Trust-region methods on Riemannian manifolds with applications in numerical linear algebra

Pierre-Antoine Absil

Christopher G. Baker

Kyle A. Gallivan

School of Computational Science and Information Technology,

Florida State University

MTNS 2004

Friday, 9 July 2004

The Problem : Minor Eigenvector Computation

Given $n \times n$ matrix $A = A^T$ with (unknown) eigen-decomposition

$$A\,[v_1|\ldots|v_n] = [v_1|\ldots|v_n]\,\mathrm{diag}(\lambda_1,\ldots,\lambda_n)$$

$$[v_1|\ldots|v_n]^T\,[v_1|\ldots|v_n] = I, \quad 0 < \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n.$$

The problem is to compute the minor eigenvector $\pm v_1$.

Simple vector iterations: Inverse Iteration

$$y_{k+1} = \frac{A^{-1}y_k}{\|A^{-1}y_k\|}$$

Properties:

- Global convergence to $\{\pm v_1, \ldots, \pm v_n\}$.
- Stable convergence to $\pm v_1$ only.
- Local linear convergence, with ratio $\frac{\lambda_1}{\lambda_2}$.

  Exemple: $n = 100$, $\lambda_i = i/n$ (evenly spaced eigenvalues on $(0, 1]$). Then $\frac{\lambda_1}{\lambda_2} = 0.5$.

  Possible evolution: error(1)=0.1, error(2)=0.05, error(3)=0.0025,...,error(27)$\simeq 1.4 \cdot 10^{-9}$.
- Computing a new iterate is expensive.

Simple vector iterations: Rayleigh Quotient Iteration (RQI)

$$\rho_k = \frac{y_k^T A y_k}{y_k^T y_k}$$

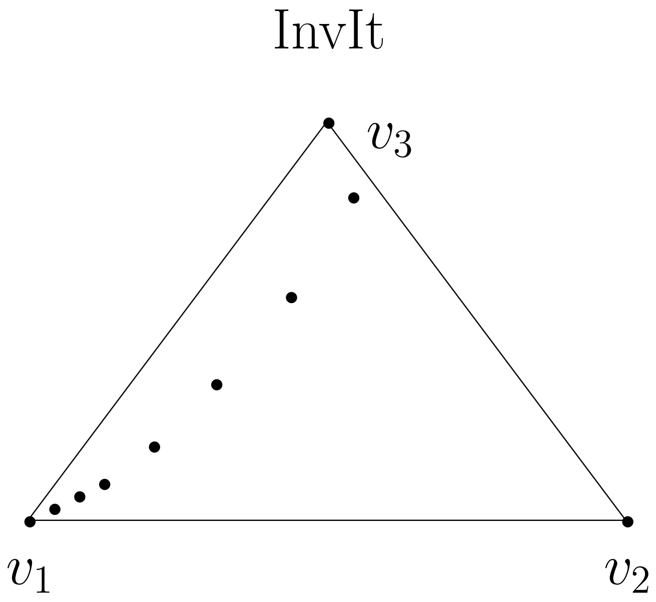
$$y_{k+1} = \frac{(A - \rho_k I)^{-1} y_k}{\|(A - \rho_k I)^{-1} y_k\|}$$

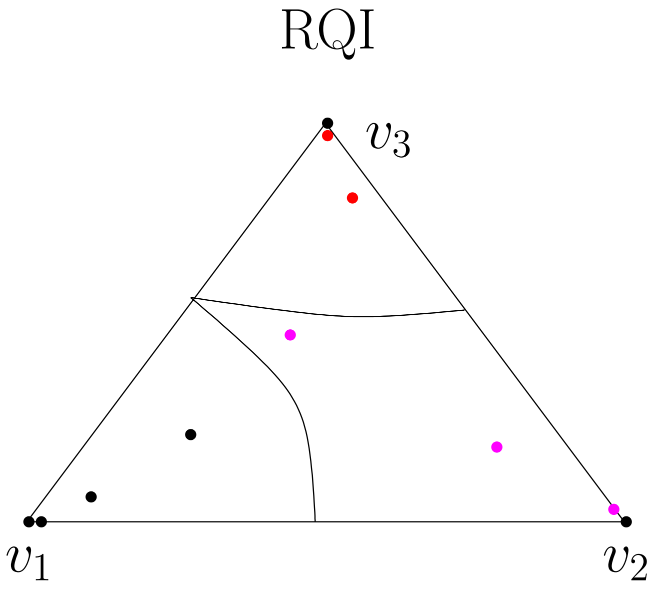
Properties:

- Converges to “nearest” eigenvector.
- Cubic local convergence.

  Possible evolution: error(1)=0.1, error(2)= 10^{-3}, error(3)= 10^{-9}.
- Computing a new iterate is expensive.

Simple vector iterations : illustrations





The ideal minor component algorithm

Given $A = A^T \succ 0$, with eigenvalues $0 < \lambda_1 \leq \ldots \lambda_n$ and associated eigenvectors $v_1, \ldots, v_n$.

1. Convergence to some eigenvector for all initial conditions.
2. Stable convergence to the minor eigenvector $\pm v_1$ only.
3. Superlinear (cubic) local convergence to $\pm v_1$.

The ideal minor component algorithm

Given $A = A^T \succ 0$, with eigenvalues $0 < \lambda_1 \leq \ldots \lambda_n$ and associated eigenvectors $v_1, \ldots, v_n$.

1. Convergence to some eigenvector for all initial conditions.
2. Stable convergence to the minor eigenvector $\pm v_1$ only.
3. Superlinear (cubic) local convergence to $\pm v_1$.
4. No factorization of A.

   Matrix A only utilized as operator $x \mapsto Ax$.

The ideal minor component algorithm

Given $A = A^T \succ 0$, with eigenvalues $0 < \lambda_1 \leq \ldots \lambda_n$ and associated eigenvectors $v_1, \ldots, v_n$.

1. Convergence to some eigenvector for all initial conditions.
2. Stable convergence to the minor eigenvector $\pm v_1$ only.
3. Superlinear (cubic) local convergence to $\pm v_1$.
4. No factorization of A.

   Matrix A only utilized as operator $x \mapsto Ax$.
5. Minimal storage space required.

Approach: Optimization on Manifolds

Rayleigh quotient cost function:

$$f : S^{n-1} \to \mathbb{R} : y \mapsto y^T A y,$$

where S^{n-1} is the unit sphere $\{y \in \mathbb{R}^n : y^T y = 1\}$.

Useful properties:

- The stationary points of f are the eigenvectors of A.
- The local (and global) minima of f are $\pm v_1$.

The ideal minor component algorithm

Given $A = A^T \succ 0$, with eigenvalues $0 < \lambda_1 \leq \ldots \lambda_n$ and associated eigenvectors $v_1, \ldots, v_n$.

1. Convergence to some eigenvector for all initial conditions.
2. Stable convergence to the minor eigenvector $\pm v_1$ only.
3. Superlinear (cubic) local convergence to $\pm v_1$.

The ideal minor component algorithm

Given $A = A^T \succ 0$, with eigenvalues $0 < \lambda_1 \leq \ldots \lambda_n$ and associated eigenvectors $v_1, \ldots, v_n$.

1. Convergence to some eigenvector for all initial conditions.
2. Stable convergence to the minor eigenvector $\pm v_1$ only.
3. Superlinear (cubic) local convergence to $\pm v_1$.
4. No factorization of A.

   Matrix A only utilized as operator $x \mapsto Ax$.

The ideal minor component algorithm

Given $A = A^T \succ 0$, with eigenvalues $0 < \lambda_1 \leq \ldots \lambda_n$ and associated eigenvectors $v_1, \ldots, v_n$.

1. Convergence to some eigenvector for all initial conditions.
2. Stable convergence to the minor eigenvector $\pm v_1$ only.
3. Superlinear (cubic) local convergence to $\pm v_1$.
4. No factorization of A.

   Matrix A only utilized as operator $x \mapsto Ax$.
5. Minimal storage space required.

Suitable optimization method?

To achieve the “ideal minor component method”, we need an optimization method with the following properties:

1. Global convergence to stationary points.
2. Stable convergence to local minima only.
3. Superlinear local convergence.
4. No factorization of the Hessian.
5. Minimal storage space needed.

