with the potential function $V_1(x, y)$

solver	f	its	nrmG	time	f	its	nrmG	time
				β =500				
	$\Omega = 0.00$				$\Omega = 0.25$			
OptM	8.5118	58	6.6e-5	1.4	8.5106	103	9.7e-5	12.3
TRQH	8.5118	4(17)	1.5e-4	2.0	8.5106	5(22)	1.9e-4	21.9
ARNT	8.5118	3(24)	1.2e-5	1.5	8.5106	4(53)	1.6e-5	17.7
RTR	8.5118	3(25)	1.3e-5	1.5	8.5106	3(23)	6.0e-5	15.1
	$\Omega = 0.50$				$\Omega = 0.60$			
OptM	8.0246	276	9.0e-5	32.3	7.5890	301	1.0e-4	19.9
TRQH	8.0246	5(53)	2.0e-4	60.7	7.5890	5(60)	1.9e-4	35.4
ARNT	8.0197	3(62)	6.5e-5	21.3	7.5890	3(67)	5.7e-5	22.1
RTR	8.0246	11(113)	1.0e-4	56.5	7.5890	3(61)	5.2e-5	23.8
	$\Omega = 0.70$				$\Omega = 0.80$			
OptM	6.9731	340	1.0e-4	56.3	6.1016	386	1.0e-4	65.2
TRQH	6.9731	7(55)	2.0e-4	61.6	6.1016	5(64)	2.0e-4	83.1
ARNT	6.9731	10(99)	8.7e-5	44.4	6.1016	10(104)	8.7e-5	70.6
RTR	6.9731	99(118)	9.3e-5	234.2	6.1016	18(130)	7.7e-5	130.1
	$\Omega = 0.90$				$\Omega = 0.95$			
OptM	4.7784	10000	1.2e-3	243.6	3.7419	10000	7.4e-4	241.6
TRQH	4.7778	277(176)	2.0e-4	1090.9	3.7416	363(181)	2.0e-4	1185.1
ARNT	4.7777	147(132)	9.6e-5	413.3	3.7414	500(147)	2.6e-4	1204.0
RTR	4.7777	500(147)	8.5e-4	1250.4	3.7415	500(172)	9.7e-4	1419.0

Table : Numerical results on BEC with the potential function $V_2(x, y)$

solver	f	its	nrmG	time	f	its	nrmG	time
β =500								
	$\Omega = 0.00$				$\Omega = 0.25$			
OptM	9.3849	108	7.6e-5	2.8	9.3849	118	7.4e-5	5.6
TRQH	9.3849	4(21)	1.9e-4	2.6	9.3849	5(17)	1.5e-4	5.8
ARNT	9.3849	3(25)	5.5e-5	1.7	9.3849	3(26)	4.6e-5	3.6
RTR	9.3849	3(27)	5.5e-5	1.8	9.3849	3(27)	5.6e-5	3.7
	$\Omega = 0.50$				$\Omega = 0.60$			
OptM	9.2053	142	9.2e-5	30.2	9.1053	132	9.8e-5	25.3
TRQH	9.2053	5(23)	1.4e-4	24.4	9.1053	5(20)	1.5e-4	20.4
ARNT	9.2053	3(27)	8.4e-5	19.5	9.1053	3(28)	7.5e-5	11.5
RTR	9.2053	3(29)	8.3e-5	20.2	9.1053	3(30)	8.5e-5	19.5
	$\Omega = 0.70$				$\Omega = 0.80$			
OptM	8.8307	264	8.3e-5	26.0	8.4819	374	8.2e-5	46.9
TRQH	8.8307	5(47)	1.8e-4	24.0	8.4819	5(64)	2.0e-4	38.7
ARNT	8.8307	5(95)	3.8e-5	26.0	8.4819	3(76)	8.8e-5	30.2
RTR	8.8307	3(87)	7.6e-5	47.2	8.4819	3(94)	7.4e-5	26.4
	$\Omega = 0.90$				$\Omega = 0.95$			
OptM	8.0659	426	1.0e-4	75.2	7.7455	9508	9.9e-5	244.4
TRQH	8.0659	5(94)	1.4e-4	124.1	7.7455	21(155)	2.0e-4	254.8
ARNT	8.0659	3(99)	9.0e-5	56.4	7.7455	30(192)	2.3e-5	171.7
RTR	8.0659	3(108)	8.7e-5	107.6	7.7455	20(270)	9.7e-5	257.4

