A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations

Bart Vandereycken

Seminar for Applied Mathematics, ETH Zürich

15th Leslie Fox Prize Meeting in Numerical Analysis
June 27, 2011
The low-rank picture
The low-rank picture

Given a matrix $X$, compute its low-rank approximation.

- Why low rank?
  - Like sparsity, low rank is a popular parsimonious structure.
  - Unlike sparsity, it can have global support.

- Why not SVD?
  - Assume that $X$ is available.
  - Does not work if (some of) the entries of $X$ are unavailable.

- Typical application: matrix equations and matrix completion.
Define the set

\[ S_+(k, n) = \{ X : X \in \mathbb{R}^{n \times n}, X = X^T, X \succeq 0, \text{rank}(X) = k \} \].

Our low-rank solver in [V./Vandewalle '10] is based on

\[
\min_X f(X) \quad \text{subject to } X \in S_+(k, n). \tag{1}
\]

Key points:

- Exploit smoothness of \( S_+(k, n) \) as a Riemannian manifold.
- Solve (1) using Riemannian optimization.
- Make the algorithm efficient: precondition the Hessian.
The Lyapunov equation
Matrix equations

- **Matrix equations**: Lyapunov, Sylvester, Riccati

  Abound in $\mathcal{H}_2$-control, model reduction by balanced truncation, stability analysis [Moore '81], [Antoulas '05], [Benner/Mehrmann/Sorensen '05], [Meerbergen/Spence '10], . . .

- The generalized symmetric Lyapunov equation:

  $$AXM^T + MXA^T = C$$

  Given $A, M, C \in \mathbb{R}^{n \times n}$, solve for unknown matrix $X \in \mathbb{R}^{n \times n}$

  - Assume symmetry: $A = A^T, M = M^T, C = C^T \Rightarrow X = X^T$
  - Assume coercivity: $A, M \succ 0, C \succeq 0 \Rightarrow X \succeq 0$

- Matrix $X$ is not directly available without first solving the Lyapunov equation.
Matrix equations applied to large-scale problems, e.g. PDEs

- FEM discretized system $A$ and mass $M$ matrix
- rhs (load) matrix $C = bb^T$, $b \in \mathbb{R}^{n \times k_c}$.

$$A X M^T + M X A^T = C$$

**Main problem**

sparse $A, M$ is $O(n)$ iff solving dense $X$ is $O(n^2)$ memory and $O(n^3)$ flops [Bartels/Stewart ’72]

$\Rightarrow$ so $n \gg 1000$ is problematic
Under reasonable conditions, we have the **low-rank phenomenon**: the singular values of $X$ decay exponentially fast.

$\Rightarrow$ Decay depends on rank rhs $C$, spectrum of $A - \lambda M$, see [Penzl '00], [Antoulas/Sorenson/Zhou '02], [Grasedyck '04], . . .

This means that $X$ has low numerical **rank $k$** for a precision $\epsilon$.

$\Rightarrow A$ has conditioning $\kappa(A)$: $k = O(\log(1/\epsilon) \log(\kappa(A)))$

\[
X = \begin{pmatrix}
V & D & V^T
\end{pmatrix}
\text{trunc} \rightarrow \tilde{X} =
\]

**Main problem**

Compute the “best” rank-$k$ approximation $\tilde{X}$ efficiently in $O(nk^c)$. 