$\boxed{\text{Yes!}}$

TRUST-REGION METHOD

where the trust-region subproblems are solved with a

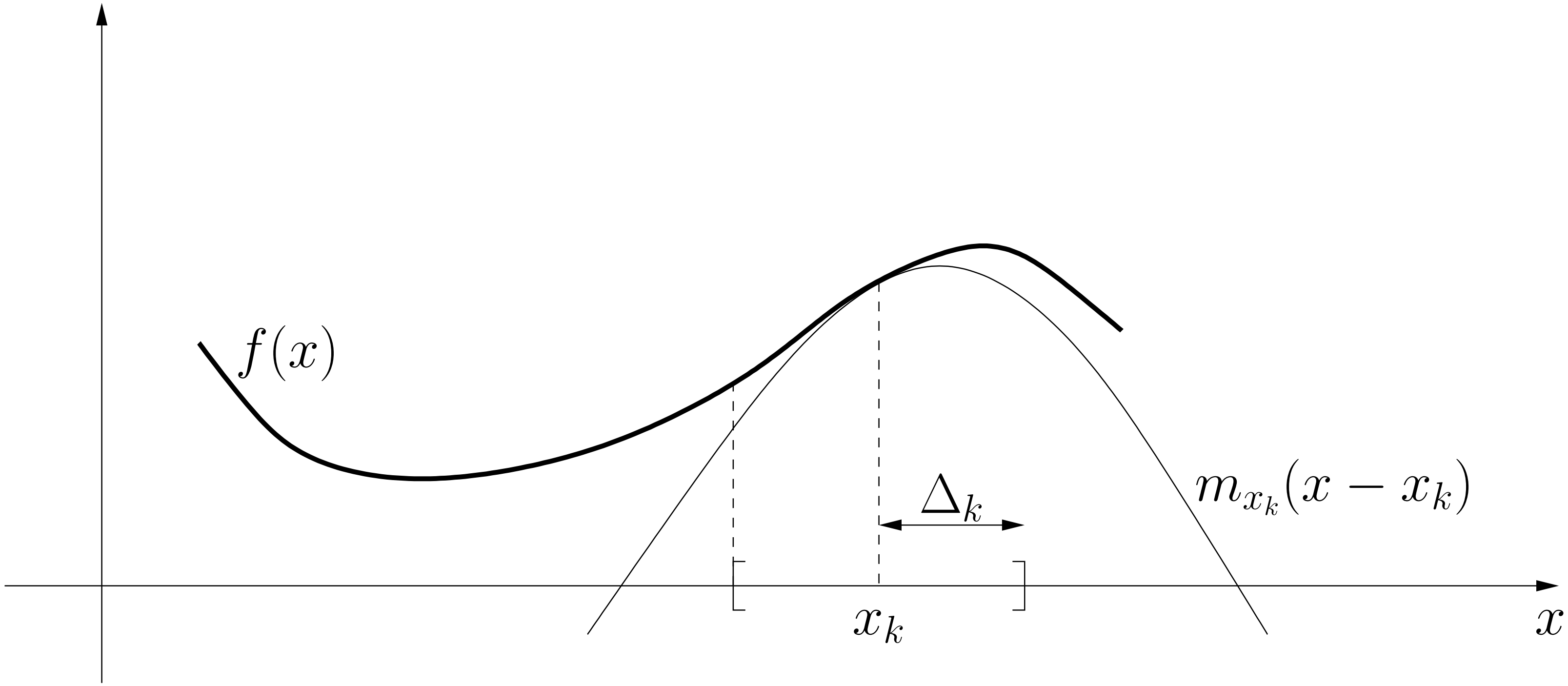
TRUNCATED CONJUGATE-GRADIENT

algorithm.

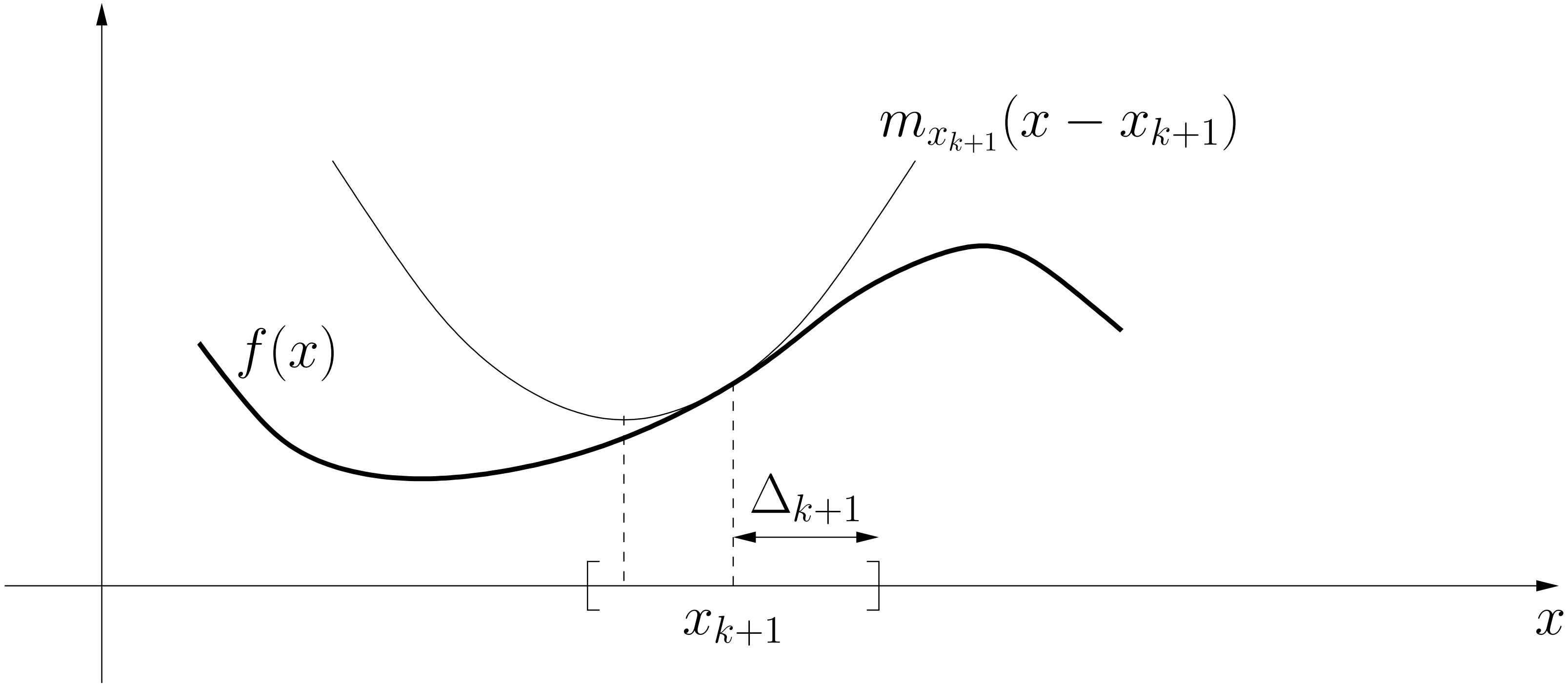
Outline

- Trust-region (TR) with truncated CG (tCG) in $\mathbb{R}^n$.
- TR-tCG on the sphere.
- Extensions, comparisons, numerical experiments.

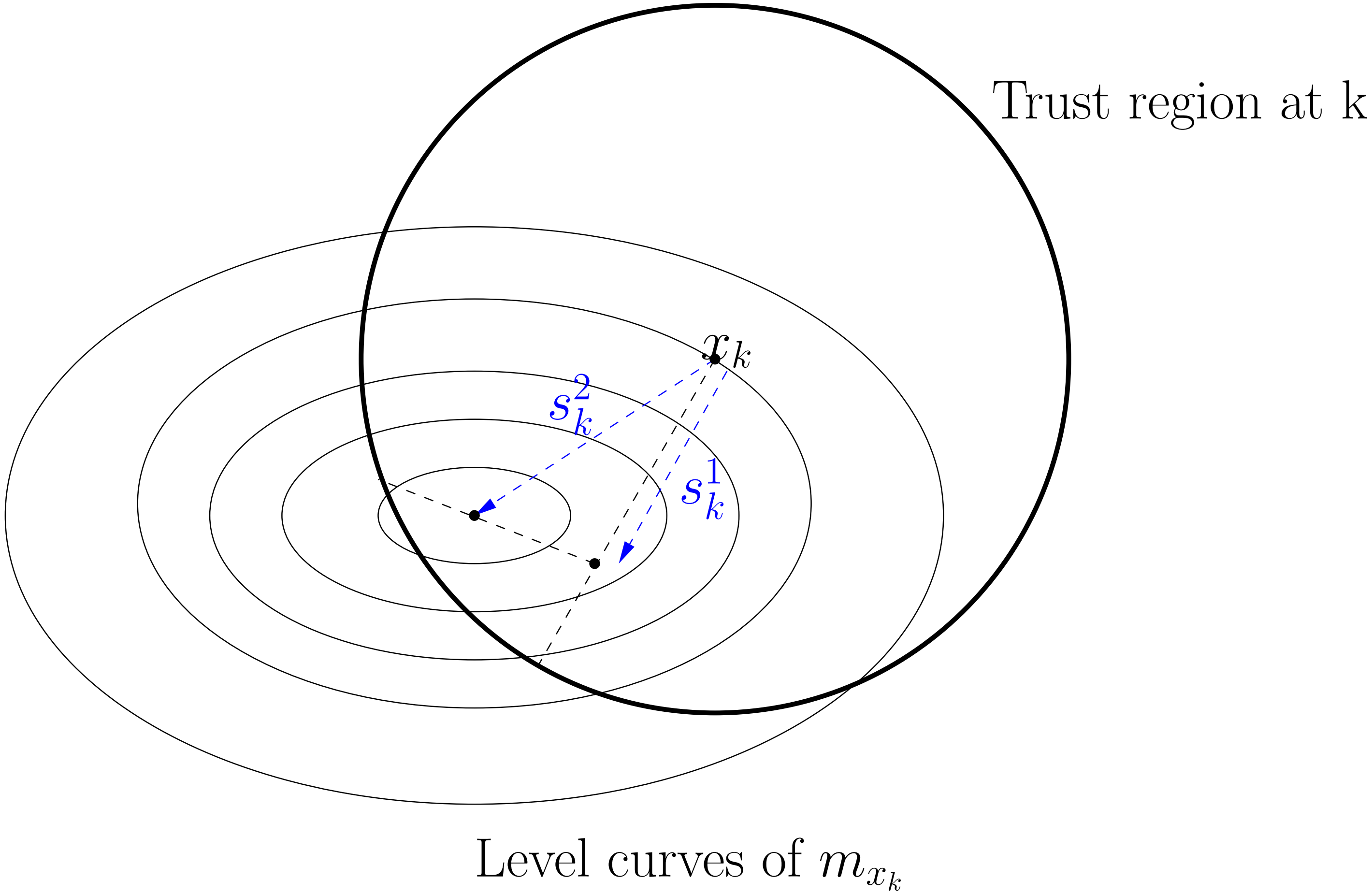
Principle of Trust-Region (TR)



Principle of Trust-Region (TR)



Principle of truncated CG (tCG)



Stopping criterion for tCG

Reasons for stopping tCG (inner iteration):

- The line-search algorithm hits the trust-region boundary. (This happens in particular when the model has a negative curvature along the current direction of search.)
- The norm of the residual has become sufficiently small. Criterion:

$$\|r_j\| \leq \|r_0\| \min(\|r_0\|^\theta, \kappa).$$

  Note that $r_n = 0$ in exact arithmetic (theory of linear CG).

