An Implicit Riemannian Trust-Region Method for the Symmetric Generalized Eigenvalue Problem

Christopher G. Baker [1,2]

Pierre-Antoine Absil [3,4]

Kyle A. Gallivan [2]

[1] *Computational Mathematics and Algorithms, Sandia National Laboratories*

[2] *School of Computational Science, Florida State University*

[3] *Département d'ingénierie mathématique, Université Catholique de Louvain*

[4] *Peterhouse, University of Cambridge*

International Conference on Computational Science 2006

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

Outline

- Symmetric Generalized Eigenvalue Problem
- Riemannian Trust-Region Method
- Implicit Riemannian Trust-Region Method
- Results

Symmetric Generalized Eigenvalue Problem

Given: $n \times n$ pencil (A, B), $A = A^T$, $B = B^T \succ 0$.

Eigenvalue $\lambda_i \in \mathbb{R}$, eigenvector $v_i \in \mathbb{R}^n$ satisfy

$$Av_i = Bv_i\lambda_i, \quad i = 1, \ldots, n$$

$$\lambda_i \in \mathbb{R}, \quad v_i \in \mathbb{R}^n$$

$$\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$

Problem: Compute the eigenvectors associated with the leftmost eigenvalues:

$$(v_i, \lambda_i), \quad i = 1, \ldots, p$$

$\boxed{\text{The Optimization Problem}}$

$V = \begin{bmatrix} v_1 & \ldots & v_p \end{bmatrix}$ is a minimizer of the generalized Rayleigh quotient:

$$f(Y) = \operatorname{trace}\left((Y^T BY)^{-1} Y^T AY\right).$$

This function depends only on subspace: $f(Y) = f(YM)$ for any invertible M.

$\Rightarrow$ Search for leftmost p eigenpairs via optimization over the set of p-dimensional subspaces of $\mathbb{R}^n$: the Grassmann manifold.

One method for this is the Riemannian Trust-Region (RTR) method.

Brief Intro to RTR

The Riemannian Trust-Region method [ABG06a, ABG06b]:

- Adapts trust-region ideas from Euclidean spaces to Riemannian manifolds;
- Preserves strong global convergence properties;
- Retains fast local convergence;
- Providing inverse-free, low-memory methods of optimization.

Trust-region Methods on Riemannian Manifolds

1. Given: smooth manifold M; Riemannian metric g; smooth cost function f on M; retraction R from the tangent bundle TM to M; current iterate x_k.

1b. Lift up the cost function to the tangent space T_xM:

$$\hat{f}_x = f \circ R_x.$$

2. Build a model $m_k(s)$ of $\hat{f}_{x_k}$ around 0.
3. Find (up to some precision) a minimizer s_k of the model within a "trust-region", i.e., a ball of radius Δ_k around x_k.

Trust-Region Methods on Riemannian Manifolds (cont'd)

4. Compute the ratio

$$\rho_k = \frac{f(x_k) - f(R_{x_k} s_k)}{m_k(0) - m_k(s_k)}$$

   to compare the actual value of the cost function at the proposed new iterate with the value predicted by the model.

5. Shrink, enlarge or keep the trust-region radius according to the value of ρ_k.
6. Accept or reject the proposed new iterate $R_{x_k} s_k$ according to the value of ρ_k.
7. Increment k and go to step 2.

Required Ingredients for Riemannian TR

- Manifold M, Riemannian metric g, and cost function f on M.
- Practical expression for $T_{x_k}M$.
- Retraction $R_{x_k} : T_{x_k}M \to M$.
- Function $\hat{f}_{x_k}(s) := f(R_{x_k}s)$.
- Gradient $\operatorname{grad} \hat{f}_{x_k}(0)$.
- Hessian $\operatorname{Hess} \hat{f}_{x_k}(0)$.

Trust-Region for Extreme SGEVP: Principles

Ingredients of the RTR method for ESGEVP [ABG06a]:

1. Manifold: $M = \{p - \text{dimensional subspaces of } \mathbb{R}^n\}$
2. $\mathcal{Y}$ represented by any $Y \in \mathbb{R}^{n\times p} : Y^T Y = I, \mathrm{col}(Y) = \mathcal{Y}$.
3. Tangent space: $T_Y M = \{Z \in \mathbb{R}^{n\times p} : Y^T B Z = 0\}$.
4. Metric: $g_Y(Z_a, Z_b) = \mathrm{trace}\left(Z_a^T Z_b\right)$.
5. Retraction: $R_Y Z = (Y + Z)M$
6. Cost function: $f(Y) = \mathrm{trace}\left((Y^T B Y)^{-1}(Y^T A Y)\right)$.