<table>
<thead>
<tr>
<th>$n &lt; O(10^4)$</th>
<th><strong>Schur form</strong> [Bartels/Stewart ‘72]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hammarling, Jonsson, Kågström, Sorensen, Zhou, Quintana-Ortí, van de Geijn, Granat, Kressner, ...</td>
</tr>
<tr>
<td>store $A^{-1}$</td>
<td><strong>Sign function iteration</strong> [Roberts ‘71]</td>
</tr>
<tr>
<td></td>
<td>Beavers, Denman, Byers, Benner, Quintana-Ortí, Grasedyck, Bauer, ...</td>
</tr>
<tr>
<td>apply $(A - \sigma I)^{-1}$</td>
<td><strong>ADI</strong> [Wachspress ‘88]</td>
</tr>
<tr>
<td></td>
<td>Penzl, Li, White, Gugercin, Simoncini, Hodel, Saak, ...</td>
</tr>
<tr>
<td>apply $A$</td>
<td><strong>Krylov subspace</strong> [Saad ‘90]</td>
</tr>
<tr>
<td></td>
<td>Hu, Reichel, Jaimoukha, Kasenally, Hochbruck, Starke, Simoncini, Kressner, ...</td>
</tr>
<tr>
<td>levels $A_i$</td>
<td><strong>Multilevel methods</strong> [Rosen-Wang ‘95]</td>
</tr>
<tr>
<td></td>
<td>Penzl, Grasedyck, Hackbusch, V., Vandewalle, ...</td>
</tr>
</tbody>
</table>

... many other solvers and hybrid combinations
Krylov methods

Krylov subspace methods for $AX +XA^T = bb^T$

- construct a Krylov basis $V_k$

\[ V_k = \text{span}\{A^i b\} \quad \text{with} \quad i = 0 \ldots k \text{ or } i = -k \ldots k \]

- Galerkin condition: solve small Lyapunov equation for $x_k$

\[ (V_k^T A V_k)x_k + x_k (V_k^T A V_k)^T = E_k \]

- Approximation is $X_k = V_k x_k V_k^T$ with $x_k$ such that the energy norm is minimized for the basis $V_k \otimes V_k$

Drawbacks (and similar for most other methods):

- Compute low-rank solutions as a (deliberate) side-effect
- Factors $V_k$ are not very good: only $x_k$ optimized
- Slow convergence $\leadsto$ high-rank factors $\leadsto$ needs truncation

Proposed solution: improve factors by optimizing $V_k$ directly.
Optimization on the manifold of low-rank matrices
The method we proposed in [V./Vandewalle ’10] will

- minimize the energy norm,
- over the manifold of positive semidefinite (PSD) matrices of fixed rank $k$.

\[
\min_{f : S_+(k, n) \rightarrow \mathbb{R}, \ X \mapsto \text{tr}(XAXM) - \text{tr}(XC)},
\\text{s.t.} \ S_+(k, n) = \{X : X \in \mathbb{R}^{n \times n}, \ X \succ 0, \ \text{rank}(X) = k\}.
\]

**Scalability constraint** for each step

- all operations,
- all data structures

must be $O(nk^c)$, $c$ small.
The objective function

\[ f : S_+(k, n) \to \mathbb{R}, \quad X \mapsto \text{tr}(XAXM) - \text{tr}(XC), \]

reflects a weighted norm of the error.

Proof:

- The vec(·) operator gives the isomorphism \( \mathbb{R}^{n^2} \cong \mathbb{R}^{n \times n} \) as

\[ \text{tr}(X^T Y) = \text{vec}(X)^T \text{vec}(Y). \]

- \( AXM + MXA = C \) is a linear system of size \( n^2 \):

\[ \mathcal{L} \text{vec}(X) = \text{vec}(C) \quad \text{with} \quad \mathcal{L} = A \otimes M + M \otimes A. \]
The objective function

- Take $\mathcal{L}$-norm of the error $E = X - X_*$:

  \[ \| \text{vec}(E) \|_{\mathcal{L}}^2 = \text{vec}(E)^T \mathcal{L} \text{vec}(E) \]
  \[ = \text{vec}(E)^T (A \otimes M + M \otimes A) \text{vec}(E) \]
  \[ = 2 \text{tr}(EMEA). \]

- Work out the error $E$:

  \[ \| \text{vec}(E) \|_{\mathcal{L}}^2 = 2 \text{tr}[(X - X_*)M(X - X_*)A] \]
  \[ = 2 \text{tr}(XMXA) - 2 \text{tr}(XC) + 2 \text{tr}(X_* MX_* A) \]
  \[ = 2f(X) + 2 \text{tr}(X_* MX_* A). \]

Minimizing $f(X) \iff$ minimizing $\| \text{vec}(E(X)) \|_{\mathcal{L}}$

Does $\| \text{vec}(E) \|_{\mathcal{L}}$ make sense? If $A, M \succ 0$, then $\mathcal{L} \succ 0$. 
How do we optimize over

\[ S_+(k, n) = \{ X : X \in \mathbb{R}^{n \times n}, X \succ 0, \text{rank}(X) = k \} \]?