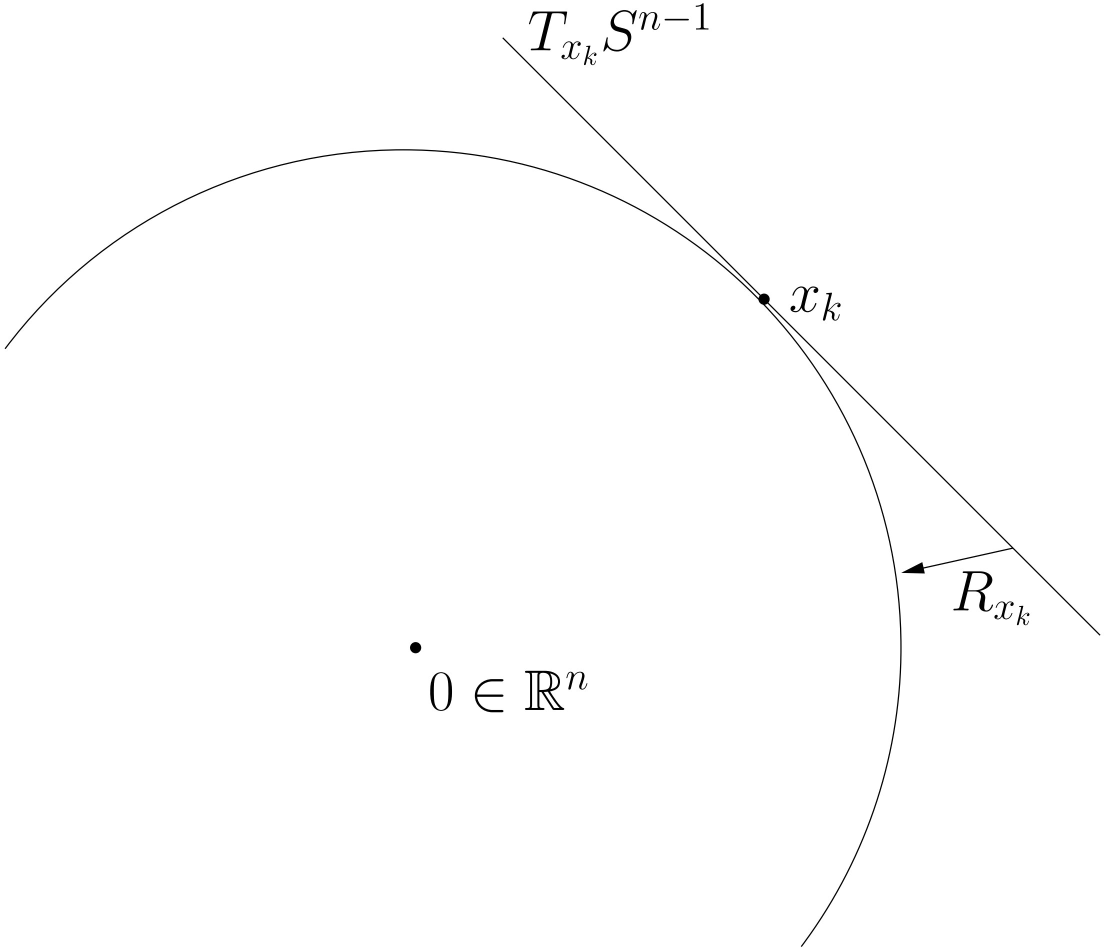
$\longrightarrow$ Expected order of convergence:

$$\min\{\theta + 1, 3\}.$$

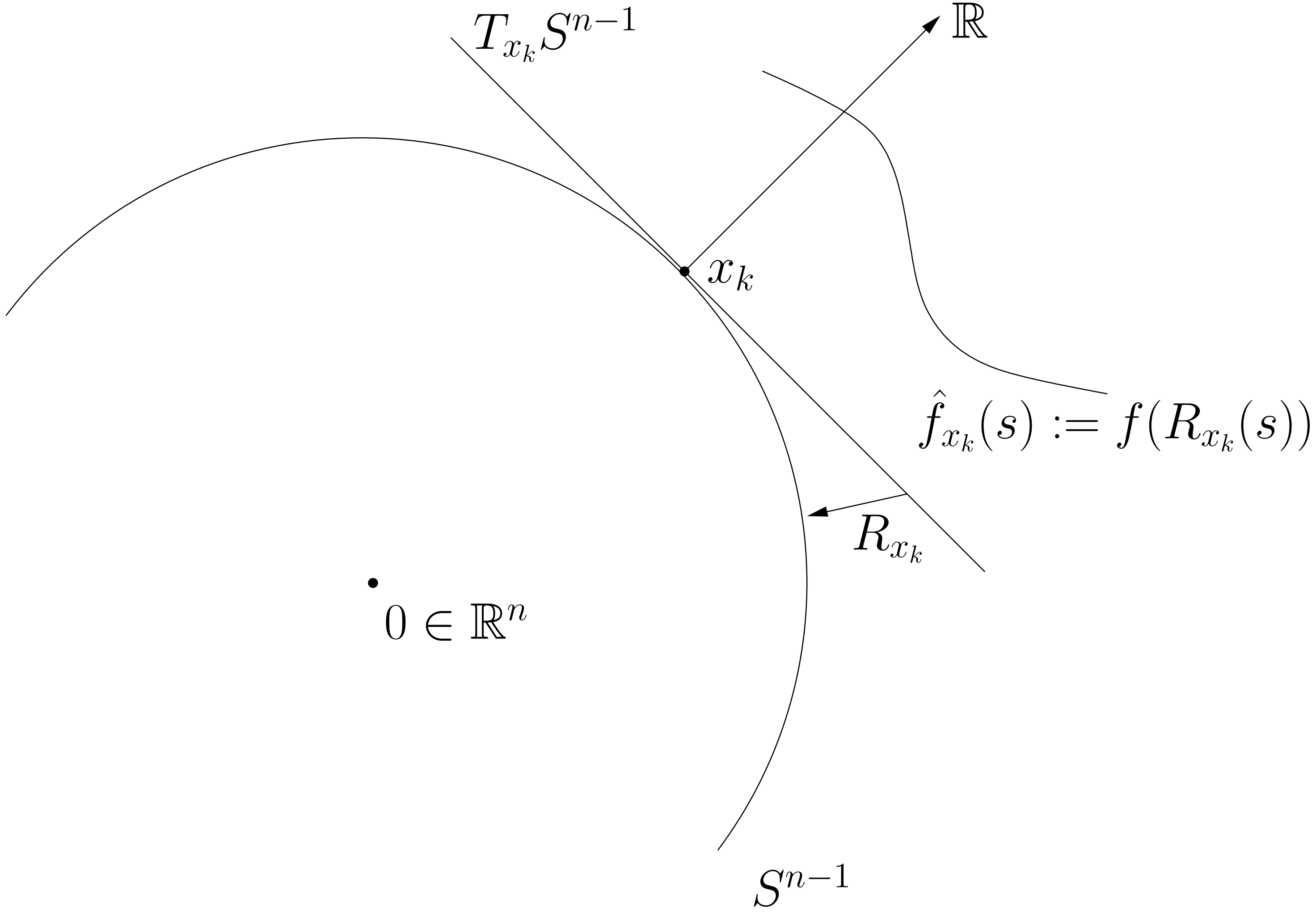
Outline

- Trust-region (TR) with truncated CG (tCG) in $\mathbb{R}^n$.
- TR-tCG on the sphere.
- Extensions, comparisons, numerical experiments.

Trust-region on the sphere



Trust-region on the sphere



Properties of the algorithm

Algorithm: Trust-region method on the sphere with truncated-CG algorithm for minimizing the Rayleigh quotient.

Properties:

1. For all initial conditions, $\{y_k\}$ converges to an eigenvector.
2. Only the minor eigenvector $\pm v_1$ is stable.
3. Superlinear rate, with exponent $\min\{\theta + 1, 3\}$.
4. No factorization of A.
5. Minimal storage space needed (CG process).

