Riemannian Multigrid Line Search for Low-Rank Problems

低秩度問題的黎曼多網格線搜索

Marco Sutti

Postdoctoral fellow at NCTS

國家理論科學研究中心 數學組、博士後

NCKU Mathematics Colloquium

May 8, 2025

Overview

Paper: Riemannian Multigrid Line Search for Low-Rank Problems, M. Sutti and B. Vandereycken, SIAM J. Sci. Comput., 43(3), A1803–A1831, 2021.

- New algorithm to solve large-scale optimization problems.
- Minimize an objective function on the Riemannian manifold of fixed-rank matrices using a multigrid idea.
  - Low-rank format for efficient implementation.
  - Multilevel idea of Multigrid Line-Search (MGLS) [Wen/Goldfarb 2009].

This talk:

I. (retraction-based) **Riemannian optimization framework**.

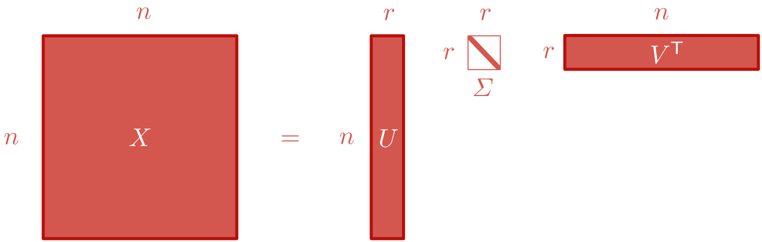
II. Optimization on the **manifold of fixed-rank matrices**.

III. **Multilevel strategy** and insight on **implementation.**

IV. **Numerical experiments.**

Low-rank format and motivation

- Often we need to discretize a problem to represent the continuous solution.
- For high-dimensional problems, a “naive” discretization with n degrees of freedom in each dimension leads to n^d coefficients.
- Since the number of coefficients scales exponentially by d but the accuracy is typically determined by n, this poses a limitation on the size of the problems $\rightsquigarrow$ *Curse of dimensionality*.
- One possible workaround $\rightsquigarrow$ use the singular value decomposition (SVD):

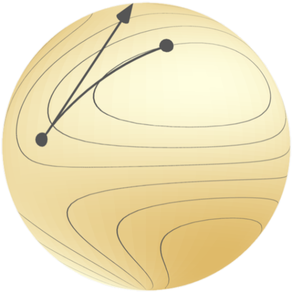


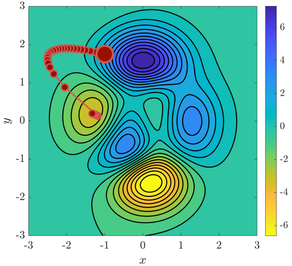
- Only $2nr + r$ coefficients instead of n^2. If $r \ll n$, then big memory savings.
- Perform the calculations directly in the factorized format.

I. Optimization on matrix manifolds

Riemannian optimization/1

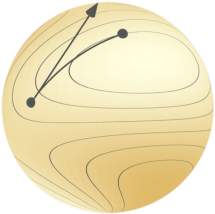
- The Riemannian optimization framework solves constrained optimization problems where the constraints have a geometric nature.
  - Exploit the underlying geometric structure of the problems. The optimization variables are constrained to a smooth manifold.
- Traditional optimization methods rely on the Euclidean vector space structure.
  - E.g., the steepest descent method for minimizing $f : \mathbb{R}^n \to \mathbb{R}$ updates $\mathbf{x}_k$ by moving in the direction $\mathbf{d}_k$ of the anti-gradient of f, by a step size α_k chosen according to a line-search rule.





Manifold optimization: [Edelman et al. 1998, Absil et al. 2008, Boumal 2023], …
The image above has been taken from the Manopt website: https://www.manopt.org/

Riemannian optimization/2



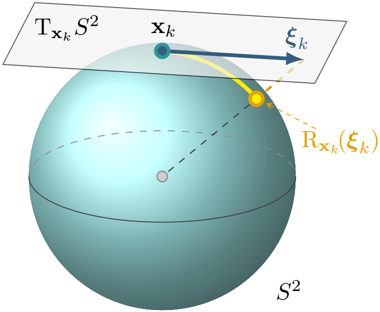
- Formally, we can state the optimization problem as
$$\min_{x \in \mathcal{M}} f(x),$$
where $f : \mathcal{M} \to \mathbb{R}$ is the objective function and $\mathcal{M}$ is some matrix manifold.
- **Matrix manifold**: any manifold that is constructed from $\mathbb{R}^{n \times p}$ by taking either **embedded submanifolds** or **quotient manifolds**.
  - Examples of embedded submanifolds: unit sphere, orthogonal Stiefel manifold, manifold of fixed-rank matrices, …
  - Examples of quotient manifolds: the Grassmann manifold, the flag manifold.
- A manifold $\mathcal{M}$ endowed with a smoothly-varying inner product (called Riemannian metric g) is called Riemannian manifold.

  $\rightsquigarrow$ A couple $(\mathcal{M}, g)$, i.e., a manifold with a Riemannian metric on it.

Manifold optimization: [Edelman et al. 1998, Absil et al. 2008, Boumal 2023], …

Riemannian optimization/3

- A line-search method in the Riemannian framework determines at $\mathbf{x}_k$ on a manifold $\mathcal{M}$ a search direction $\boldsymbol{\xi}_k$ on $\mathrm{T}_{\mathbf{x}_k}\mathcal{M}$.
- $\mathbf{x}_{k+1}$ is then determined by a line search along a curve $\alpha \mapsto \mathrm{R}_{\mathbf{x}_k}(\alpha\boldsymbol{\xi}_k)$ where $\mathrm{R}_{\mathbf{x}_k} : \mathrm{T}_{\mathbf{x}_k}\mathcal{M} \to \mathcal{M}$ is the retraction mapping.
- Repeat for $\mathbf{x}_{k+1}$ in the role of $\mathbf{x}_k$.



- Search directions can be the negative of the Riemannian gradient, leading to the Riemannian gradient descent method (RGD).
  - Other choices of search directions $\rightsquigarrow$ other methods, e.g., **Riemannian trust-region method** or **Riemannian BFGS method**.