Main obstacle: \( S_+(k, n) \) is not a vector space since

\[ \exists X, Y \in S_+(k, n) \implies X + Y \notin S_+(k, n). \]

In general, rank constraints are very difficult. Existing approaches

- Factoring \( X = YY^T \) (non-local optimizers \( YQ \))
- SDP relaxation (drop rank constraint)

are not suitable.
**Problem:** How to optimize on the *curved* space $S_+(k, n)$?

$S_+(k, n)$ is a $C^\infty$ smooth manifold.

Manifold property is well known in [algebraic geometry] and [Helmke/Moore '94].
Riemannian algorithms

- The idea of exploiting the geometry of manifolds turns up in several areas: geometric integration, Lie group methods, . . .

- **Riemannian optimization**: several “classic” algorithms for unconstrained optimization have been adapted to smooth manifolds
  - Steepest descent, conjugate gradients (CG), Newton
  - See: [Luenberger '72], [Gabay '82], [Shub '86], [Smith '93], [Udrişte '94], [Helmke/Moore '94], [Mahony '94], [Owren/Welfert '96], [Edelman/Arias/Smith '98], . . . , [Adler *et al.* '02], [Absil/Mahony/Sepulchre '08], . . .

- Relies on a few basic principles from differential geometry.

- We need new derivations for the geometry of $S_+(k, n)$. 
**Riemannian optimization?**

**Classic unconstrained optimization:** find \( \min f \) on \( \mathbb{R}^n \)

At the current iterate \( x \)

1. Determine a **step** \( p \)
   e.g. steepest descent, conjugate gradient, newton direction

2. Compute a **better point** \( x_+ = x + p \)
   robust with line-search or trust region

3. Loop: \( x \leftarrow x_+ \)
What are the steps $p$?

- Steepest descent: $\text{grad } f(x)$
- Newton direction: second-order model with $\text{Hess } f(x)$

How to get $x_+ = x + p$?

- Every iterate $x, y \in S_+(k, n)$ but $x + y \not\in S_+(k, n)$
**Property:** $S_+(k, n)$ is locally Euclidean = *tangent space*

Properties of the tangent space $T_xS_+(k, n)$:

- Contains tangent vectors
  \[ \dot{\gamma}(0) = \xi \] with a curve $\gamma(t)$ on $S_+(k, n)$

- Linear space: $\xi + \eta \in T_xS_+(k, n)$ for all $\xi, \eta \in T_xS_+(k, n)$

- We can go back to the manifold by retracting, e.g., projecting
  retraction $R_x$ is a smooth map $T_xS_+(k, n) \to S_+(k, n)$
Optimization on manifolds

Result: *Standard unconstrained* optimization

At the current iterate $x$ in the tangent space $T_x S_+ (k, n)$:

1. Determine a step $\xi \in T_x S_+ (k, n)$
   steps are based on the Riemannian gradient and Hessian
2. Compute a better point $x_+ = R_x (\xi)$
3. Loop: $x \leftarrow x_+$

Optimize $\hat{f}_x$, the pullback of $f$ through $T_x S_+ (k, n)$:

$$\hat{f}_x : T_x S_+ (k, n) \rightarrow \mathbb{R}, \quad \xi \mapsto f \circ R_x.$$
The embedded geometry of $S_+(k, n)$
$S_+(k, n)$ as an embedded submanifold

Elements of $S_+(k, n)$: $X = V D V^T$ as EVD,

$$X = \begin{bmatrix} V & V_\perp \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V^T \\ V_\perp^T \end{bmatrix}, \quad D \in \mathbb{R}^{k \times k \text{ diagonal}}.$$