Outline

- Trust-region (TR) with truncated CG (tCG) in $\mathbb{R}^n$.
- TR-tCG on the sphere.
- Extensions, comparisons, numerical experiments.

Extensions of the algorithm

- $A = A^T \not\succ 0$. Then the algorithm computes the "leftmost eigenvector".
- Therefore, applied to $-A$, the algorithm computes the rightmost eigenvector.
- Algorithm for the symmetric/positive-definite generalized eigenvalue problem

  $$Ax = \lambda Bx,$$

  using the Rayleigh quotient $y \mapsto (y^T Ay)/(y^T By)$.
- Block version. The iterates are $n \times p$ matrices Y. The cost function is $Y \mapsto \operatorname{trace}\left(Y^T AY(Y^T BY)^{-1}\right)$ and the relevant domain is the *Grassmann manifold*.

Competitors ?

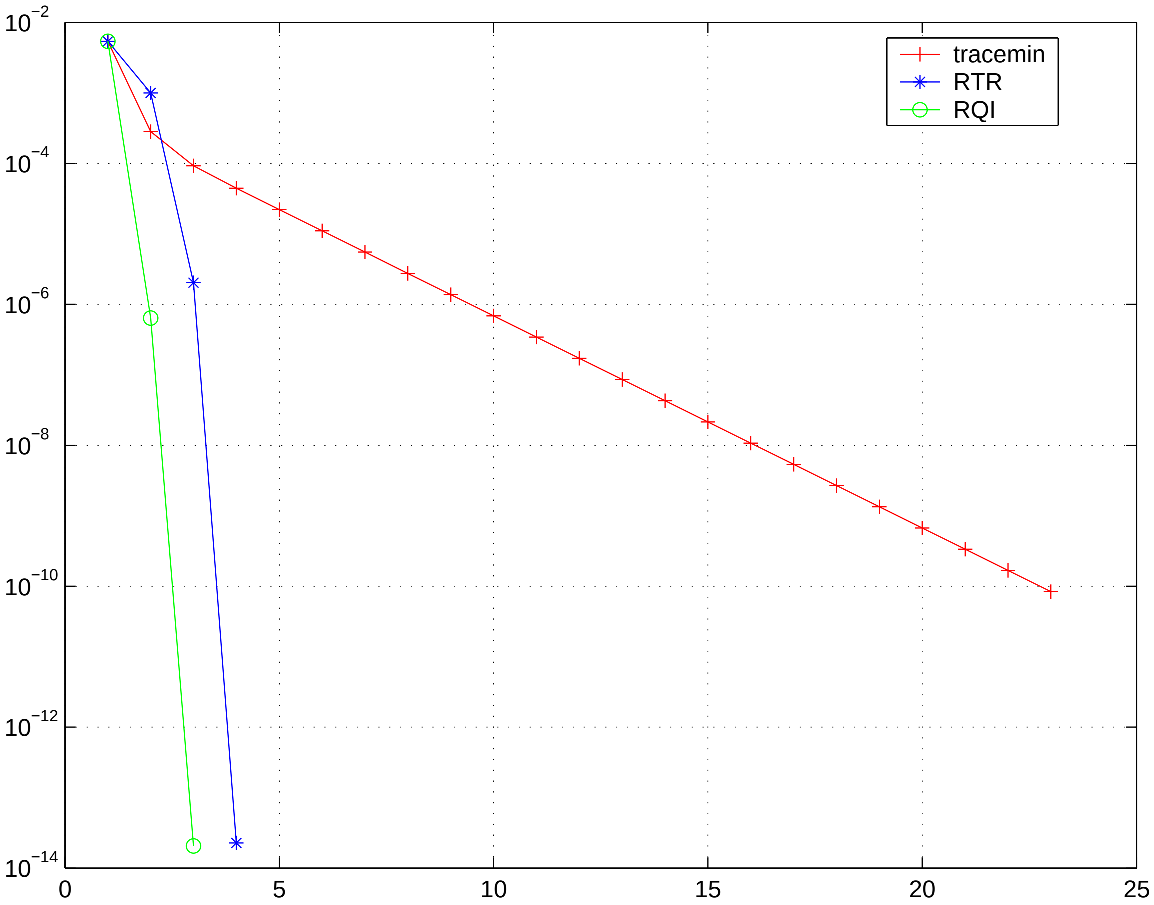
- Scott [Sco81]. Restarted Lanczos method for the generalized eigenproblem, superlinear convergence, without matrix inversion.
  But the storage space becomes very large to ensure superlinear convergence. No proof of convergence.

- Golub and Ye [GY02]. Restarted Lanczos method for the generalized eigenproblem.
  But linear convergence (unless ideal preconditioning).

- SG algorithms of Lippert and Edelman [LE00]. Close precursors.
  But global convergence and generalized eigenproblem not considered.

- Nikpour *et al.* [NMMA04]...
  (See also Mongeau and Torki [MT99].)

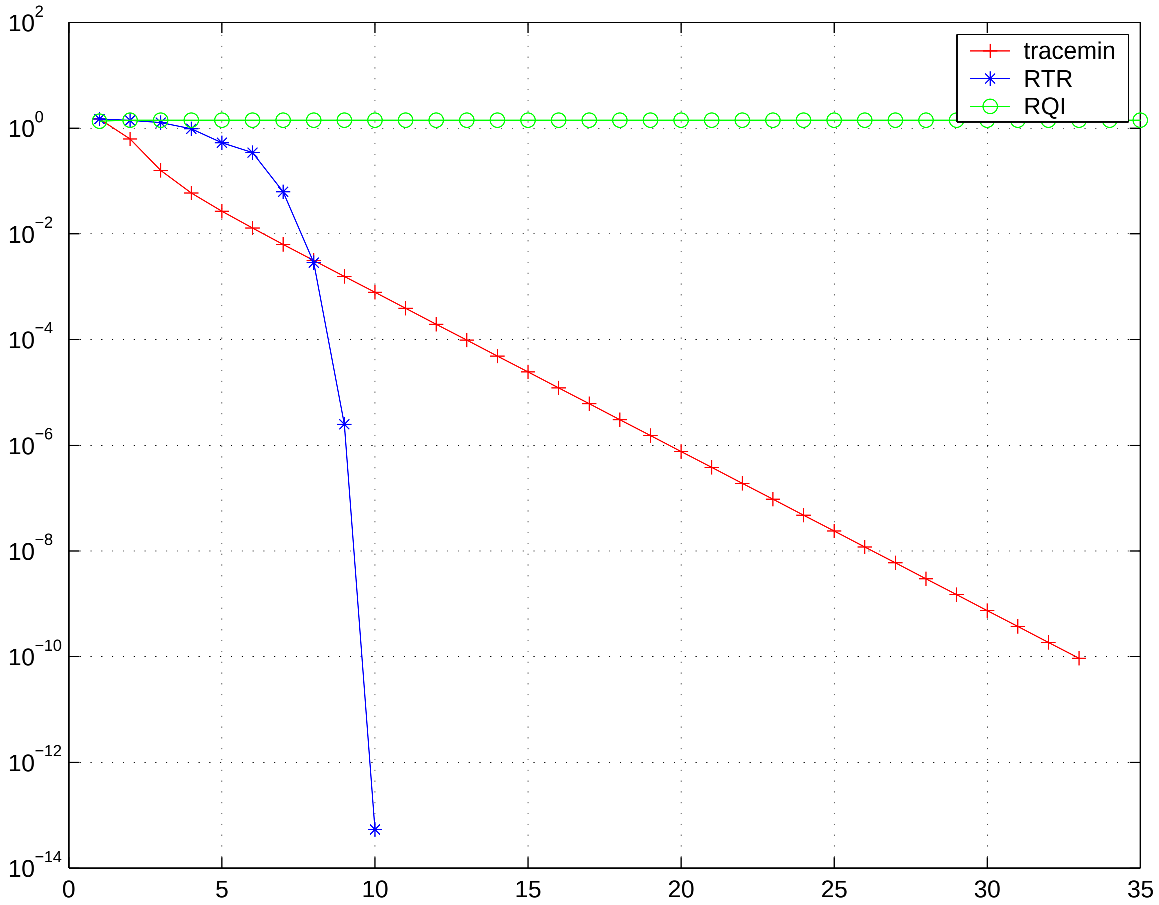
Numerical experiments



Distance to target versus number of outer iterations.

Simple symmetric positive-definite eigenvalue problem.

