Extending GMRES to Nonlinear Optimization, with Application to Tensor Optimization

UNIVERSITY OF WATERLOO

uwaterloo.ca

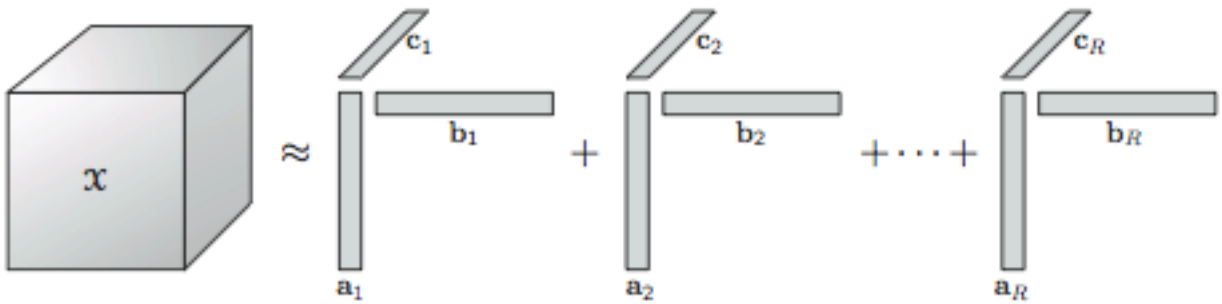
Hans De Sterck

Department of Applied Mathematics
University of Waterloo

12th Copper Mountain Conference on Iterative Methods, 2012

1. introduction

- tensor = *N*-dimensional array
- *N*=3:

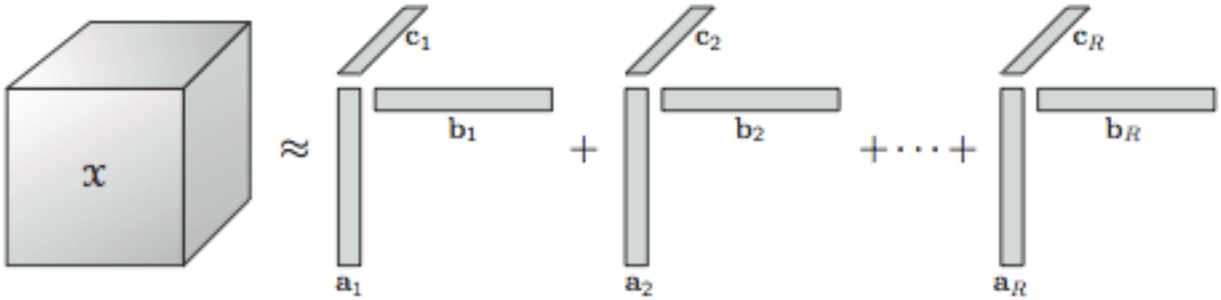


(from "Tensor Decompositions and Applications", Kolda and Bader, SIAM Rev., 2009 [1])

- canonical decomposition: decompose tensor in sum of *R* rank-one terms (approximately)

UNIVERSITY OF WATERLOO

introduction



(from "Tensor Decompositions and Applications", Kolda and Bader, SIAM Rev., 2009 [1])

OPTIMIZATION PROBLEM

given tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, find rank-$R$ canonical tensor $\mathcal{A}_R \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ that minimizes

$$f(\mathcal{A}_R) = \frac{1}{2} \|\mathcal{T} - \mathcal{A}_R\|_F^2.$$

FIRST-ORDER OPTIMALITY EQUATIONS

$$\nabla f(\mathcal{A}_R) = \mathbf{g}(\mathcal{A}_R) = 0.$$

(problem is non-convex, multiple (local) minima, solution may not exist, ... ; but smooth, and assume there is a local minimum)

(de Silva and Lim, SIMAX, 2009)