Riemannian trust-region method: [Absil/Baker/Gallivan 2007], Riemannian BFGS: [Ring/Wirth 2012]

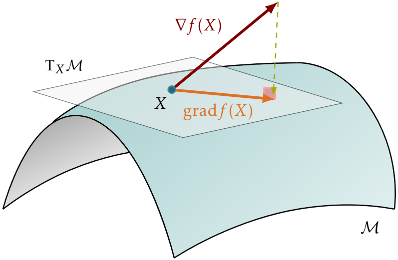
Riemannian gradient

Let $f : \mathcal{M} \to \mathbb{R}$. E.g., the objective function in an optimization problem.

$\rightsquigarrow$ For any embedded submanifold (Prop. 3.6.1 in Absil et al., 2008):

- Riemannian gradient: projection onto $\mathrm{T}_X\mathcal{M}$ of the **Euclidean gradient**

$$\operatorname{grad} f(X) = \mathrm{P}_{\mathrm{T}_X\mathcal{M}}(\nabla f(X)).$$

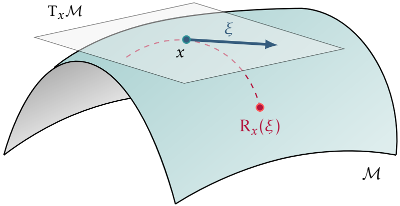


$\rightsquigarrow$ $\nabla f(X)$ is the **Euclidean gradient** of $f(X)$.

Matrix and vector calculus: The Matrix Cookbook, www.matrixcalculus.org, …
Automatic differentiation on low-rank manifolds: [Novikov/Rakhuba/Oseledets 2022]

Retraction mapping

- Move in the direction of ξ while remaining constrained to $\mathcal{M}$.
- Smooth mapping $\mathrm{R}_x : \mathrm{T}_x\mathcal{M} \to \mathcal{M}$ with a local condition that preserves gradients at x.



- The Riemannian exponential mapping is also a retraction, but it is not computationally efficient.
- Retractions: first-order approximation of the Riemannian exponential!

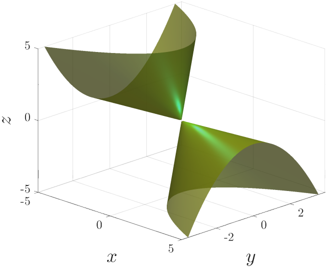
Constructing retractions: [Absil/Malick 2012]

II. Optimization on $\mathcal{M}_k$

Optimization on $\mathcal{M}_k$/I

- Optimization problem on the manifold of fixed-rank matrices

$$\mathcal{M}_k = \{X \in \mathbb{R}^{m\times n} : \operatorname{rank}(X) = k\}.$$



- Using the SVD, one has the equivalent characterization

$$\mathcal{M}_k = \{U\Sigma V^\top : \; U^\top U = I_k, \; V^\top V = I_k, \\ \Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k) \in \mathbb{R}^{k\times k}, \; \sigma_1 \geq \cdots \geq \sigma_k > 0\}.$$

- A tangent vector ξ at $X = U\Sigma V^\top$ is represented as

$$\xi = UMV^\top + U_p V^\top + U V_p^\top,$$

$$M \in \mathbb{R}^{k\times k}, \quad U_p \in \mathbb{R}^{m\times k}, \quad U_p^\top U = 0, \quad V_p \in \mathbb{R}^{n\times k}, \quad V_p^\top V = 0.$$

$\rightsquigarrow$ ξ is a rank-$2k$ bounded matrix. Useful in implementation.

Optimizing on submanifold $\mathcal{M}_k$: [Vandereycken 2013]

Optimization on $\mathcal{M}_k$/II

- The Riemannian metric is

$$g_X(\xi,\eta) = \langle \xi,\eta\rangle = \mathrm{Tr}(\xi^\top \eta), \quad \text{with} \quad X \in \mathcal{M}_k \quad \text{and} \quad \xi,\eta \in \mathrm{T}_X\mathcal{M}_k,$$

  where ξ, η are seen as matrices in the ambient space $\mathbb{R}^{m\times n}$.

  $\rightsquigarrow$ Flop count: $4nk + 2k^2$.

- Orthogonal projection onto the tangent space at X is

$$\mathrm{P}_{\mathrm{T}_X\mathcal{M}_k} \colon \mathbb{R}^{m\times n} \to \mathrm{T}_X\mathcal{M}_k, \qquad Z \mapsto \mathrm{P}_U\, Z\, \mathrm{P}_V + \mathrm{P}_U^\perp\, Z\, \mathrm{P}_V + \mathrm{P}_U\, Z\, \mathrm{P}_V^\perp.$$

  $\rightsquigarrow$ If Z allows for fast matvec product, then $\mathrm{P}_{\mathrm{T}_X\mathcal{M}_k}$ can also be computed efficiently in the above tangent vector format.

- Riemannian gradient: projection onto $\mathrm{T}_X\mathcal{M}_k$ of the Euclidean gradient

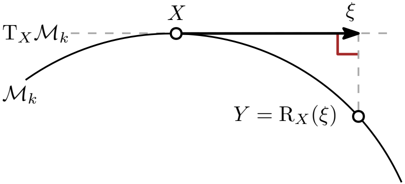
$$\mathrm{grad}\, f(X) = \mathrm{P}_{\mathrm{T}_X\mathcal{M}_k}(\nabla f(X)).$$

Optimization on $\mathcal{M}_k$/III

- Smooth map: Retraction $\mathrm{R}_X : \mathrm{T}_X\mathcal{M}_k \to \mathcal{M}_k$. Typical: truncated SVD.
- Alternative: Orthographic retraction. Given $X = USV^\top$ and $\xi = UMV^\top + U_pV^\top + UV_p^\top$ with $U^\top U_p = 0$ and $V^\top V_p = 0$,

$$\mathrm{R}_X(\xi) = (U(S+M) + U_p)(S+M)^{-1}((S+M)V^\top + V_p^\top).$$

$\rightsquigarrow$ Flop count: $12nk^2 + \mathcal{O}(k^3)$.



- Inverse orthographic retraction of Y at X:

$$\mathrm{R}_X^{-1}(Y) = \mathrm{P}_{\mathrm{T}_X\mathcal{M}_k}(Y - X).$$

Many retractions: [Absil/Malick 2012, Absil/Oseledets 2015]

III. Multigrid and multilevel optimization: from Euclidean to Riemannian

(Linear) Multigrid (in Euclidean space)

- Among the most efficient methods for (discretized) elliptic PDEs.
- Idea: hierarchy of grids $\Omega_{h_{\ell_f}}, \Omega_{h_{\ell_f - 1}}, \ldots, \Omega_{h_{\ell_c}}$.