Trust-Region for Extreme SGEVP: Details

Lifted cost function:

$$\hat{f}_Y(Z) = f(R_Y Z) = \text{trace}\left(\left((Y+Z)^T B(Y+Z)\right)^{-1}(Y+Z)^T A(Y+Z)\right)$$
$$= \text{trace}\left(Y^T AY\right) + 2\text{trace}\left(Z^T AY\right) + \text{trace}\left(Z^T (AZ - BZY^T AY)\right) + HOT$$

The second order approximation of $\hat{f}_Y(Z)$ is

$$m_Y(Z) = f(Y) + g_Y(\text{grad}\, f(Y), Z) + \frac{1}{2} g_Y(\mathcal{H}_Y Z, Z)$$
$$= \text{trace}\left(Y^T AY\right) + 2\text{trace}\left(Z^T AY\right) + \text{trace}\left(Z^T\left(AZ - BZY^T AY\right)\right).$$

Compute an approximate minimizer $\tilde{Z}$ using truncated CG [CGT00]. Update: $Y_+ = R_Y \tilde{Z} = (Y + \tilde{Z})M$.

Complaints Against Trust-Region Methods

- Trust-region radius is heuristic.
  - Radius of current trust-region based on performance of last model minimization.
  - This may constrain current model minimization.
- Iterate may be rejected.
  - Wasted time spent computing potential iterate.
  - It can take a number of outer iterations to adjust trust-region radius.
- Inner iteration may run too long on the last iteration
  - As soon as outer/global stopping criterion is realized, iteration should be stopped.

Proposal for New Trust-Region

Idea: Base trust-region on the current performance of m_x.

Old trust-region was

$$\{s \in T_x M : \|s\| \le \Delta_k\}, \Delta_k > 0 \ .$$

New trust-region is

$$\{s \in T_x M : \rho_x(s) \ge \rho'\}, \quad \rho' > 0 \ .$$

ρ_x is as before:

$$\rho_x(s) = \frac{f(x) - f(R_x s)}{m_x(0) - m_x(s)}.$$

Implicit Riemannian Trust-Region (IRTR)

1. Given: smooth manifold M; Riemannian metric g; smooth cost function f on M; retraction R from the tangent bundle TM to M; current iterate x_k.

1b. Lift up the cost function to the tangent space $T_x M$:

$$\hat{f}_x = f \circ R_x.$$

2. Build a model $m_k(s)$ of $\hat{f}_{x_k}$ around 0.
3. Find (approximately) a minimizer s_k of the model within the new trust-region.
4. Accept $x_{k+1} = R_{x_k} s_k$.
5. Increment k and go to step 2.

Solving Model Minimization in IRTR

- Use truncated CG to solve model minimization.
- New trust-region definition requires some modifications.
- Boundary test:
  - Before: check $\|s^j\| \leq \Delta_k$
  - Now: check $\rho_{x_k}(s^j) \geq \rho'$
- If $\rho_{x_k}(s^j) < \rho'$:
  - Before: compute τ such that $\|s^{j-1} + \tau\delta_j\| = \Delta_k$
  - Now: Compute τ such that $\rho_{x_k}(s^{j-1} + \tau\delta_j) = \rho'$
  - This is potentially much more difficult.

Required Ingredients for Implicit RTR

- Manifold M, Riemannian metric g, and cost function f on M
- Practical expression for $T_{x_k}M$
- Retraction $R_{x_k} : T_{x_k}M \to M$
- Function $\hat{f}_{x_k}(s) := f(R_{x_k}s)$
- Gradient $\operatorname{grad} \hat{f}_{x_k}(0)$
- Hessian $\operatorname{Hess} \hat{f}_{x_k}(0)$
- Trust-region test: $\rho_{x_k}(s)$
- Trust-region search: find τ s.t. $\rho_{x_k}(s + \tau\delta) = \rho'$

Convergence Results of IRTR

- The trust-region definition is very strong.
- As a result, standard TR global convergence results follow easily.
- Global Convergence of IRTR for ESGEVP: Let $\{y_k\}$ be a sequence of iterates produced via IRTR-tCG with $\rho' \in (0,1)$. Then

$$\lim_{k\to\infty} \|\mathrm{grad} f(y_k)\| = 0.$$

- The local convergence theory should be easily adaptable from RTR to IRTR.