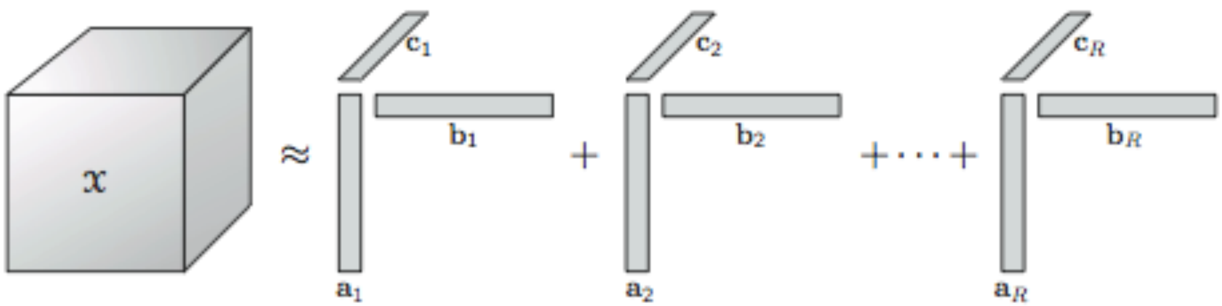
UNIVERSITY OF WATERLOO

link with singular value decomposition

- SVD of $A \in \mathbb{R}^{m \times n}$ $\quad m \geq n$

$$A = U\,\Sigma\,V^t = \sigma_1\,u_1\,v_1^T + \ldots + \sigma_n\,u_n\,v_n^T$$

- canonical decomposition of tensor



(from “Tensor Decompositions and Applications”, Kolda and Bader, SIAM Rev., 2009 [1])

a difference with the SVD

truncated SVD is best rank-*R* approximation:

$$A = \sigma_1 u_1 v_1^T + \ldots + \sigma_R u_R v_R^T + \sigma_{R+1} u_{R+1} v_{R+1}^T + \ldots + \sigma_n u_n v_n^T$$

$$\underset{B \text{ with rank } \leq R}{\arg\min} \|A - B\|_F = \sigma_1 u_1 v_1^T + \ldots + \sigma_R u_R v_R^T$$

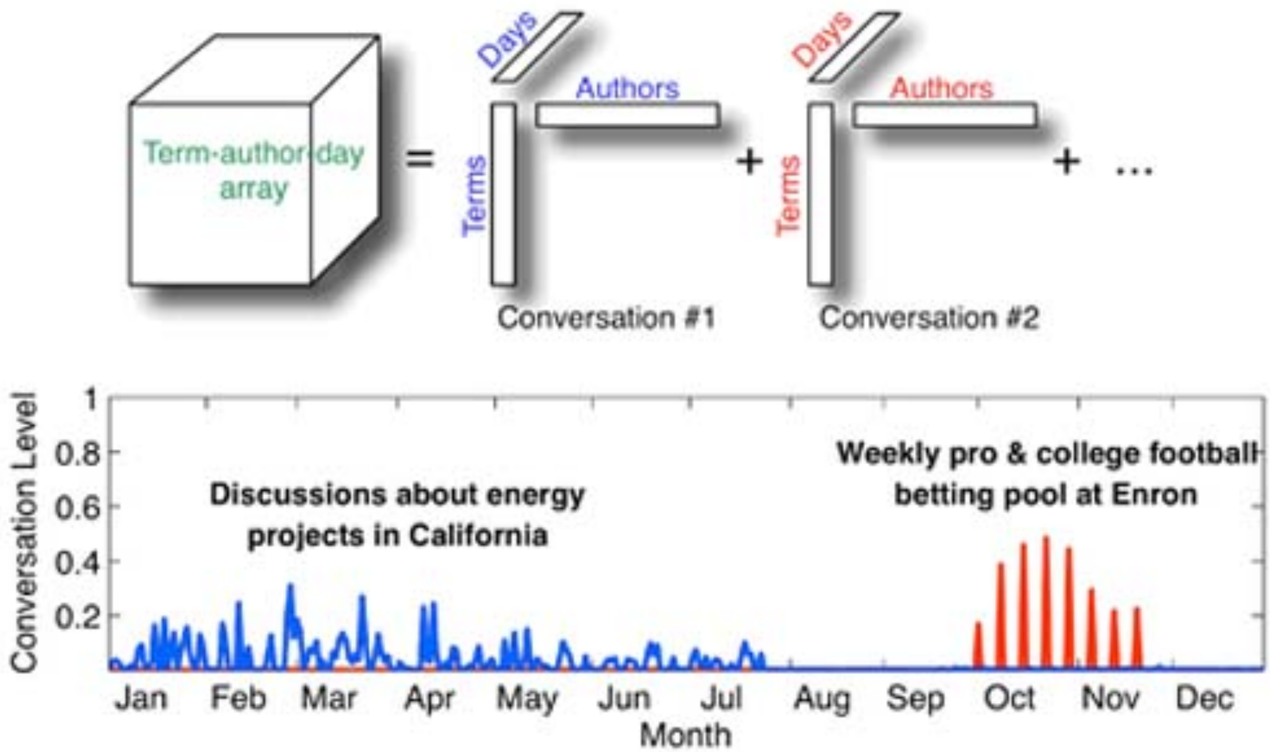
BUT best rank-*R* tensor cannot be obtained by truncation: different optimization problems for different *R*!

given tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, find rank-$R$ canonical tensor $\mathcal{A}_R \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ that minimizes

$$f(\mathcal{A}_R) = \frac{1}{2} \|\mathcal{T} - \mathcal{A}_R\|_F^2.$$

2. tensor approximation applications

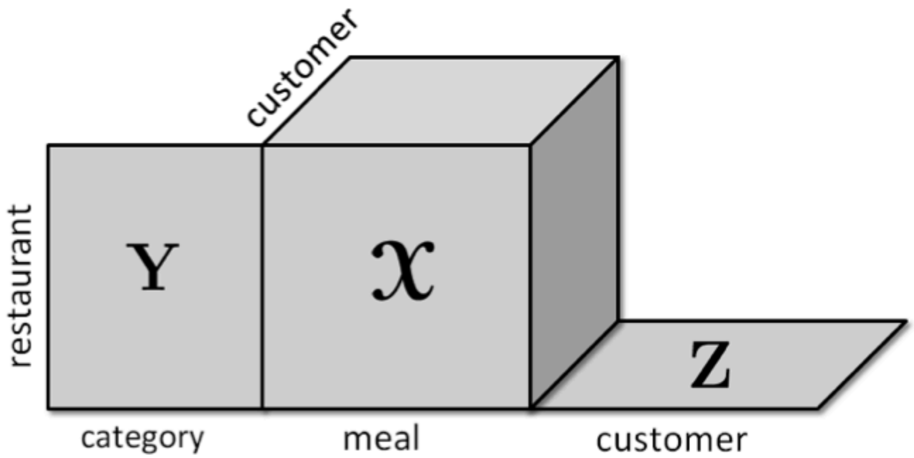
(1) “Discussion Tracking in Enron Email Using PARAFAC” by Bader, Berry and Browne (2008) (sparse, nonnegative)

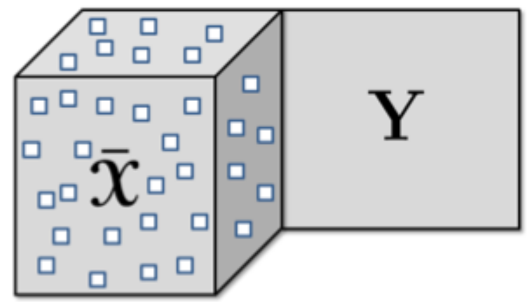


UNIVERSITY OF WATERLOO

tensor approximation applications

(2) “All-at-once Optimization for Coupled Matrix and Tensor Factorizations” by Acar, Kolda and Dunlavy (2011)





$$\left\| \mathcal{W} * \left(\mathcal{X} - [\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!] \right) \right\|^2 + \frac{1}{2} \left\| \mathbf{Y} - \mathbf{A}^{(n)} \mathbf{V}^{\mathsf{T}} \right\|^2$$