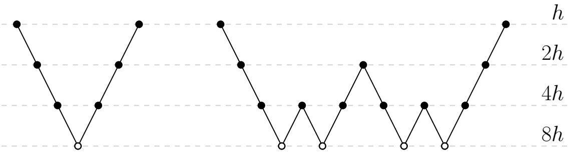


Figure: Illustration of a V- and a W-cycle with cycle index $\gamma = 2$ and four grid levels.

Two basic principles of multigrid:

1. **Smoothing principle**. Many classical iterative methods (e.g., Gauss–Seidel) when applied to discrete elliptic problems show a strong smoothing effect on the error of any approximation.
2. **Coarse-grid correction principle**. A smooth error term can be well represented on a coarse grid.

- Most desirable property of multigrid: mesh-independent convergence.

Nonlinear multigrid in Euclidean space

Full Approximation Scheme (**FAS**) for nonlinear PDE, $A(x) = b$.

- Multigrid idea for solving A on several fine and coarse grids.
- Fine grid $\cdot_h$ smooths the error (with cheap algorithm). Coarse grid $\cdot_H$ computes smooth correction (by recursion). Transfer operators I_h^H and I_H^h between grids (by interpolation).
- Discretized **nonlinear** equation on fine grid:

$$A_h(x_h) = b_h.$$

- **Principle behind FAS**: solve for the error e_H in the the coarse-grid equation as a full approximation $\bar{x}_H + e_H$,

$$A_H(\bar{x}_H + e_H) = \underbrace{r_H + A_H(\bar{x}_H)}_{=: b_H},$$

  with restricted residual $r_H = I_h^H(A_h(x_h) - b_h)$.

FAS: [Hackbusch 1985, Brandt et al. 1985]

Multilevel optimization in Euclidean space/I

FAS can be generalized to multilevel minimization of an objective f.

- Objective functions on fine and coarse grids: f_h and f_H.
- MG/Opt: FAS applied to $\nabla f_h(x_h) = 0$ as optimization.

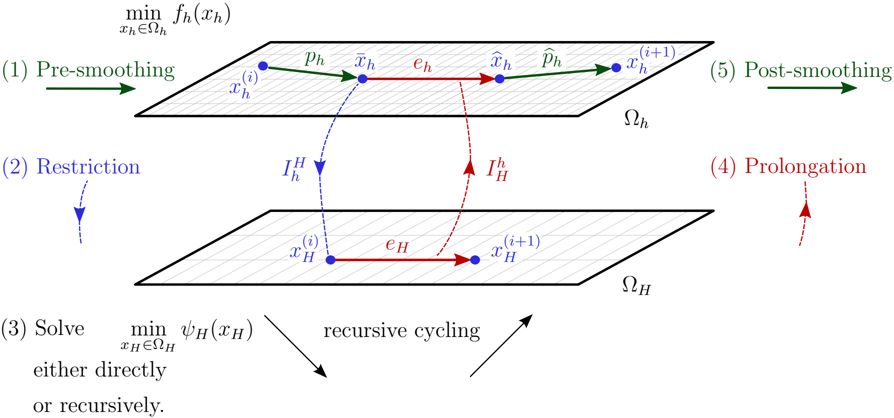
- Linear FAS modification to the coarse-grid correction equation

$$\psi_H(x_H^{(i)} + \boldsymbol{e_H}) := f_H(x_H^{(i)} + \boldsymbol{e_H}) - \langle x_H^{(i)} + \boldsymbol{e_H}, \nabla f_H(x_H^{(i)}) - I_h^H \nabla f_h(\bar{x}_h)\rangle.$$

- The correction $\boldsymbol{e_H}$ has to be smooth.
- Smoothers: cheap first-order optimization methods (SD, L-BFGS).
- Multi-Grid Line Search (MGLS): Modified line search to enforce convergence to local minima.

MG/Opt: [Nash 2000, Lewis/Nash 2005], MGLS: [Wen/Goldfarb 2009]

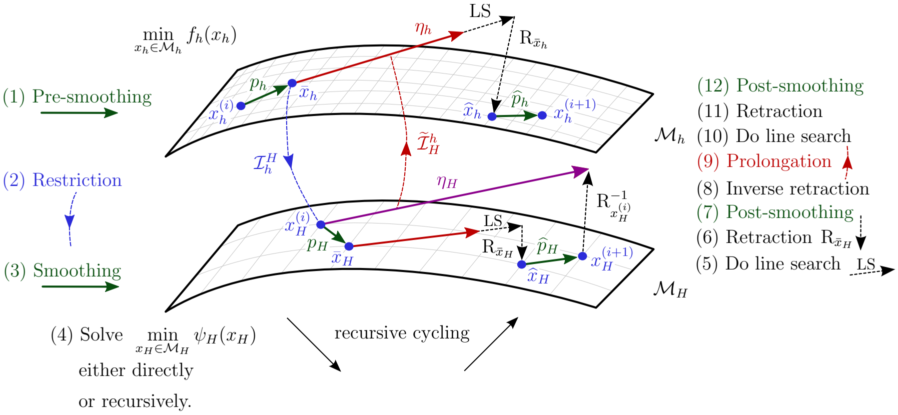
Multilevel optimization in Euclidean space/II



MG/Opt: [Nash 2000, Lewis/Nash 2005], MGLS: [Wen/Goldfarb 2009]

Generalization to Riemannian manifolds: RMGLS

Our contribution: extend MGLS to manifolds $\rightsquigarrow$ Riemannian Multigrid Line-Search (RMGLS).



RMGLS: [S./Vandereycken 2021]

Coarse-grid correction

Recall for MG/Opt: for fixed $x_H^{(i)}$, minimize for e_H the coarse-grid objective

$$\psi_H(x_H^{(i)} + \boldsymbol{e_H}) := f_H(x_H^{(i)} + \boldsymbol{e_H}) - \langle x_H^{(i)} + \boldsymbol{e_H}, \nabla f_H(x_H^{(i)}) - I_h^H \nabla f_h(\bar{x}_h)\rangle.$$