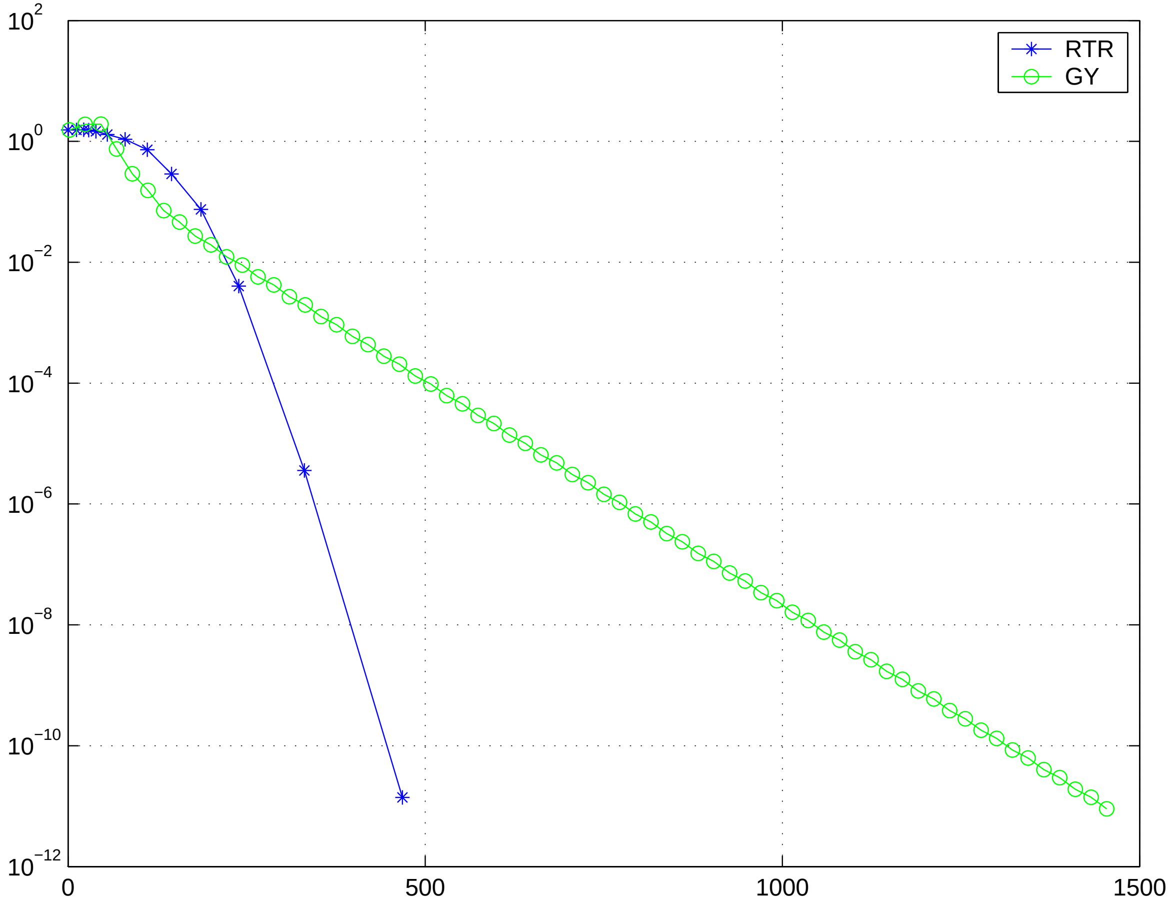
Numerical experiments



Distance to target versus number of outer iterations.

Simple symmetric positive-definite eigenvalue problem.

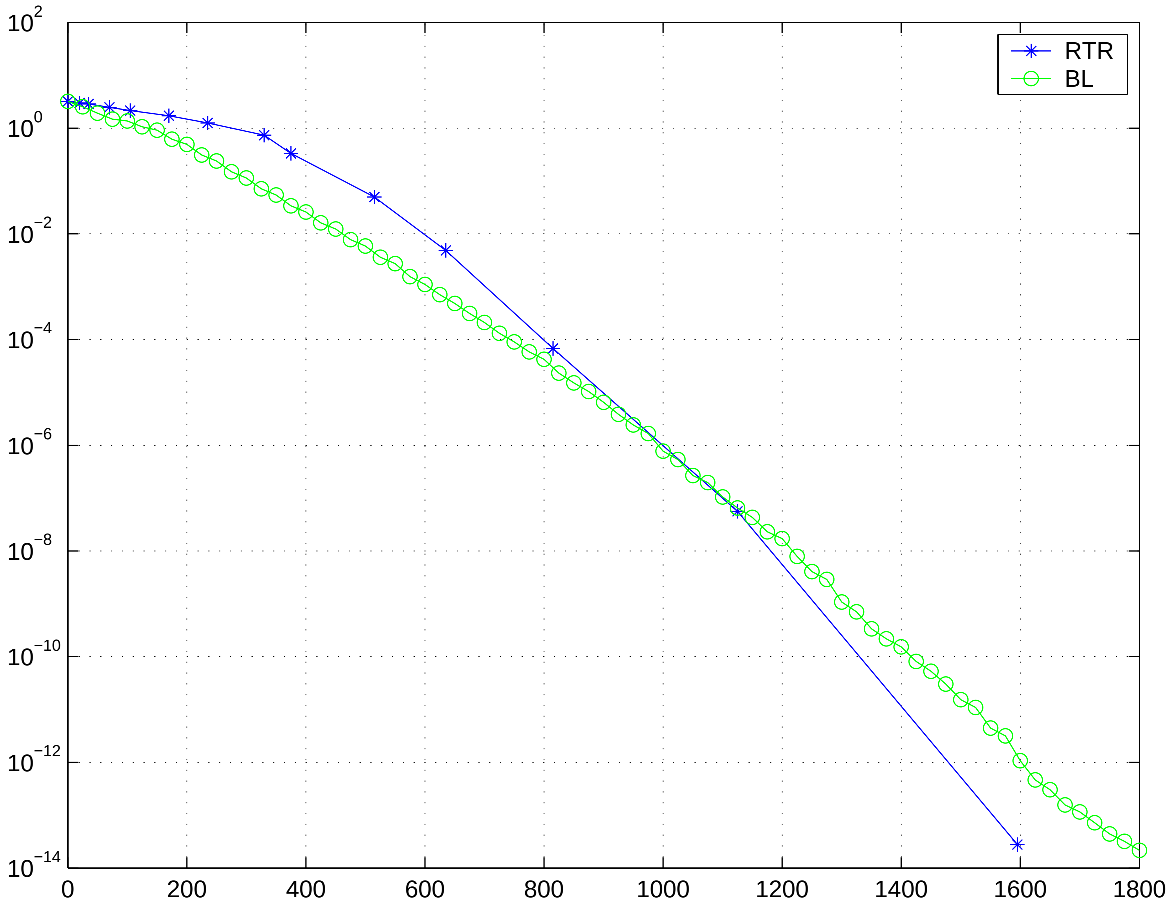
Numerical experiments



Distance to target versus matrix-vector multiplications.

Symmetric/positive-definite generalized eigenvalue problem.

Numerical experiments



Distance to target versus matrix-vector multiplications.

Block version, simple symmetric eigenvalue problem.

Conclusion

The "ideal" minor component algorithm

1. Convergence to some eigenvector for **all** initial conditions.
2. Stable convergence to the leftmost/rightmost eigenvector only.
3. Superlinear local convergence to $\pm v_1$.
4. Matrix A only utilized as operator $x \mapsto Ax$:
   - No exact system solve with matrix A.
   - No factorization of A.
5. Minimal storage space required.

Current work

Hybrid Lanczos-tCG method.

THE END

Numerical linear algebra problems

Several problems in numerical linear algebra can be expressed as finding a minimizer of a well-chosen cost function on a certain manifold. Examples:

- Full EVD.
- Full SVD.
- Computation of left and right, dominant or dominated, p-dimensional invariant subspaces. (Problem on Grassmann)
- Balanced factorization.
- Nonlinear eigenvalue problem.
- Low rank approximation.
- ...

See Helmke and Moore [HM94], Lippert and Edelman [LE00] and references therein.

Manifolds

Roughly speaking, a manifold is a set that looks locally like $\mathbb{R}^n$. Local mappings from the manifold to $\mathbb{R}^n$ are called charts, and the inverse mappings are called parameterizations or systems of coordinates.

The following manifolds are involved in the differential geometric approach to numerical linear algebra problems:

- Orthogonal group.
- Stiefel manifold: $n \times p$ orthonormal matrices.
- Grassmann manifold: p-dimensional subspaces in $\mathbb{R}^n$.
- Oblique manifold: matrices with normalized columns.
- Ellipsoids: $\{Y \in \mathbb{R}^{n\times p} : X^T RX = I\}$.
- Products of these manifolds.

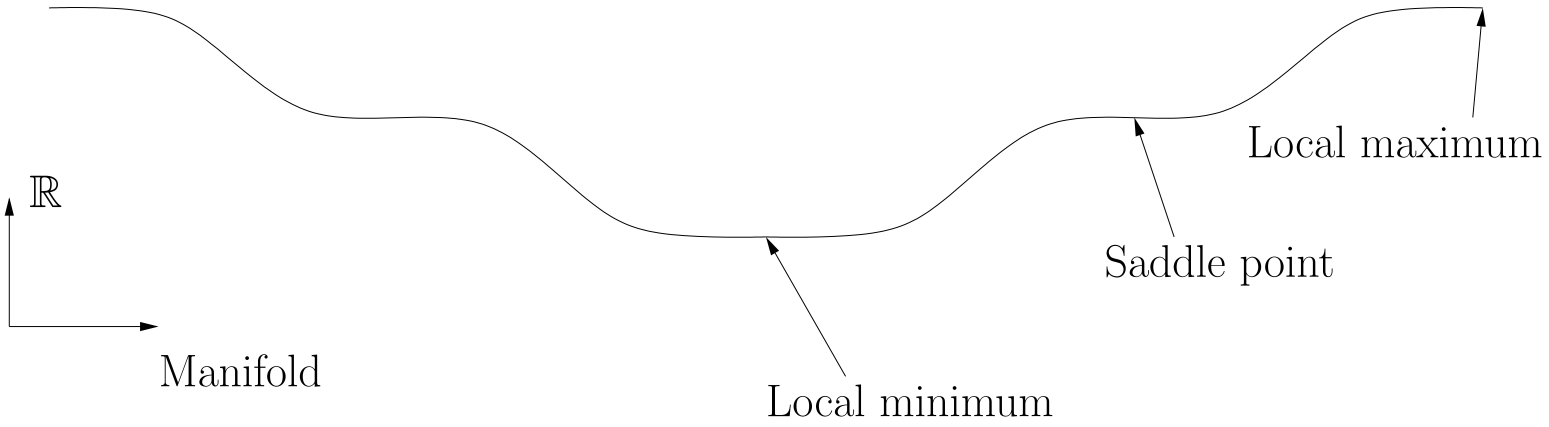
All these manifolds can be turned into Riemannian manifolds by smoothly defining an inner product on the tangent spaces.