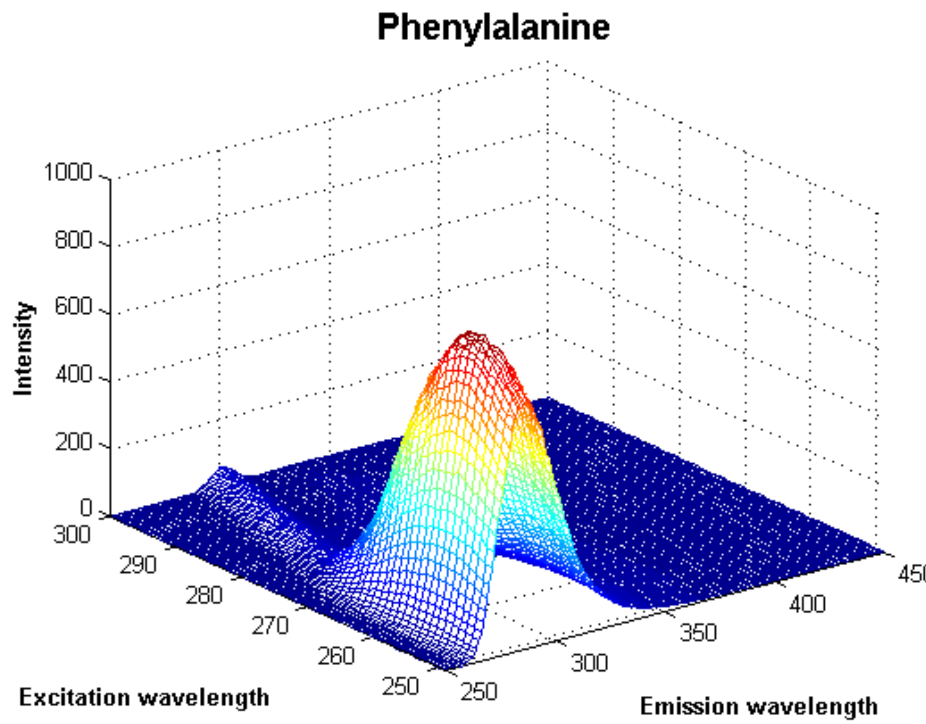
$$f(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{V}) = \| \mathcal{X} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \|^2 + \left\| \mathbf{Y} - \mathbf{A} \mathbf{V}^{\mathsf{T}} \right\|^2$$

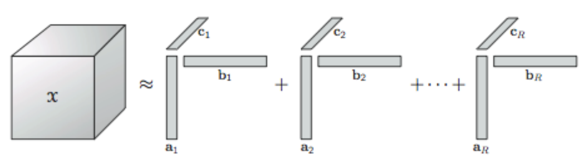
UNIVERSITY OF WATERLOO

tensor approximation applications

(3) chemometrics: analyze spectrofluorometer data (dense) (Bro et al., http://www.models.life.ku.dk/nwaydata1)

- 5 x 201 x 61 tensor: 5 samples (with different mixtures of three amino acids), 61 excitation wavelengths, 201 emission wavelengths
- goal: recover emission spectra of the three amino acids (to determine what was in each sample, and in which concentration)
- also: psychometrics, ...





(from [1])

UNIVERSITY OF WATERLOO

3. alternating least squares (ALS)

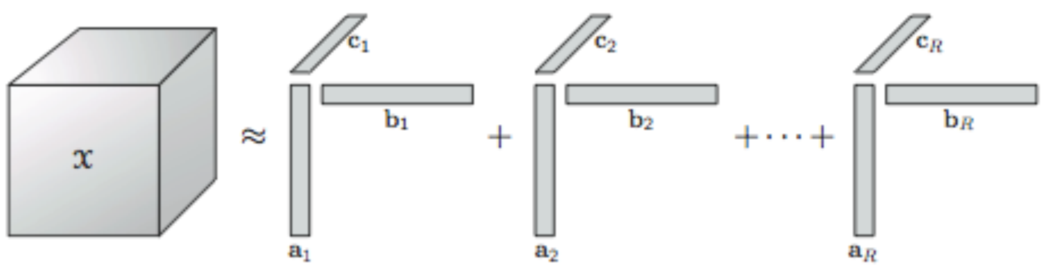
$$f(\mathcal{A}_R) = \frac{1}{2} \left\| \mathcal{T} - \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ a_r^{(3)} \right\|_F^2$$