- To extend to manifolds, we interpret $\boldsymbol{e_H}$ as a tangent vector, the summation "+" as a retraction, and $\langle\cdot,\cdot\rangle$ as the Riemannian metric $g(\cdot,\cdot)$.
- The **linear modification** of the coarse-grid objective function

$$\widehat{\psi}_{x_H^{(i)}} : \mathrm{T}_{x_H^{(i)}}\mathcal{M}_H \to \mathbb{R},$$

is defined by

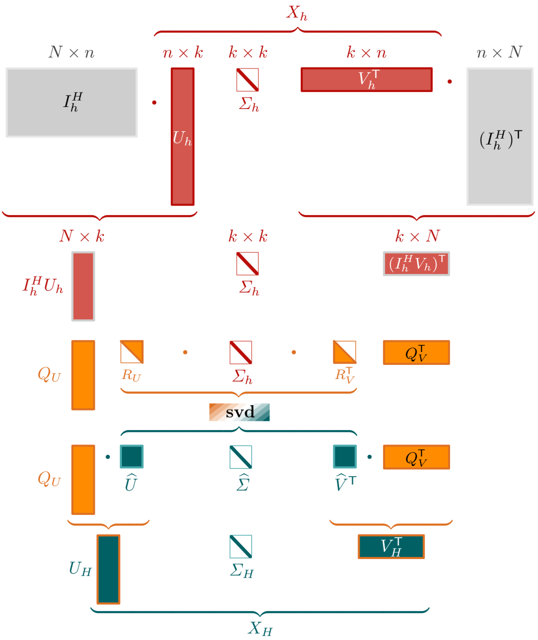
$$\widehat{\psi}_{x_H^{(i)}}(\eta_H) := f_H(R_{x_H^{(i)}}(\eta_H)) - \boldsymbol{g_{x_H^{(i)}}(\eta_H, \kappa_H)},$$

with retraction $R_{x_H^{(i)}}$, Riemannian metric $g_{x_H^{(i)}}$ and

$$\kappa_H = \operatorname{grad} f_H(x_H^{(i)}) - \widetilde{\mathcal{I}}_h^H(\operatorname{grad} f_h(\bar{x}_h)) \in \mathrm{T}_{x_H^{(i)}}\mathcal{M}_H.$$

Transfer operators

- Restriction $\mathcal{I}_h^H : \mathbb{R}^{n\times n} \to \mathbb{R}^{N\times N}$ and prolongation $\mathcal{I}_H^h : \mathbb{R}^{N\times N} \to \mathbb{R}^{n\times n}$.
- Exploit low rank of iterates and Riemannian gradients / tangent vectors!



IV. Numerical experiments

Lyapunov functional – problem statement

- Consider the minimization problem

$$\begin{cases} \min\limits_{w} \mathcal{F}(w(x,y)) = \int_{\Omega} \frac{1}{2}\|\nabla w(x,y)\|^2 - \gamma(x,y)\, w(x,y)\, \mathrm{d}x\, \mathrm{d}y \\ \quad \text{such that} \quad w = 0 \text{ on } \partial\Omega, \end{cases}$$

  where $\Omega = [0,1] \times [0,1]$ and $\gamma = 0$ on $\partial\Omega$.

- The variational derivative (Euclidean gradient) of $\mathcal{F}$ is

$$\frac{\delta \mathcal{F}}{\delta w} = -\Delta w - \gamma.$$

- Discretization gives the LHS of a Lyapunov equation

$$A_h W_h + W_h A_h - \Gamma_h,$$

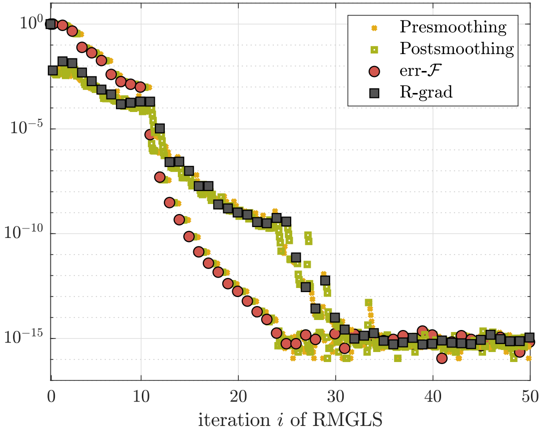
  where A_h is the discretized minus Laplacian.

- Linear problem, but typical problem for which low-rank methods work very well [Grasedyck 2004, Sabino 2006, Simoncini 2016]:

$$r = \mathcal{O}(\operatorname{rank}(\Gamma_h) \log(1/\varepsilon) \log \kappa(A_h)).$$

Lyapunov functional – typical convergence

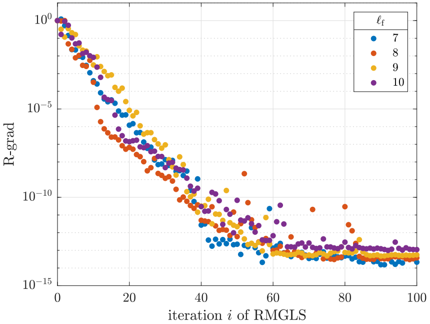
Example: V-cycle, finest level = 8 (about 250 000 gridpoints), coarsest level = 2, rank = 5, number of smoothing steps = 5.



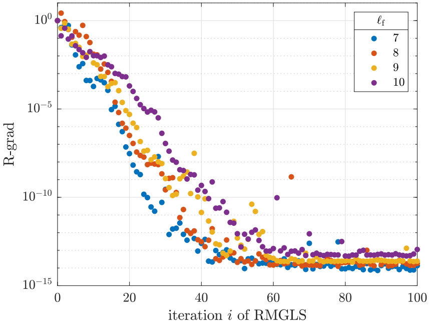
Lyapunov functional – mesh-independence

- V-cycle, coarsest level = 2.
- The sizes of the discretizations are 16 384 (●), 65 536 (●), 262 144 (●) and 1 048 576 (●).

rank $k = 5$

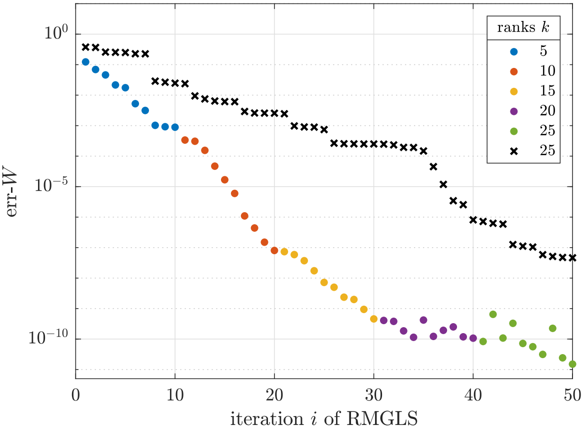


rank $k = 10$



Lyapunov functional – rank adaptivity

Example: V-cycle, coarsest level = 4, finest level = 10, rank is increased every 10 iterations.