Tangent space at $X \in S_+(k, n)$,

$$T_xS_+(k, n) = \begin{bmatrix} V & V_\perp \end{bmatrix} \begin{bmatrix} S & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} V^T \\ V_\perp^T \end{bmatrix}, \quad S = S^T \in \mathbb{R}^{k \times k}, \quad C \in \mathbb{R}^{n-k \times k}$$

$$= V SV^T + \underbrace{(V_\perp C)}_{Z_U \in \mathbb{R}^{n \times k}} V^T + V \underbrace{(V_\perp C)}_{Z_V \in \mathbb{R}^{n \times k}}^T.$$

The Euclidean metric $\langle \cdot, \cdot \rangle$ restricted to $T_xS_+(k, n)$

$$\langle \xi, \eta \rangle_x := \text{tr}(\xi^T \eta).$$
Retraction: orthogonal projection onto a non-convex set

\[ R_x : T_x S_+ (k, n) \to S_+ (k, n) \]

\[ \xi \mapsto P_{S_+ (k, n)} (X + \xi) = \arg \min \{ \| X + \xi - Z \|_F : Z \in S_+ (k, n) \}. \]

In [V./Vandewalle ’10] we showed:
- Locally, well-defined and \( C^\infty \).
- Since \( \text{rank}(X + \xi) = 2k \), can be computed in \( O(nk^2) \).
- Second-order approximation of the geodesic with expansion

\[ R_X (\xi) = X + \xi + P^p_X (\xi) X^\dagger P^p_X + O(\| \xi \| ^3) \]

Useful for deriving the Riemannian Hessian . . .
The Newton step $\xi$ is the minimizer of the second-order model

$$m_k(\xi) = f(X_k) + \langle \text{grad} f(X_k), \xi \rangle_{X_k} + \frac{1}{2} \langle \text{Hess} f(X_k)[\xi], \xi \rangle_{X_k}$$

- The Euclidean metric $\langle \cdot, \cdot \rangle$ on each $T_X S_+(k, n)$:

$$\langle \xi, \eta \rangle_X := \text{tr}(\xi^T \eta).$$

- The Riemannian gradient of $f$ is the vector $\text{grad} f$ such that

$$\frac{\text{grad} f(X)}{\parallel \text{grad} f(X) \parallel} = \arg \max_{\xi \in T_X S_+(k, n), \parallel \xi \parallel=1} Df(X)[\xi].$$

- Gradient is the direction of steepest ascent w.r.t. $\langle \cdot, \cdot \rangle_X$. 

Riemannian gradient
The Newton step $\xi$ is the minimizer of the second-order model

$$m_k(\xi) = f(X_k) + \langle \text{grad} f(X_k), \xi \rangle_{X_k} + \frac{1}{2} \langle \text{Hess} f(X_k)[\xi], \xi \rangle_{X_k}$$

- The Riemannian Hessian of $f$ is the unique linear and symmetric mapping $\text{Hess} f$

  $$\text{Hess} f : T_XS_+(k, n) \to T_XS_+(k, n),$$

  such that

  $$\langle \text{Hess} f(X)[\xi], \xi \rangle_X = \frac{d^2}{dt^2} \bigg|_{t=0} f(R_X(t \xi)).$$

- Valid because $R_X$ is a second-order appr. to geodesic!
Second-order model

Applied to $f(X) = \text{tr}(XAXM) - \text{tr}(XC)$, we obtained analytical expressions for

- the **gradient** of $f(X)$

$$
\text{grad} f(X) = P_T(R), \quad R := AXM + MXA - C,
$$

with $P_T(Z) := P_V Z P_V + P_V^\perp Z P_V + P_V Z P_V^\perp$ the orthogonal projection onto $T_X S_+(k, n)$.

- the **Hessian** as matrix vector product

$$
\text{Hess} f(X)[\xi] = P_T(A\xi M + M\xi A) + P_T^p(RP_T^p(\xi)X^\dagger + X^\dagger P_T^p(\xi)R)
$$

with $P_T^p(Z) := P_V^\perp Z P_V + P_V Z P_V^\perp$

**implies** second-order model can be evaluated in $O(nk^2)$ flops.
Trust-Region Newton on the manifold
RLyap: final algorithm to solve for low-rank approximation of $X$.