(1) freeze all $a_r^{(2)}$, $a_r^{(3)}$, compute optimal $a_r^{(1)}$ via a least-squares solution (linear, overdetermined)

(2) freeze $a_r^{(1)}$, $a_r^{(3)}$, compute $a_r^{(2)}$

(3) freeze $a_r^{(1)}$, $a_r^{(2)}$, compute $a_r^{(3)}$

- repeat



(from [1])

UNIVERSITY OF WATERLOO

alternating least squares (ALS)

$$f(\mathcal{A}_R) = \frac{1}{2}\left\| \mathcal{T} - \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ a_r^{(3)} \right\|_F^2$$

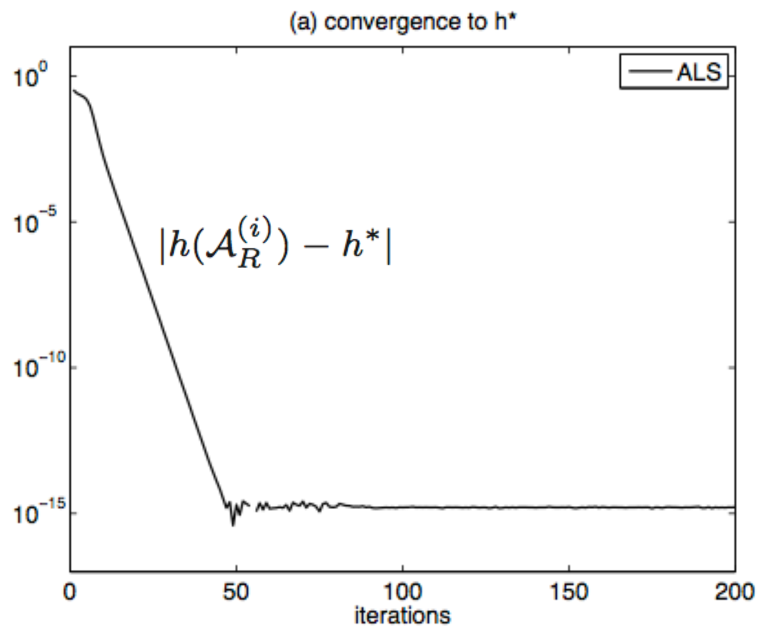
- ALS is monotone
- ALS is sometimes fast, but can also be extremely slow (depending on problem and initial condition)
- ALS is block nonlinear Gauss-Seidel

UNIVERSITY OF WATERLOO

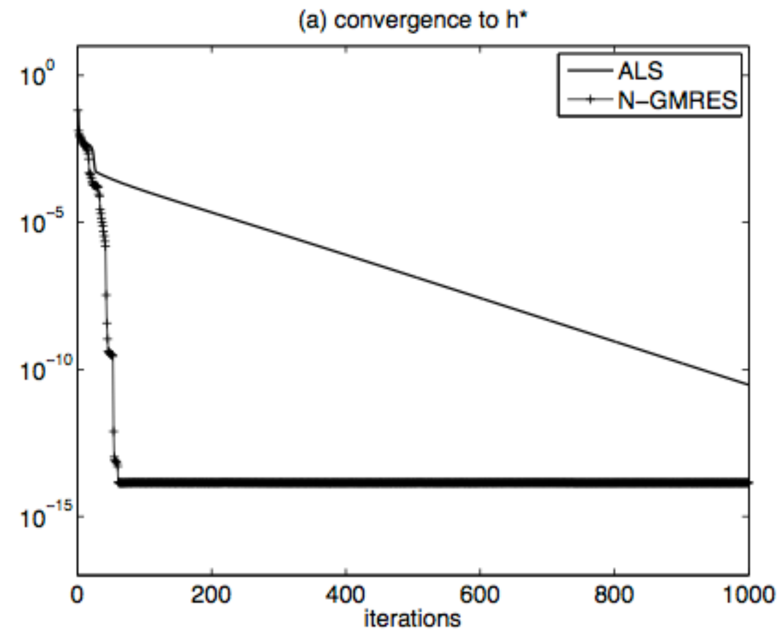
alternating least squares (ALS)