Low-rank matrix completion (B. Vandereycken 2013)

- Given a partially observed matrix $A \in \mathbb{R}^{m\times n}$, one wants to find the lowest-rank matrix to fit A on the known elements. It can be formulated as

$$\begin{cases} \min\limits_{X\in\mathbb{R}^{m\times n}} & f(X) := \frac{1}{2}\|P_\Omega(X) - A\|_F^2, \\ \text{s.t.} & X \in \mathcal{M}_k := \{X \in \mathbb{R}^{m\times n} : \mathrm{rank}(X) = k\}, \end{cases}$$

  where $P_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}, X_{i,j} \to \begin{cases} X_{i,j} & \text{if } (i,j) \in \Omega, \\ 0 & \text{if } (i,j) \notin \Omega \end{cases}$ is the projection onto Ω and Ω is a subset of $\{1,\ldots,m\} \times \{1,\ldots,n\}$.

- In numerical tests, we take two Gaussian random matrices $A_L, A_R \in \mathbb{R}^{n\times k}$, then uniformly sample the indices set Ω for a given cardinality and set the matrix $A := P_\Omega(A_L A_R^\top)$. Since the degrees of freedom in a nonsymmetric matrix of rank k is $k(2n-k)$, we define the ratio $r_S = \frac{|\Omega|}{k(2n-k)}$.

Table : Numerical results on low rank matrix completion with the but different n.

	GBB			ARNT			RTR		
	$k = 10, r_S = 0.8$								
n	its	nrmG	time	its	nrmG	time	its	nrmG	time
1000	603	5.1e-7	12.5	6(84)	3.4e-7	7.7	8(91)	6.6e-7	8.2
2000	570	9.2e-7	43.9	5(72)	8.9e-7	23.6	8(86)	6.2e-7	28.2
4000	671	9.7e-7	179.8	6(82)	4.6e-7	94.8	9(85)	2.0e-7	104.8
8000	666	9.8e-7	694.2	5(104)	5.2e-7	320.1	8(130)	5.4e-7	394.5
	$n = 4000, r_S = 0.95$								
k	its	nrmG	time	its	nrmG	time	its	nrmG	time
10	5252	1.0e-6	1415.9	13(133)	7.4e-7	392.1	12(236)	4.4e-7	438.2
20	2126	1.0e-6	600.8	7(125)	3.9e-7	269.5	9(195)	2.3e-7	315.9
30	1488	1.0e-6	438.8	6(132)	3.1e-7	255.2	9(214)	2.6e-7	329.9
40	1010	9.3e-7	311.4	5(103)	1.1e-7	220.5	5(103)	1.1e-7	219.4
50	1494	7.9e-7	477.1	4(103)	1.5e-7	273.8	4(103)	1.6e-7	272.5
60	1398	9.9e-7	480.4	4(110)	5.7e-7	313.3	4(114)	5.7e-7	315.2
	$n = 8000, k = 10$								
r_S	its	nrmG	time	its	nrmG	time	its	nrmG	time
0.1	86	3.7e-7	88.1	3(11)	4.4e-7	54.7	3(11)	4.3e-7	53.2
0.2	89	8.6e-7	93.9	3(14)	3.4e-7	55.5	3(14)	3.4e-7	54.0
0.5	173	8.5e-7	178.8	3(18)	7.0e-7	111.4	3(18)	7.0e-7	109.7
0.8	666	9.8e-7	700.2	5(104)	5.2e-7	318.7	8(130)	5.4e-7	388.8

Many Thanks For Your Attention!