- Choose\(^1\) a rank $k$, minimize

$$\min_{X \in S_+(k,n)} \tr(XA XM) - \tr(XC).$$

by the Riemannian Trust-Region (RTR) method of [Absil/Baker/Gallivan ’07].

- Key step: solve Newton system

$$m_k : T_{X_k} S_+(k, n) \rightarrow \mathbb{R},$$

$$\xi \mapsto f(X_k) + \langle \text{grad } f(X_k), \xi \rangle + \frac{1}{2} \langle \text{Hess } f(X_k)[\xi], \xi \rangle$$

with truncated PCG to obtain step

$$\eta_k = \arg \min m_k(\xi) \quad \text{s.t. } \|\xi\| \leq \Delta_k.$$

\(^1\)perform an outer loop to get the minimum rank for a desired residual
Experimental results for RTR

Superlinear convergence of TR Newton

Test problem: 1D Laplace with $n = 1000$, ranks $k = 5, 10, 15, 20$. 
Experimental results for RTR

Relative error of low-rank approximations for different ranks


$^2$[Penzl ’99],[Li/White ’04] $^2$[Simoncini ’07]
Preconditioning
RTR uses (truncated) CG to solve the Newton system.

- Too many iterations for PDEs.

Example RTR for 2D Laplace, $k = 15$, tol. gradient $= 10^{-10}$:

<table>
<thead>
<tr>
<th>$n$</th>
<th>150²</th>
<th>200²</th>
<th>250²</th>
<th>300²</th>
<th>350²</th>
<th>400²</th>
<th>450²</th>
<th>500²</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{outer}$</td>
<td>46</td>
<td>44</td>
<td>49</td>
<td>44</td>
<td>43</td>
<td>44</td>
<td>56</td>
<td>48</td>
</tr>
<tr>
<td>$\sum n_{inner}$</td>
<td>1913</td>
<td>2173</td>
<td>2984</td>
<td>3158</td>
<td>4076</td>
<td>4185</td>
<td>5375</td>
<td>5622</td>
</tr>
<tr>
<td>max $n_{inner}$</td>
<td>414</td>
<td>529</td>
<td>624</td>
<td>731</td>
<td>757</td>
<td>858</td>
<td>1004</td>
<td>1080</td>
</tr>
</tbody>
</table>

- Can we precondition CG?

The Riemannian Hessian is a modified Euclidean Hessian:

$$
\mathcal{H}_X = P_X \left( A \otimes M + M \otimes A \right) P_X + P_X^P \left( X^\dagger \otimes R + R \otimes X^\dagger \right) P_X^P.
$$

- Neglect curvature $\leadsto$ precondition with $P_X \mathcal{L} P_X$.
- $P_X \mathcal{L} P_X$ is the (first-order) Gauss–Newton model of $f(X)$ on $S_+(k, n)$, cfr. [Adler/Dedieu/Margulies/Martens/Shub '02].
Does it reduce the number of iterations?

- Observed to be mesh-independent.
- Same 2D Laplace example:

<table>
<thead>
<tr>
<th>prec.</th>
<th>( n )</th>
<th>150(^2)</th>
<th>200(^2)</th>
<th>250(^2)</th>
<th>300(^2)</th>
<th>350(^2)</th>
<th>400(^2)</th>
<th>450(^2)</th>
<th>500(^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>( n_{\text{outer}} )</td>
<td>46</td>
<td>44</td>
<td>49</td>
<td>44</td>
<td>43</td>
<td>44</td>
<td>56</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>( \sum n_{\text{inner}} )</td>
<td>1913</td>
<td>2173</td>
<td>2984</td>
<td>3158</td>
<td>4076</td>
<td>4185</td>
<td>5375</td>
<td>5622</td>
</tr>
<tr>
<td></td>
<td>( \max n_{\text{inner}} )</td>
<td>414</td>
<td>529</td>
<td>624</td>
<td>731</td>
<td>757</td>
<td>858</td>
<td>1004</td>
<td>1080</td>
</tr>
<tr>
<td>( P_x \mathcal{L} P_x )</td>
<td>( n_{\text{outer}} )</td>
<td>39</td>
<td>40</td>
<td>42</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>47</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>( \sum n_{\text{inner}} )</td>
<td>83</td>
<td>83</td>
<td>91</td>
<td>94</td>
<td>96</td>
<td>101</td>
<td>88</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>( \max n_{\text{inner}} )</td>
<td>14</td>
<td>13</td>
<td>15</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>12</td>
<td>10</td>
</tr>
</tbody>
</table>