Nonlinear PDE – problem statement

- Nonlinear PDE

$$\begin{cases} -\Delta w + \lambda w(w+1) - \gamma = 0 & \text{in } \Omega, \\ w = 0 & \text{on } \partial\Omega. \end{cases}$$

- **Prescribe** as exact solution (**numerical rank 9**):

$$w_{\mathrm{ex}} = \tfrac{1}{10}\sin(4\pi^2(x^2-x)(y^2-y)).$$

- We get the term

$$\gamma = -\Delta w_{\mathrm{ex}} + \lambda w_{\mathrm{ex}}(w_{\mathrm{ex}}+1).$$

- Obtain the variational problem

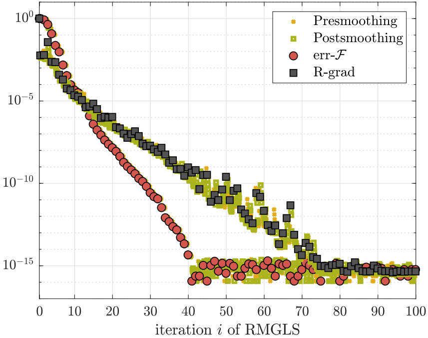
$$\begin{cases} \min\limits_{w} \mathcal{F}(w) = \displaystyle\int_{\Omega} \tfrac{1}{2}\|\nabla w\|^2 + \lambda w^2\left(\tfrac{1}{3}w + \tfrac{1}{2}\right) - \gamma\, w \,\mathrm{d}x\,\mathrm{d}y \\ \quad \text{such that} \quad w = 0 \text{ on } \partial\Omega. \end{cases}$$

Existing variational problem: [Henson 2003, Wen/Goldfarb 2009]

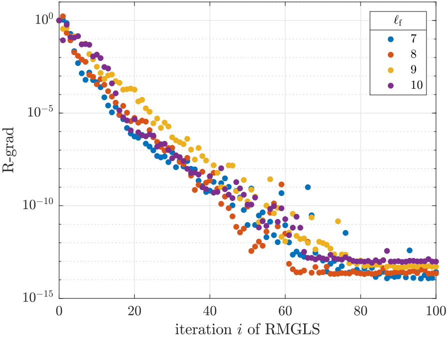
Nonlinear PDE – similar numerical experiment

Mesh-independent convergence

Error err-W with $\ell = 8$



Gradient R-grad with $k = 5$



Nonlinear PDE – Rank truncated Euclidean multilevel

Rank-truncated Euclidean multilevel (EML) vs RMGLS for different ranks.

In both cases, 8 smoothing steps and coarsest level 7 are used.

	level	size	EML		RMGLS		
			time (s)	$r(W_h^{(\mathrm{end})})$	**time (s)**	$\Vert\xi_h^{(\mathrm{end})}\Vert_{\mathrm{F}}$	$r(W_h^{(\mathrm{end})})$
rank 10	9	262 144	30	4.7324×10^{-7}	**21**	7.8437×10^{-13}	3.7321×10^{-7}
	10	1 048 576	123	3.4975×10^{-7}	**61**	4.0398×10^{-13}	1.8660×10^{-7}
	11	4 194 304	797	1.2826×10^{-5}	**153**	5.5800×10^{-13}	9.3301×10^{-8}
rank 15	9	262 144	107	7.4928×10^{-10}	**92**	2.0183×10^{-13}	4.2886×10^{-10}
	10	1 048 576	380	9.6225×10^{-10}	**207**	6.5306×10^{-13}	2.6044×10^{-10}
	11	4 194 304	3 113	4.3682×10^{-10}	**532**	1.3610×10^{-13}	8.3563×10^{-11}

N.B.: “size” means total number of gridpoints in **both** spatial directions ($= 2^{2\ell}$).

Conclusion and outlook

This talk:

$\rightsquigarrow$ Riemannian Multigrid Line Search for Low-Rank Problems, M. Sutti and B. Vandereycken, SIAM J. Sci. Comput., 43(3), A1803–A1831, 2021.

- New algorithm with low-rank approximations to solve large-scale optimization problems.
- Optimization on $\mathcal{M}_k$ using multilevel idea of [Wen/Goldfarb 2009].

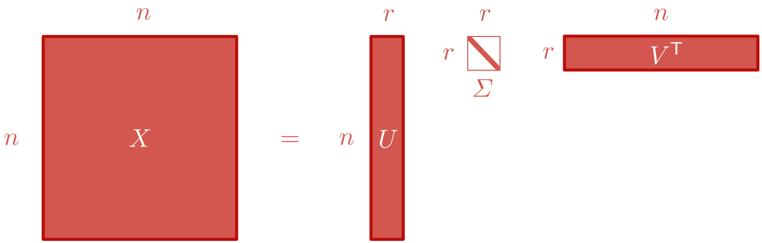
Further research:

- Extend the convergence proof from [Wen/Goldfarb 2009].
- Generalization to tensor problems, coming from high-dimensional PDEs (e.g., Schrödinger equation, Black–Scholes equation…).
- RMGLS for other manifolds (other than $\mathcal{M}_k$).

Thank you for your attention!

V. Bonus material
RMGLS

Motivation for the low-rank format/2



- Storing a dense 5000×5000 matrix in double precision takes $5000^2 \times 8/2^{20} \approx 191$ MB.
  - If it has rank 10 and we store only its factors, it takes $(2 \times 5000 \times 10 + 10) \times 8/2^{20} = 0.76$ kB.
  - If it has rank 100 and we store only its factors, it takes $(2 \times 5000 \times 100 + 100) \times 8/2^{20} = 7.63$ MB.
- For a matrix stored in the dense format, the storage complexity grows as n^2, but if the matrix is stored in low-rank format, then the storage grows as nr.

Line-search (LS) method

⇝ **How to calculate t_k?**

- **Exact** line search (LS):

$$\min_{t \geq 0} f(x_k + t\eta_k)$$

  - t_k^{EX} is the unique minimizer if f is strictly convex.
  - Can sometimes be computed. Good for theory.

- In practice, for generic f, we do not use exact LS. Replace exact LS with something computationally cheaper, but still effective.

  ⇝ **Armijo line-search** (also known as Armijo backtracking, Armijo condition, sufficient decrease condition, …).