Structure of the cost functions

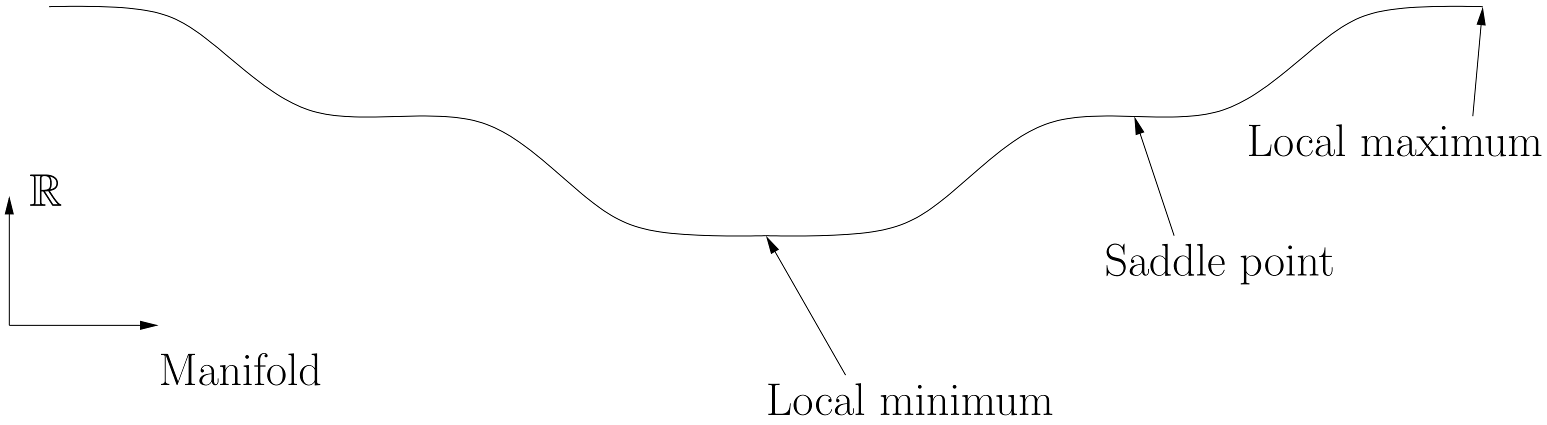
It turns out that most of the cost functions related to linear algebra problems have:

- one or a few local minima which are also global minima,
- several other stationary points (i.e., critical points) that are either saddle points or local maxima.

We assume that only local minima are sought, although the other stationary points are sometimes interesting, too.


Local maximum
ℝ
Saddle point
Manifold
Local minimum

Objectives of iteratives methods on the manifold



- Global convergence: converge to a local minimum from (almost all) initial point.
- Local convergence: superlinear rate of convergence.

Available optimization schemes on manifolds

A few references: Gabay [Gab82], Udrişte [Udr94], Smith [Smi94], Edelman *et al.* [EAS98], Manton [Man02].

It seems that all currently available methods on manifolds are either

- globally convergent but slow (linear), for example gradient descent methods; or
- fast but not (provably) globally convergent, for example the Newton method.

Moreover, in the assessment of speed, one has to take into account the cost of computing iterates. For example, Newton (see e.g. [Smi94, EAS98]) has quadratic convergence but the iterate update is expressed as the solution of a linear system. Approximate solvers can be used (inner iteration) but how they affect the properties of the outer iteration is not well known.

A remedy: trust-region methods

Trust-region is a well-known strategy for optimization in $\mathbb{R}^n$ (see e.g. [CGT00]).

Under mild assumptions, trust-region algorithms converge to a set of stationary points. Cycling behaviour is ruled out.

Moreover, since they are descent methods, the saddle points and local maxima are unstable. We do not expect to observe convergence to them. (This argument has to be improved...)

Schemes to solve the trust-region subproblems are available that yield superlinear rate of convergence. Example: truncated CG (also called Steihaug-Toint). This scheme can be viewed as an approximate Newton method, or as an improved gradient descent method.

Principle of trust-region methods in $\mathbb{R}^n$

1. Consider a cost function f in $\mathbb{R}^n$. Let x_k be the current iterate.
2. Build a model $m_k(s)$ of f around x_k. The model should agree to f at x_k to the first order at least, and to the second order if superlinear convergence is sought.
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.

Principle of trust-region methods in $\mathbb{R}^n$ (cont'd)

4. Compute the ratio

$$\rho = \frac{f(x_k) - f(x_k + s_k)}{m_k(0) - m_k(s_k)}$$

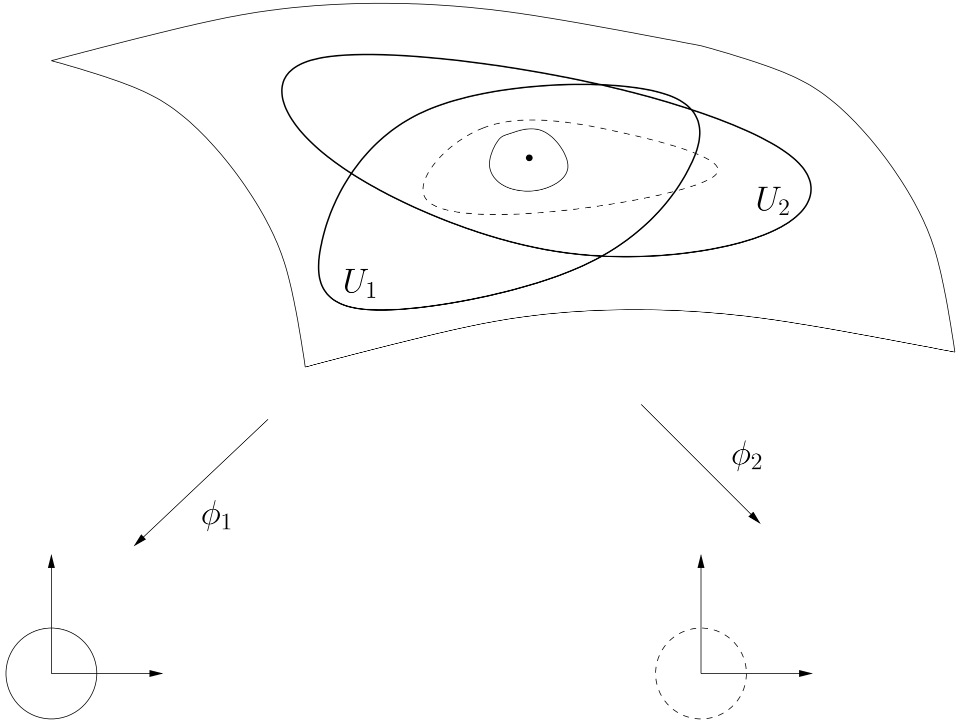
   to compare the actual value of the cost function at the proposed new iterate with the value predicted by the model.

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

For more detail, see e.g. [NW99, CGT00].