Is it faster?

- Can be solved analytically \([\text{V.} / \text{Vandewalle '10]}\) for \( M = I \).
- Assuming \( (A + \lambda I)^{-1} \) is \( O(n) \), total cost is \( O(nk^2) \) \( \rightsquigarrow \) AMG.
- Gauss–Newton as solver is not efficient (not small residual).
Numerical results for RLyap

Performance of the Riemannian optimization approach is comparable with the state-of-the-art, yet more general.

RLyap compared with CFADI [Penzl ’99],[Li/White ’04] and KPIK [Simoncini ’07] for 2D Laplace, rank one rhs.

<table>
<thead>
<tr>
<th></th>
<th>PCG with AMG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RLyap</td>
</tr>
<tr>
<td>$n = 500^2$</td>
<td>time (s.)</td>
</tr>
<tr>
<td></td>
<td>rank $X$</td>
</tr>
<tr>
<td>$n = 1000^2$</td>
<td>time (s.)</td>
</tr>
<tr>
<td></td>
<td>rank $X$</td>
</tr>
<tr>
<td>$n = 1500^2$</td>
<td>time (s.)</td>
</tr>
<tr>
<td></td>
<td>rank $X$</td>
</tr>
</tbody>
</table>

Tol. on rel. residual $= 10^{-6}$; $(A + \lambda I)^{-1}$ solved by PCG+AMG.
When the r.h.s. $C$ is not of low rank, RLyap can be more efficient.

- Laplace; full matrix $C$, rank $k$ approximated $C_k$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\tau$</th>
<th>solver</th>
<th>RLyap</th>
<th>CF-ADI</th>
<th>RLyap</th>
<th>CF-ADI</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 40000$</td>
<td>$1e-6$</td>
<td>time (s.)</td>
<td>70.3</td>
<td>(38.7)</td>
<td>48.9</td>
<td>111.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rank $X$</td>
<td>23</td>
<td>35</td>
<td>25</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>residual</td>
<td>9.87e−7</td>
<td>2.67e−6</td>
<td>9.30e−7</td>
<td>8.61e−7</td>
</tr>
<tr>
<td>$n = 80000$</td>
<td>$1e-6$</td>
<td>time (s.)</td>
<td>169.7</td>
<td>(103.1)</td>
<td>116.8</td>
<td>(232.1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rank $X$</td>
<td>25</td>
<td>35</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>residual</td>
<td>9.89e−7</td>
<td>2.68e−6</td>
<td>9.81e−7</td>
<td>2.98e−6</td>
</tr>
<tr>
<td>$n = 160000$</td>
<td>$5e-5$</td>
<td>time (s.)</td>
<td>176.8</td>
<td>139.5</td>
<td>104.7</td>
<td>300.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>rank $X$</td>
<td>12</td>
<td>33</td>
<td>12</td>
<td>48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>residual</td>
<td>1.44e−5</td>
<td>3.57e−5</td>
<td>3.35e−5</td>
<td>3.47e−5</td>
</tr>
</tbody>
</table>

- RLyap can use a matrix-free $C$, other methods can not.
Thank you for your attention
Second-order model

Accuracy of models with different Hessians

Figure: Relative error of the linear and quadratic models. The triangles indicate the second and third order convergence of the error.

Optimization problem is non-convex $\rightsquigarrow$ robustify Newton by TR!