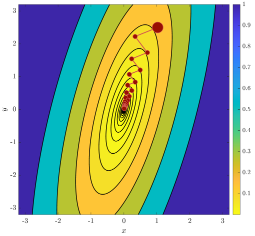
Armijo line-search technique: [Armijo 1966]

Steepest descent on a manifold

- Steepest descent in $\mathbb{R}^n$ is based on the update formula

$$x_{k+1} = x_k + t_k \eta_k,$$

where $t_k \in \mathbb{R}$ is the step size and $\eta_k \in \mathbb{R}^n$ is the search direction.



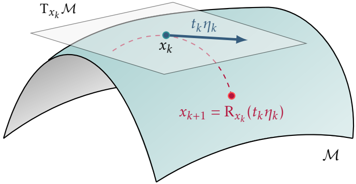
$\rightsquigarrow$ On nonlinear manifolds:

- η_k will be a tangent vector to $\mathcal{M}$ at x_k, i.e., $\eta_k \in \mathrm{T}_{x_k}\mathcal{M}$.

  Remark: If $\eta_k = -\operatorname{grad} f(x_k)$, we get the **Riemannian steepest descent**.

- Search along a curve in $\mathcal{M}$ whose tangent vector at $t_k = 0$ is η_k.

$\rightsquigarrow$ Retraction.



Steepest descent on a manifold (reprise)

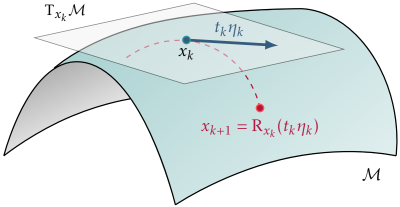
Steepest descent on manifolds is based on the update formula

$$x_{k+1} = \mathrm{R}_{x_k}(t_k \eta_k),$$

where $t_k \in \mathbb{R}$ and $\eta_k \in \mathrm{T}_{x_k}\mathcal{M}$.

Recipe for constructing the steepest descent method on a manifold:

- Choose a retraction R (previous slide).
- Select a search direction η_k (the anti-gradient $\eta_k = -\operatorname{grad} f(x_k)$).
- Select a step length t_k (with a line-search technique).



Retractions on embedded submanifolds

Let $\mathcal{M}$ be an embedded submanifold of a vector space $\mathcal{E}$. Thus $\mathrm{T}_x\mathcal{M}$ is a linear subspace of $\mathrm{T}_x\mathcal{E} \simeq \mathcal{E}$. Since $x \in \mathcal{M} \subseteq \mathcal{E}$ and $\xi \in \mathrm{T}_x\mathcal{M} \subseteq \mathrm{T}_x\mathcal{E} \simeq \mathcal{E}$, with little abuse of notation we write $x + \xi \in \mathcal{E}$.

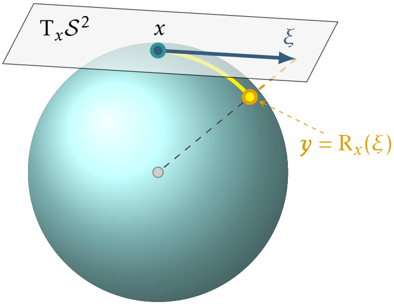
$\rightsquigarrow$ General recipe to define a retraction $\mathrm{R}_x(\xi)$ for embedded submanifolds:

- Move along ξ to get to $x + \xi$ in $\mathcal{E}$.
- Map $x + \xi$ back to $\mathcal{M}$. For matrix manifolds, use matrix decompositions.

Example. Let $\mathcal{M} = S^{n-1}$, then the retraction at $x \in S^{n-1}$ is

$$\mathrm{R}_x(\xi) = \frac{x + \xi}{\|x + \xi\|},$$

defined for all $\xi \in \mathrm{T}_x S^{n-1}$. $\mathrm{R}_x(\xi)$ is the point on S^{n-1} that minimizes the distance to $x + \xi$.



One RMGLS iteration starting at $x_h^{(i)}$ to minimize f_h.

(1) **Pre-smoothing**: $\bar{x}_h = \mathrm{SMOOTH}^{\nu_1}(x_h^{(i)}, f_h)$

(2) **Coarse-grid correction**:

(a) **Restrict** to the coarse manifold: $x_H^{(i)} = \mathcal{I}_h^H(\bar{x}_h)$

(b) Compute the **linear correction term**:

$$\kappa_H = \operatorname{grad} f_H(x_H^{(i)}) - \widetilde{\mathcal{I}}_h^H(\operatorname{grad} f_h(\bar{x}_h))$$

(c) Define the **coarse-grid objective function**

$$\psi_H(x_H) = f_H(x_H) - g_{x_H^{(i)}}(R^{-1}_{x_H^{(i)}}(x_H), \kappa_H)$$

(d) Compute an **approximate minimizer** $x_H^{(i+1)}$ starting at $x_H^{(i)}$ to minimize ψ_H using either

- a Riemannian trust-region method (if $\mathcal{M}_H$ is small)
- one recursive RMGLS iteration (otherwise)

(e) Compute the **coarse-grid correction**: $\eta_H = R^{-1}_{x_H^{(i)}}(x_H^{(i+1)})$

(f) **Interpolate** to the fine manifold: $\eta_h = \widetilde{\mathcal{I}}_H^h(\eta_H)$

(g) Compute the **corrected approximation** on the fine manifold:

$$\widehat{x_h} = R_{\bar{x}_h}(\alpha^* \eta_h) \quad \text{with } \alpha^* \text{ obtained from line search}$$

(3) **Post-smoothing**: $x_h^{(i+1)} = \mathrm{SMOOTH}^{\nu_2}(\widehat{x_h}, f_h)$

Smoothers

- Many options for smoothers, but they need to be compatible with optimization, like SD or L-BFGS.
- Point smoother, but also line smoothers are possible using cheap preconditioning or quasi-Newton.
- We take half the step size in steepest descent. Similar to Jacobi iteration as smoother, i.e., we do line search, get α and then set $\alpha \leftarrow \alpha/2$.
- For isotropic problems, a small number (5) of **steepest descent** steps for Riemannian manifolds suffices. Steepest descent plays the role of a smoother.