$\boxed{\text{Extreme SGEVP: } p = 1}$

If $p = 1$, then $\rho_y(s) = \frac{\hat{f}_y(0) - \hat{f}_y(s)}{m_y(0) - m_y(s)} = \frac{1}{1 + s^T B s}$.

- Checking trust-region inclusion requires checking $\|s\|_B$
- Solving ρ_y along a tangent vector has an analytical solution: τ s.t. $\rho_y(s + \tau\delta) = \rho'$ given by

$$\begin{aligned} \tau &= \frac{-\delta^T B s + \sqrt{(\delta^T B s)^2 + \delta^T B \delta (\Delta_{\rho'}^2 - s^T B s)}}{\delta^T B \delta} \\ \Delta_{\rho'} &= \sqrt{\frac{1}{\rho'} - 1} \end{aligned}$$

- IRTR for $p = 1$ ESGEVP is straightforward.

$\boxed{\text{Case } p > 1}$

$$\rho_Y(S) = \frac{\operatorname{trace}\left((I + S^T BS)^{-1}(S^T BS(Y^T AY) - 2S^T AY - S^T AS)\right)}{\operatorname{trace}\left(S^T BS(Y^T AY) - 2S^T AY - S^T AS\right)}$$

Assume that $Y^T BY = I$ and $Y^T AY = \Sigma$. Then

$$\begin{aligned} m_Y(S) &= \operatorname{trace}\left(Y^T AY + 2S^T AY + S^T(AS - BSY^T AY)\right) \\ &= \sum_{i=1}^{p} \left(\sigma_i + 2s_i^T Ay_i + s_i^T(As_i - Bs_i\sigma_i)\right) \\ &= \sum_{i=1}^{p} m_{y_i}(s_i). \end{aligned}$$

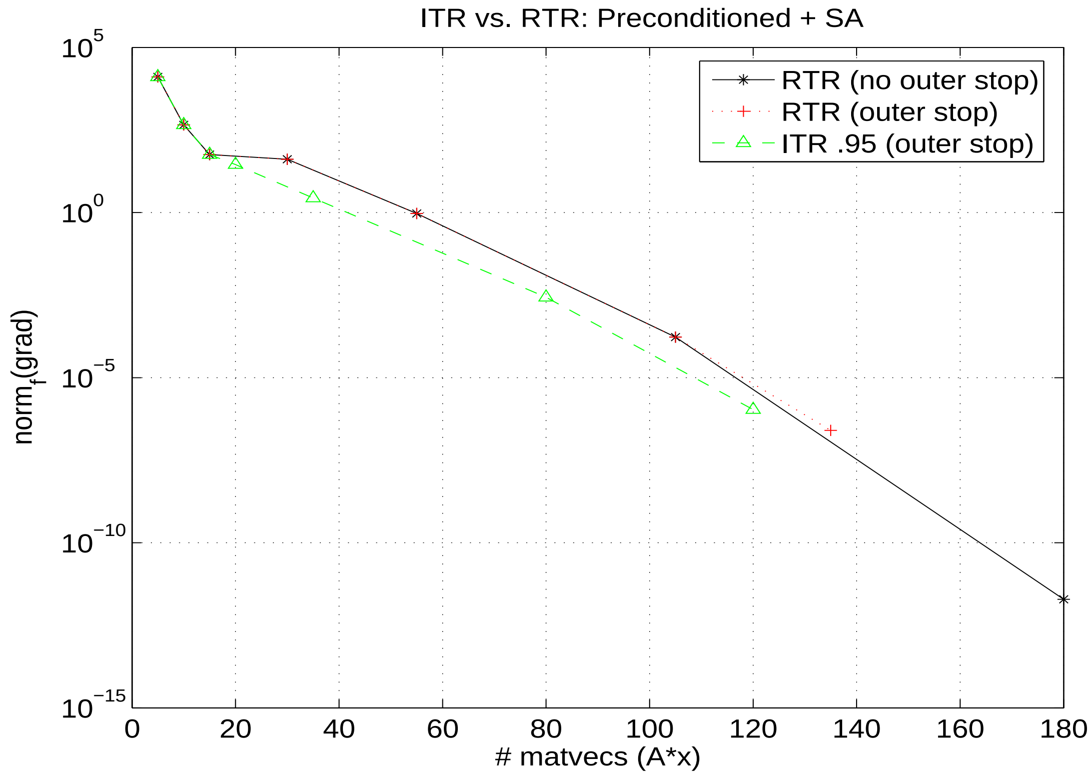
$\boxed{\text{Case } p > 1}$

- The $p > 1$ model $m_Y(S)$ can be decoupled into p "scalar" models, for which we have a formula for ρ.
- The block algorithm runs p simultaneous tCG algorithms.
- All processes are stopped if any satisfies a stopping criterion.
- Global convergence is still guaranteed.
- But $\rho_Y(S) \not\geq \rho'$: not a true IRTR!