Trust-region methods on Riemannian manifolds: difficulties

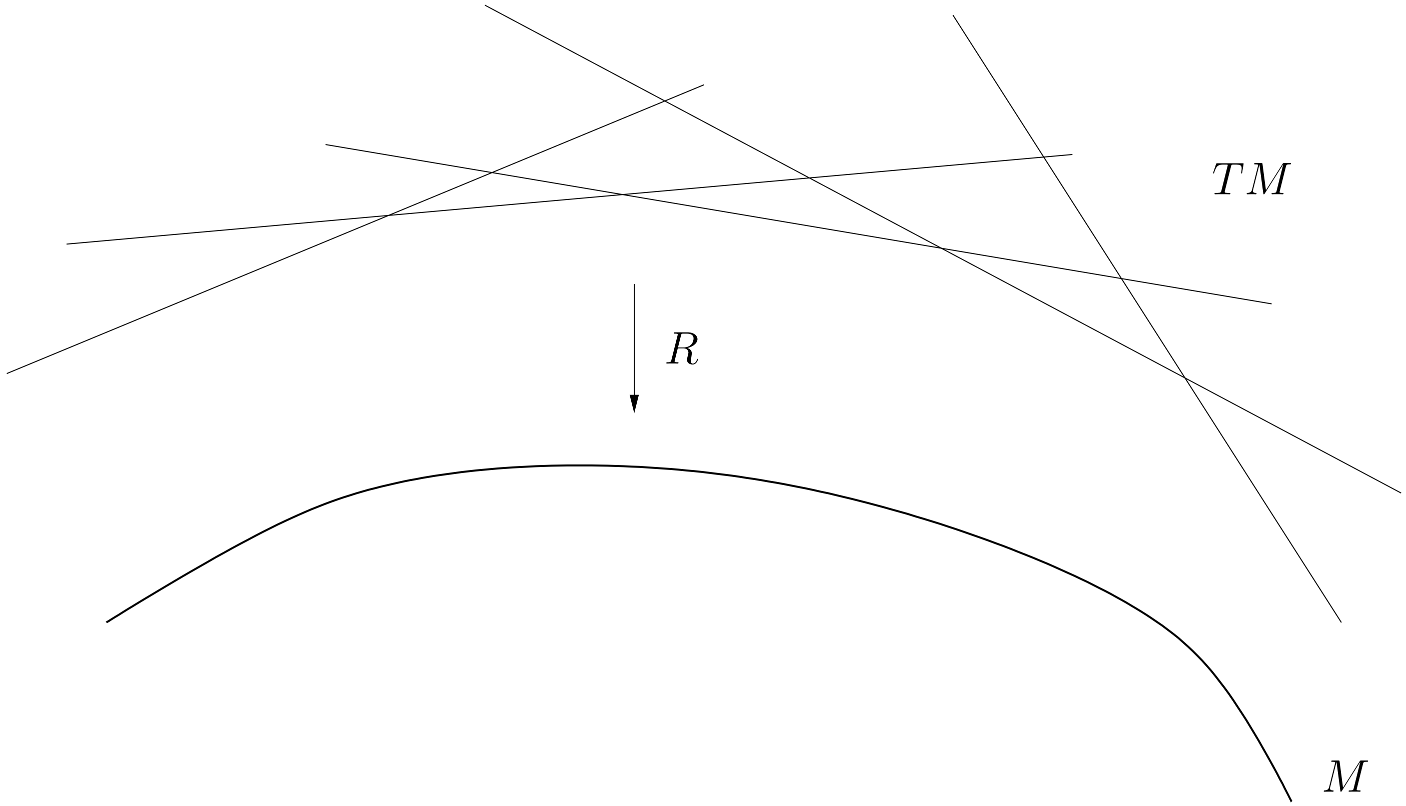
In general, coordinates systems can be scaled without restriction: If ϕ is a chart, then $\alpha\phi$ is still a chart, with $\alpha \in \mathbb{R}$.



To define a notion of trust-region on Riemannian manifolds, one has to use charts with some “rigidity” property.

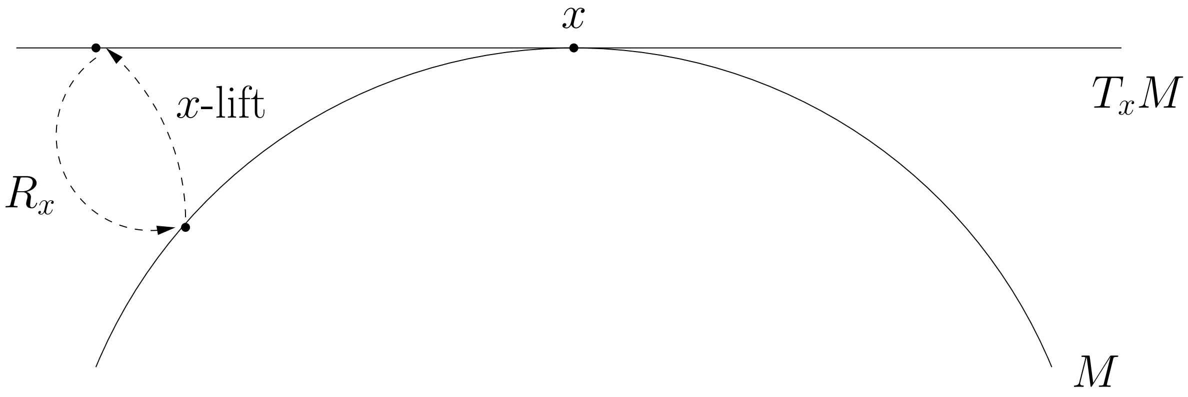
Trust-region methods on Riemannian manifolds: remedies

To assign a "locally rigid" chart to any point on a manifold M, we use the concept of *retraction* introduced (?) in Adler *et al.* [ADM$^+$02].


TM
R
M

Trust-region methods on Riemannian manifolds: remedies (cont'd)

Concept of *retraction*:



1. R_x is defined and one-to-one in a neighbourhood of 0_x in T_xM.
2. $R_x(0_x) = x$.
3. $\mathrm{D}R_x(0_x) = \mathrm{id}_{T_xM}$, the identity mapping on T_xM, with the canonical identification $T_{0_x}T_xM \simeq T_xM$.

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 $\hat{f}_x$ around x_k.

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

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

(note the presence of R_{x_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.
6. Accept or reject the proposed new iterate $R_{x_k} s_k$ according to the value of ρ.
7. Increment k and go to step 2.

Solving the TR subproblem: truncated CG

- Start from the point $s^0 = 0$.
- Compute the first search direction $\delta^0 = -\mathrm{grad}\, f(x_k)$.
- Minimize the model $m_k(s)$ along δ_0 within the trust region. This yields s^1. If the boundary is reached, then stop.
- Compute the conjugate-gradient direction δ^1.
- Minimize the model along $s^1 + \alpha\delta^2$. If the boundary if reached, then stop.
- ... Repeat the procedure until some stopping criterion is satisfied, and return $s_k := s^j$.

Stopping criteria are based on the norm of the residual $\nabla m_k(s^j)$ and on the number of inner iterations (should not grow above than the dimension of the manifold).

Convergence results

Work in progress...

Under mild conditions, one still proves convergence to a set of stationary points.

Convergence can be proved to be locally superlinear if the stopping criterion is adequately chosen and some regularity assumptions are satisfied.

NLA application: full EVD

Manifold: $M = O_n = \{Q \in \mathbb{R}^{n\times n} : Q^T Q = I_n\}$.

Metric: $g_Q(Q\Omega_1, Q\Omega_2) = \mathrm{trace}(\Omega_1^T \Omega_2)$.

Geodesic retraction: $R_Q Q\Omega = \mathrm{Exp}_Q Q\Omega = Q\exp(\Omega)$.

Cheap approximation: $R_Q Q\Omega = \mathrm{qf}(Q(I + \Omega + \frac{1}{2}\Omega^2))$.

Cost function: $f(Q) = \mathrm{trace}(Q^T A Q N)$.

Easy to obtain a formula for the gradient and Hessian:

$$\mathrm{grad}\,\hat{f}(0) = Q[Q^T A Q, N]$$

$$\mathrm{Hess}\hat{f}(0)[Q\Omega] = \frac{1}{2}Q[[Q^T A Q, \Omega], N] + \frac{1}{2}Q[[N, \Omega], Q^T A Q]$$

Then just plug the expressions into the RTR scheme.

NLA application: full SVD

$$M = O_n \times O_p = \{(U,V) : U \in O_n, V \in O_p\}$$

$$g_{(U,V)}((U\Omega_{U1}, V\Omega_{V1}), (U\Omega_{U2}, V\Omega_{V2})) = \mathrm{trace}(\Omega_{U1}^T\Omega_{U2}+\Omega_{V1}^T\Omega_{V2}).$$

$$f(U,V) = \mathrm{trace}(U^T AVN)$$

...

NLA application: dominant eigenvector computation

Reference: [ABG04].

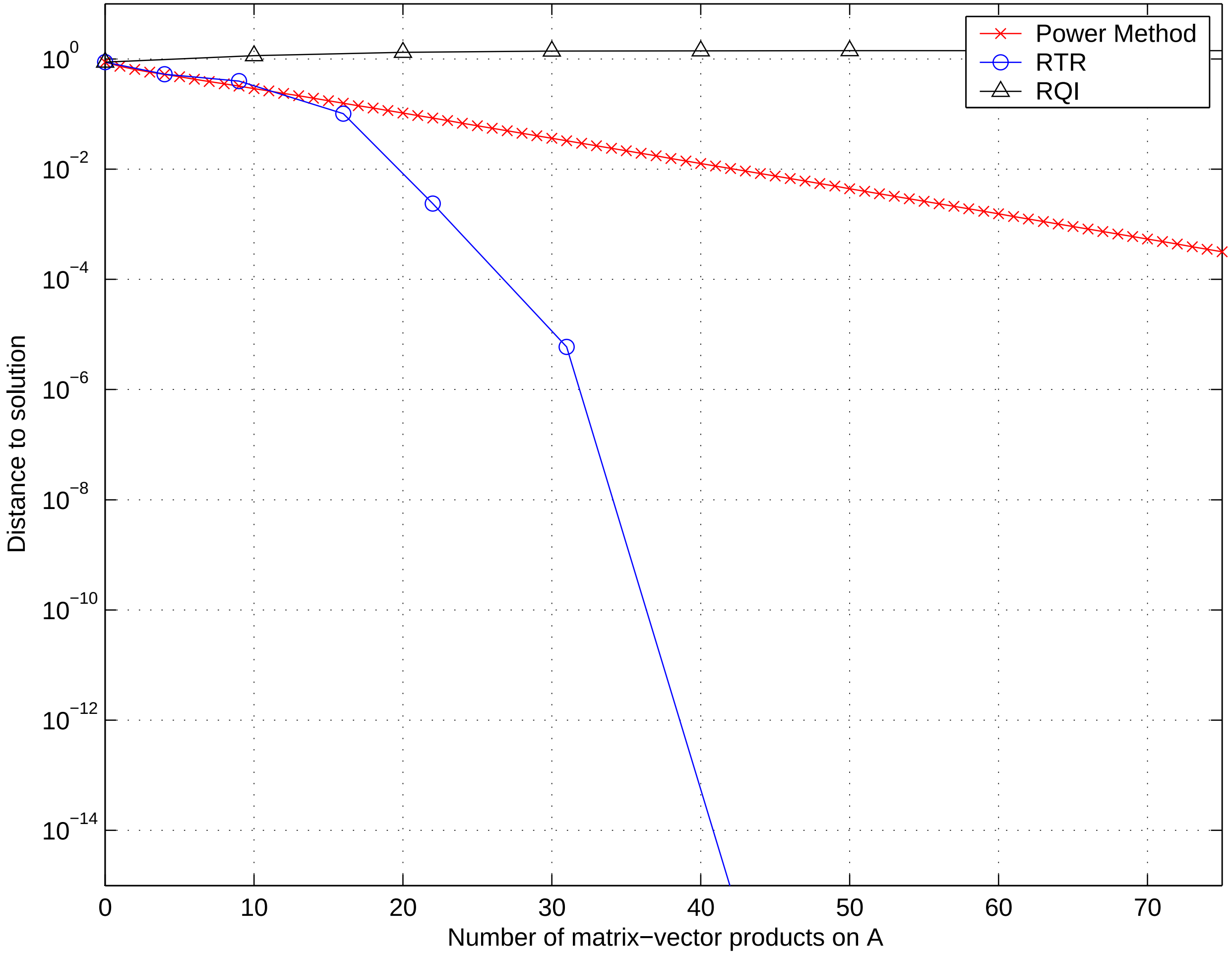
$$M = S^{n-1} = \{x \in \mathbb{R}^n : x^T x = 1\}$$