[Wen/Goldfarb 2009, Vandereycken 2010]

“LYAP” variational problem

		Rank 5						Rank 10					
Prec.	size	10	11	12	13	14	15	10	11	12	13	14	15
No	n_{outer}	51	54	61	59	162	92	300	103	61	63	62	59
	$\sum n_{\text{inner}}$	4561	9431	21066	36556	30069	30096	27867	30025	33818	45760	44467	38392
	$\max n_{\text{inner}}$	1801	3191	7055	9404	1194	1851	2974	3385	8894	24367	24537	25013
Yes	n_{outer}	41	45	50	52	56	60	44	64	62	53	56	56
	$\sum n_{\text{inner}}$	44	45	50	52	56	60	69	104	82	60	69	56
	$\max n_{\text{inner}}$	4	1	1	1	1	1	9	9	8	8	8	1

- Stopping criterion: maximum number of outer iterations $n_{\text{max outer}} = 300$. The inner solver is stopped when $\sum n_{\text{inner}}$ first exceeds $30\,000$.
- Impressive reduction in the number of iterations of the inner solver.
- n_{outer} and $\sum n_{\text{inner}}$ depend (quite mildly) on size, while $\max n_{\text{inner}}$ is basically constant.

An example of factorized gradient

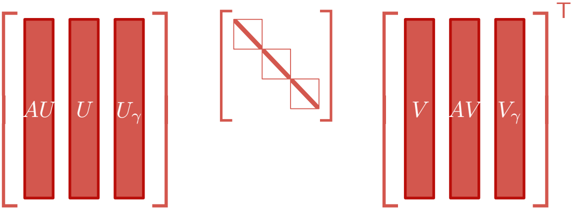
- "LYAP" functional: $\mathcal{F}(w(x,y)) = \int_\Omega \frac{1}{2}\|\nabla w(x,y)\|^2 - \gamma(x,y)\,w(x,y)\,\mathrm{d}x\,\mathrm{d}y$.
- The gradient of $\mathcal{F}$ is the variational derivative $\frac{\delta\mathcal{F}}{\delta w} = -\Delta w - \gamma$.
- The discretized Euclidean gradient in matrix form is given by

$$G = AW + WA - \Gamma.$$

  with A is the second-order periodic finite difference differentiation matrix.
- The first-order optimality condition $G = AW + WA - \Gamma = 0$ is a Lyapunov (or Sylvester) equation.

$\rightsquigarrow$ Factorized Euclidean gradient:

$$G = \begin{bmatrix} AU & U & U_\gamma \end{bmatrix} \operatorname{blkdiag}\left(\Sigma,\ \Sigma,\ \Sigma_\gamma\right) \begin{bmatrix} V & AV & V_\gamma \end{bmatrix}^\top.$$

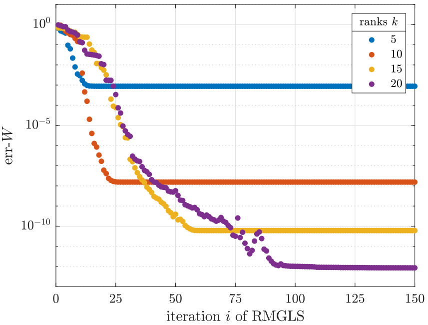


Lyapunov – importance of line search

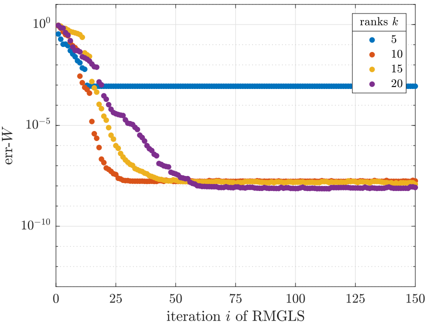
- Compare LS: standard Wolfe vs Hager-Zhang (developed for nonlinear CG in $\mathbb{R}^n$ but can be extended to general manifolds like $\mathcal{M}_k$).
- The relative error in Frobenius norm of the low-rank approximation:

$$\text{err-}W(i) \coloneqq \|W_h^{(i)} - W_h^{(*)}\|_\mathrm{F} / \|W_h^{(*)}\|_\mathrm{F}.$$

Hager–Zhang



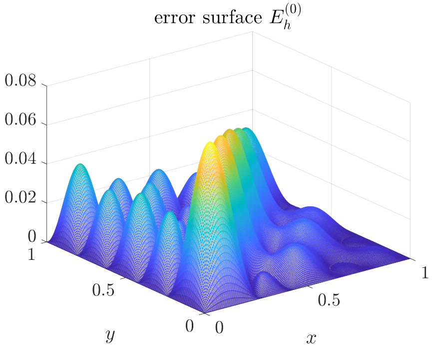
weak Wolfe

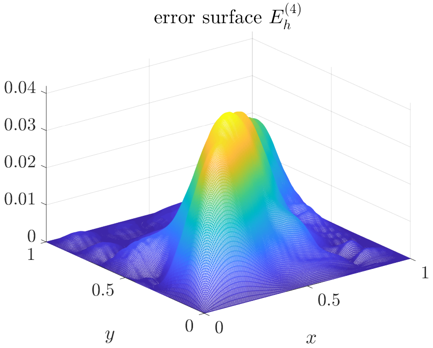


Lyapunov – smoothness of the error

Example: V-cycle, finest level = 8 (about 250 000 gridpoints), coarsest level = 5, rank = 5, number of smoothing steps = 5.

$$E_h^{(i)} := |W_h^{(i)} - W_h^{(*)}|.$$

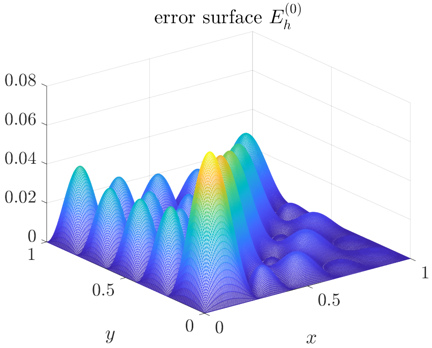




Nonlinear PDE – smoothness of the error

Example: V-cycle, finest level = 8 (about 250 000 gridpoints), coarsest level = 5, rank = 5, number of smoothing steps = 5.

$$E_h^{(i)} := |W_h^{(i)} - W_h^{(*)}|.$$



VI. Bonus material
Riemannian Hager–Zhang line search

A motivating example: Quadratic cost function

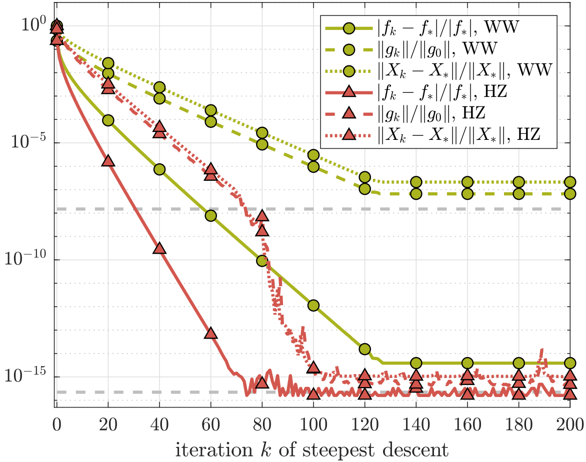
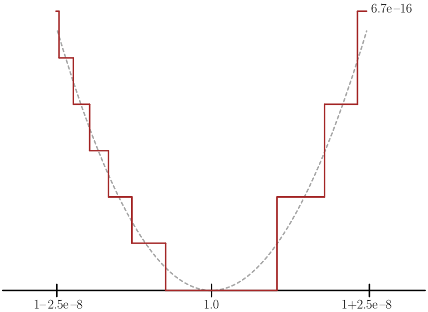