Outer Criterion Monitoring

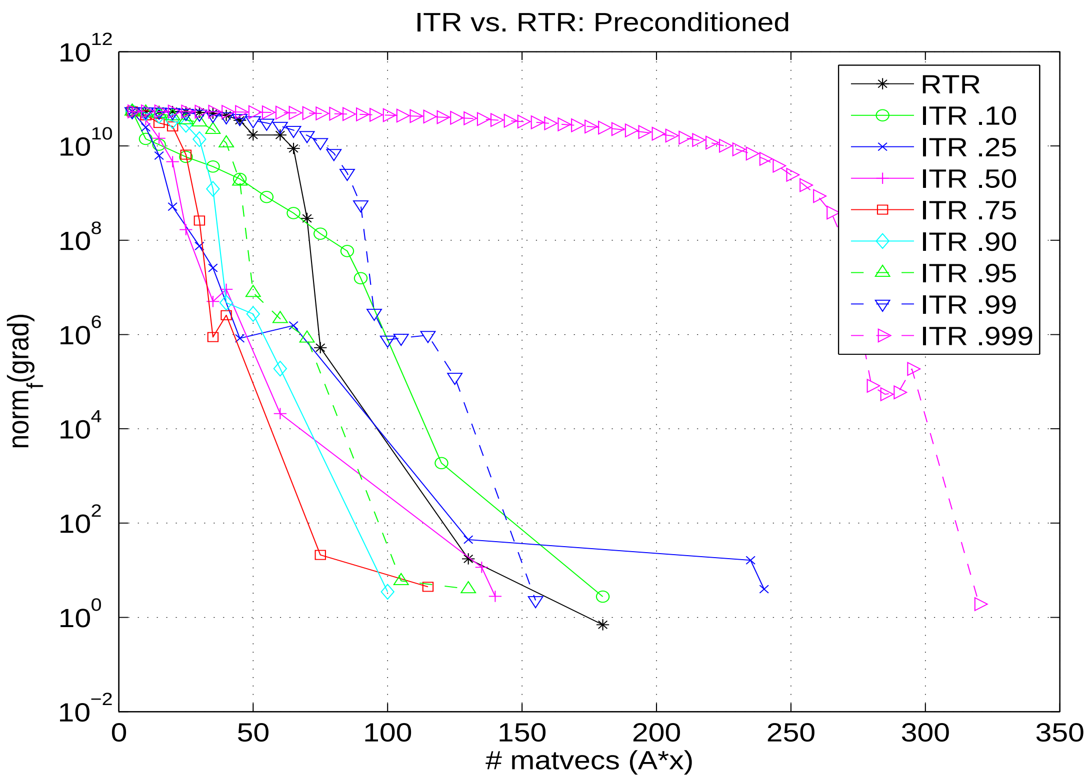
- Last call to tCG often performs more work than necessary to satisfy outer stopping criterion.
- Problem is typical for methods employing an inner iteration.
- Solution is (occasionally) compute outer residual in inner iteration, check stopping criterion.
- Similar to suggestion in [Not02], except we have no efficient formula for the residual norm.