$$f : \mathbb{R}^n_* \to \mathbb{R} : y \mapsto f_A(y) = \frac{y^T A y}{y^T y} \tag{1}$$

Convergence theory:

- For **all** initial condition, it converges to a set of eigenvectors with same eigenvalue.
- The non-minor eigenvectors are **“unstable”**.
- Convergence is **superlinear** with rate $\min\{3, \theta\}$, θ prescribed.

NLA application: dominant eigenvector computation



NLA application: dominant eigenspace computation

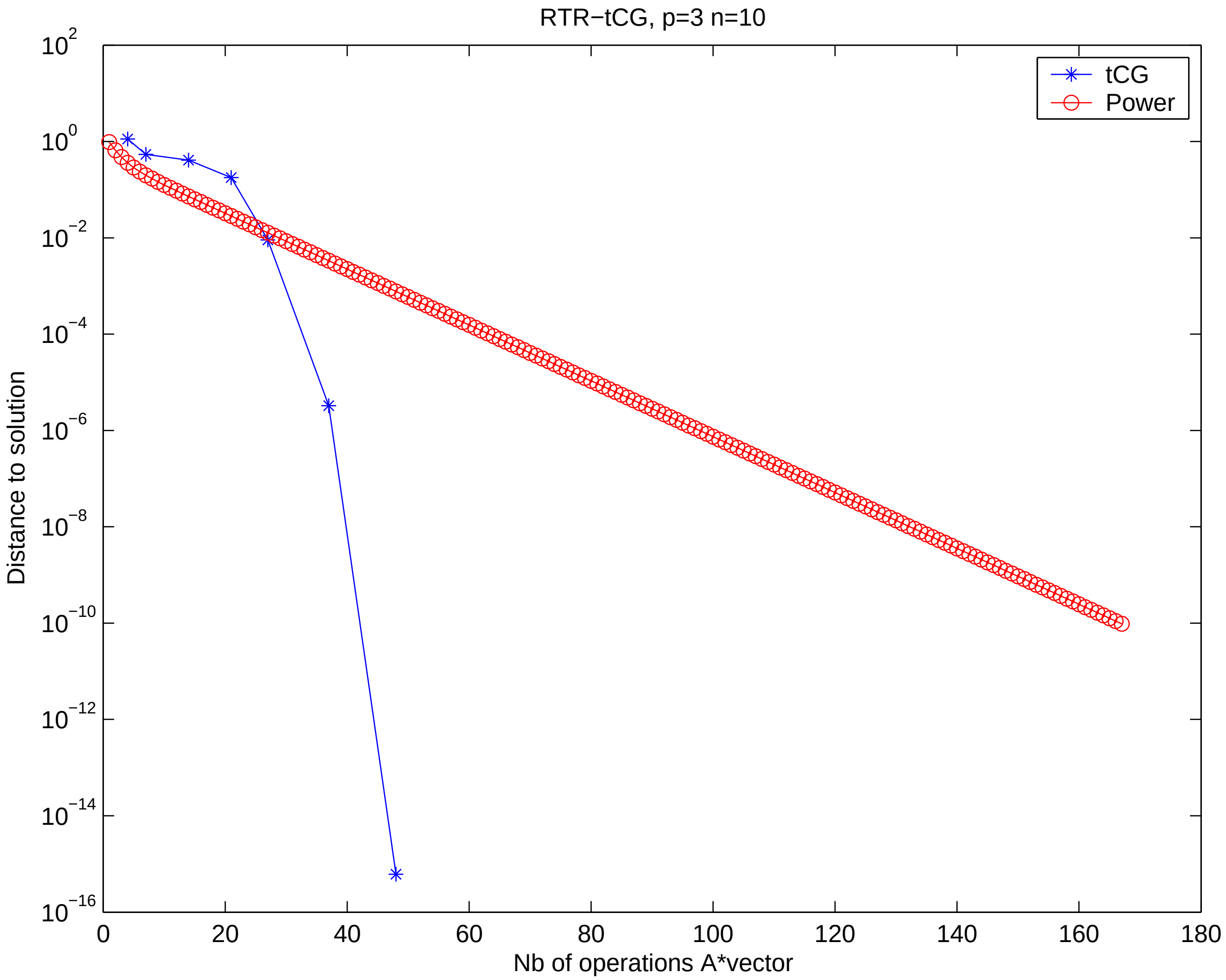
$M = \mathrm{Grass}(p, n)$, the set of p-dimensional subspaces of $\mathbb{R}^n$.

$$f(\mathrm{span}(Y)) = \mathrm{trace}(Y^T AY(Y^T BY)^{-1})$$

where Y is full-rank $n \times p$.

Convergence theory: same...

NLA application: dominant eigenspace computation



References

[ABG04] P.-A. Absil, C. G. Baker, and K. A. Gallivan, *A superlinear method with strong global convergence properties for computing the extreme eigenvectors of a large symmetric matrix*, submitted to the 43rd IEEE Conference on Decision and Control, March 2004.

[ADM$^+$02] R. L. Adler, J.-P. Dedieu, J. Y. Margulies, M. Martens, and M. Shub, *Newton's method on Riemannian manifolds and a geometric model for the human spine*, IMA J. Numer. Anal. **22** (2002), no. 3, 359–390.

[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.

[EAS98] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl. **20** (1998), no. 2, 303–353.

[Gab82] D. Gabay, *Minimizing a differentiable function over a differential manifold*, Journal of Optimization Theory and Applications **37** (1982), no. 2, 177–219.

[GY02] G. H. Golub and Q. Ye, *An inverse free preconditioned Krylov subspace method for symmetric generalized eigenvalue problems*, SIAM J. Sci. Comput. **24** (2002), no. 1, 312–334 (electronic).

[HM94] U. Helmke and J. B. Moore, *Optimization and dynamical systems*, Springer, 1994.

[LE00] R. Lippert and A. Edelman, *Nonlinear eigenvalue problems with orthogonality constraints (Section 9.4)*, Templates for the Solution of Algebraic Eigenvalue Problems (Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and

Henk van der Vorst, eds.), SIAM, Philadelphia, 2000, pp. 290–314.

[Man02] J. H. Manton, *Optimization algorithms exploiting unitary constraints*, IEEE Trans. Signal Process. **50** (2002), no. 3, 635–650.

[MT99] M. Mongeau and M. Torki, *Computing eigenelements of real symmetric matrices via optimization*, Tech. Report MIP 99-54, Université Paul Sabatier, Toulouse, 1999, to appear in Computational Optimization and Applications.

[NMMA04] Maziar Nikpour, Jonathan H. Manton, Iven M. Y. Mareels, and Vadim Adamyan, *Algorithms for extreme eigenvalue problems*, Proceedings of the

16th International Symposium on Mathematical Theory of Networks and Systems (MTNS2004), Leuven, Belgium, 5-9 July 2004, 2004.

[NW99] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999.

[Sco81] David S. Scott, *Solving sparse symmetric generalized eigenvalue problems without factorization*, SIAM J. Numer. Anal. **18** (1981), no. 1, 102–110. MR 82d:65039

[Smi94] S. T. Smith, *Optimization techniques on Riemannian manifolds*, Hamiltonian and gradient flows, algorithms and control, Fields Inst.

Commun., vol. 3, Amer. Math. Soc., Providence, RI, 1994, pp. 113–136.

[Udr94] C. Udrişte, *Convex functions and optimization methods on Riemannian manifolds*, Kluwer Academic Publishers, 1994.