Figure: Convergence behavior of line search with weak Wolfe (WW) or Hager–Zhang (HZ) when applied to a quadratic function f. The objective function is denoted by f_k and the gradient by g_k. The horizontal dashed lines indicate $\sqrt{\varepsilon_{\text{mach}}}$ and $\varepsilon_{\text{mach}}$.

HZLS/I

- Line searches usually have sufficient decrease as **stopping criteria** (Wolfe, Armijo).
- In finite precision, this condition cannot be satisfied close to the local minimum.
- One can only expect the minimum to be determined within $\sqrt{\varepsilon_{\mathrm{mach}}}$.



- Use **approximate Wolfe conditions** instead, based on the derivative of the objective function. Accurate within $\varepsilon_{\mathrm{mach}}$.
- Reason: Finding the zero of the derivative f' of an approximate quadratic f is numerically more accurate than minimizing f.
- In principle: simply use MATLAB's `fzero` on f' but this is too expensive for local optimization methods.

[Hager/Zhang 2005–2006]

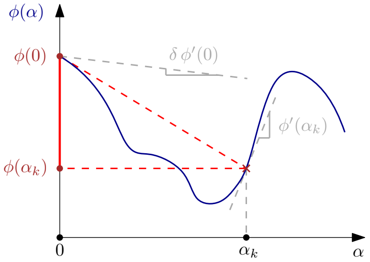
HZLS/II

- (weak) **Wolfe conditions** in terms of $\phi(\alpha) := f(x_k + \alpha\, d_k)$

$$\phi(\alpha_k) - \phi(0) \leqslant \alpha_k\, \delta\, \phi'(0),$$

$$\phi'(\alpha_k) \geqslant \sigma\, \phi'(0),$$

$$0 < \delta \leqslant \sigma < 1.$$



- Around a local minimum, the 1st condition is difficult to satisfy since $\phi(\alpha) \approx \phi(0)$.
- **Approximate Wolfe conditions**

$$\boldsymbol{(2\delta - 1)\,\phi'(0) \geqslant \phi'(\alpha_k)} \geqslant \sigma\, \phi'(0), \qquad 0 < \delta < 0.5, \quad \delta \leqslant \sigma < 1.$$

- Only the 1st inequality is an approximation of the original conditions, so it would be more appropriate to talk about **approximate Armijo**.

HZLS/III

- Why **approximate**?
- Build a special quadratic interpolant $q(\alpha)$ of $\phi(\alpha)$ such that the finite difference (FD) quotient in the 1^{st} Wolfe condition can be approximated by

$$\frac{\phi(\alpha_k) - \phi(0)}{\alpha_k} \approx \frac{q(\alpha_k) - q(0)}{\alpha_k} = \frac{\phi'(\alpha_k) + \phi'(0)}{2}.$$

- This gives the inequality

$$(2\delta - 1)\,\phi'(0) \geqslant \phi'(\alpha_k).$$

- With this approximation we circumvent the numerical errors inherent in the original FD quotient.
- HZLS can be generalized to the **Riemannian framework** using the derivative of the retraction R'_X. Seems restrictive but, in practice, it is not: automatic differentiation and we are allowed to change retraction.

Rayleigh quotient on the sphere

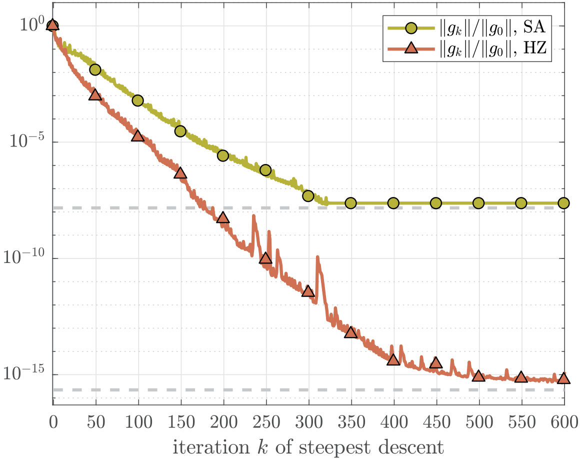


Figure: Convergence behavior of steepest descent with standard Armijo (SA) or Hager–Zhang (HZ) line search when applied to the Rayleigh quotient on the sphere. The gradient is denoted by g_k. The horizontal dashed lines indicate $\sqrt{\varepsilon_{\mathrm{mach}}}$ and $\varepsilon_{\mathrm{mach}}$.

Brockett cost function on the Stiefel manifold

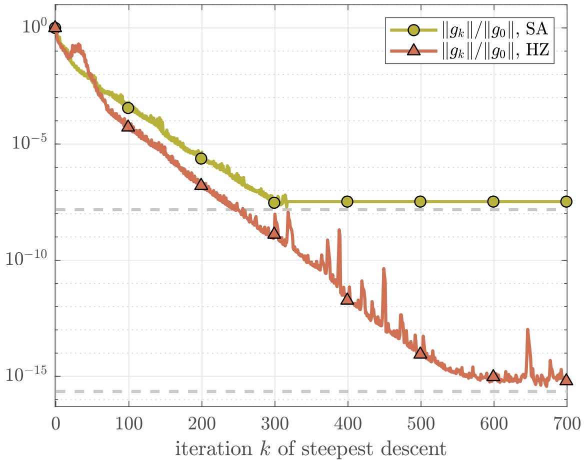


Figure: Convergence behavior of steepest descent with standard Armijo (SA) or Hager–Zhang (HZ) line search when applied to the Brockett cost function on the Stiefel manifold. The gradient is denoted by g_k. The horizontal dashed lines indicate $\sqrt{\varepsilon_{\mathrm{mach}}}$ and $\varepsilon_{\mathrm{mach}}$.