EXP: Monitoring Outer Stopping Criterion



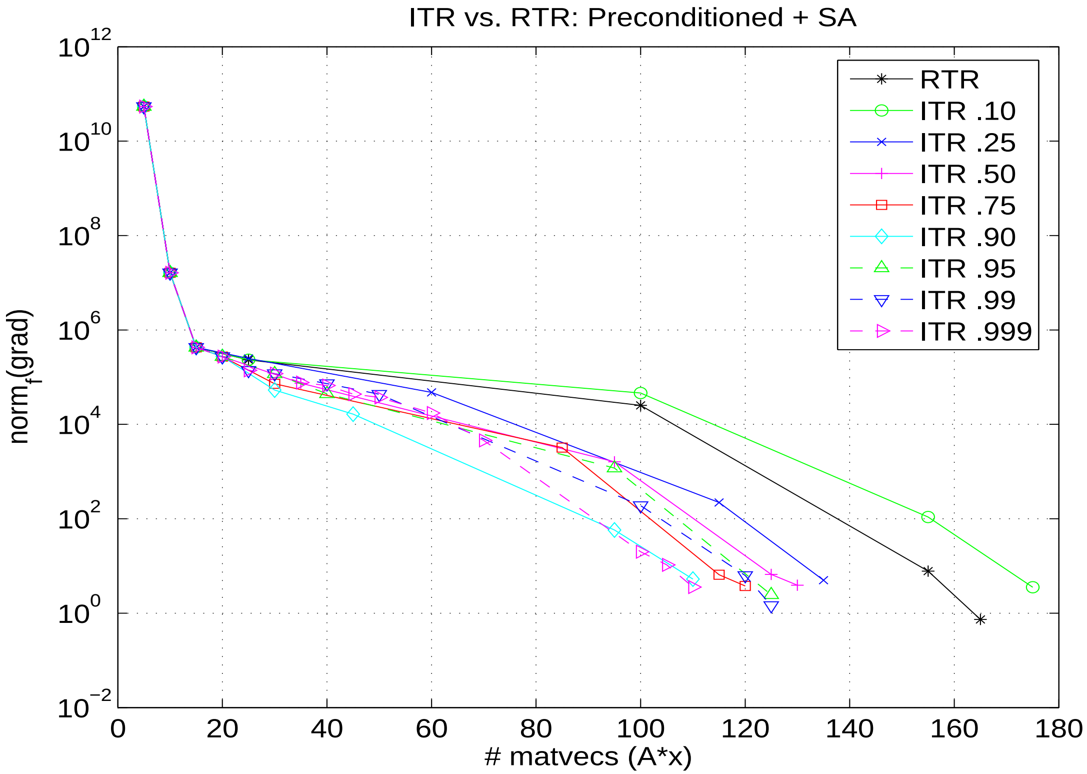
2-D Laplacian, $n = 10000$; precond. using exact factorization of A after `symamd`; 10-D subspace acceleration; $p = 5$

EXP: IRTR vs. RTR



BCSST24; precond. with exact factorization of A after `symamd`; no subspace acceleration; $p = 5$

EXP: IRTR vs. RTR



BCSST24; precond. with exact factorization of A after `symamd`; 10-D subspace acceleration; $p = 5$

Summary

- Take-home idea: Break down the barrier between inner and outer iteration:
  - Outer criterion monitoring stops when iteration is ultimately satisfied; always maintain awareness of outer error
  - Base trust-region on the performance of surrogate model; always maintain awareness of cost function
- Result: Globally convergent, block eigensolver with superlinear local convergence; more efficient than the RTR.

References

[ABG06a] P.-A. Absil, C. G. Baker, and K. A. Gallivan, *A truncated-CG style method for symmetric generalized eigenvalue problems*, J. Comput. Appl. Math. **189** (2006), no. 1–2, 274–285.

[ABG06b] P.-A Absil, C. G. Baker, and K. A. Gallivan, *Trust-region methods on Riemannian manifolds*, to be published in Foundations of Computational Mathematics.

[CGT00] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-region methods*, MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, and Mathematical Programming Society (MPS), Philadelphia, PA, 2000.

[Not02] Y. Notay, *Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem*, Numer. Linear Algebra Appl. **9** (2002), no. 1, 21–44.

THE END

Algorithm 1 (Prec. Truncated CG for IRTR)

Set $s^0 = 0$, $r_0 = \mathrm{grad}\hat{f}_y$, $z_0 = M^{-1}r_0$, $d^0 = -z_0$

for $j = 0, 1, 2, \ldots$

Check inner stopping criterion

Check $\delta_j^T H_y[\delta_j]$

Compute $\tau \geq 0$ *s.t.* $s = s^j + \tau\delta_j$ *satisfies* $\rho_y(s) = \rho'$; ***return*** s

Set $\alpha^j = (z_j^T r_j)/(\delta_j^T H_y[\delta_j])$

Set $s^{j+1} = s^j + \alpha_j \delta_j$

if $\rho_y(s^{j+1}) < \rho'$

Compute $\tau \geq 0$ *s.t.* $s = s^j + \tau\delta_j$ *satisfies* $\rho_y(s) = \rho'$; ***return*** s

Check outer stopping criterion

Set $r_{j+1} = r_j + \alpha^j H_y[\delta_j]$

Set $z_{j+1} = M^{-1} r_{j+1}$

Set $\beta^{j+1} = (z_{j+1}^T r_{j+1})/(z_j^T r_j)$

Set $\delta_{j+1} = -z_{j+1} + \beta^{j+1}\delta_j$

end.