$$f(\mathcal{A}_R) = \frac{1}{2}\left\| \mathcal{T} - \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ a_r^{(3)} \right\|_F^2 \qquad h(\mathcal{A}_R^{(i)}) = \frac{\|\mathcal{T} - \mathcal{A}_R^{(i)}\|_F}{\|\mathcal{T}\|_F}$$

fast case



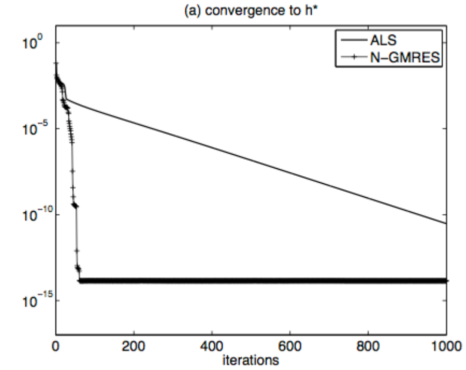
slow case



(we used Matlab with Tensor Toolbox (Bader and Kolda) and Poblano Toolbox (Dunlavy et al.) for all computations)

UNIVERSITY OF WATERLOO

alternating least squares (ALS)

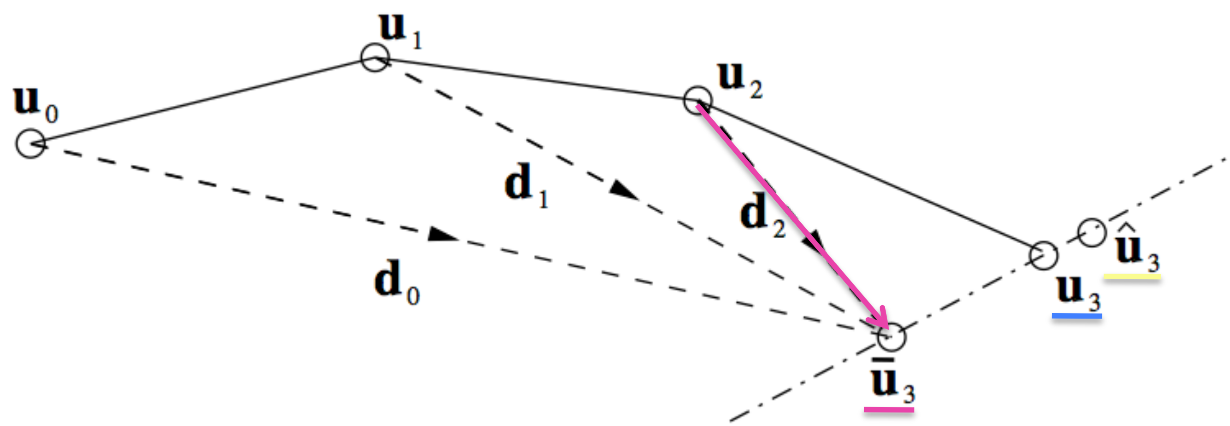


$$f(\mathcal{A}_R) = \frac{1}{2} \left\| \mathcal{T} - \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ a_r^{(3)} \right\|_F^2$$

- for linear systems $\mathbf{A}\,\mathbf{u} = \mathbf{b}$, when a simple iterative method is slow, we accelerate it with
  - GMRES
  - CG, multigrid, etc.
- the simple iterative method is called the ‘preconditioner’
- BUT: for optimization problems, general approaches to accelerate simple iterative methods are uncommon (do not exist?)
- let’s try to accelerate ALS for the tensor optimization problem (this talk: GMRES, Killian’s talk: multigrid)
- issues: nonlinear, optimization context

UNIVERSITY OF WATERLOO

4. nonlinear GMRES acceleration of ALS



Algorithm 1: N-GMRES optimization algorithm (window size w)

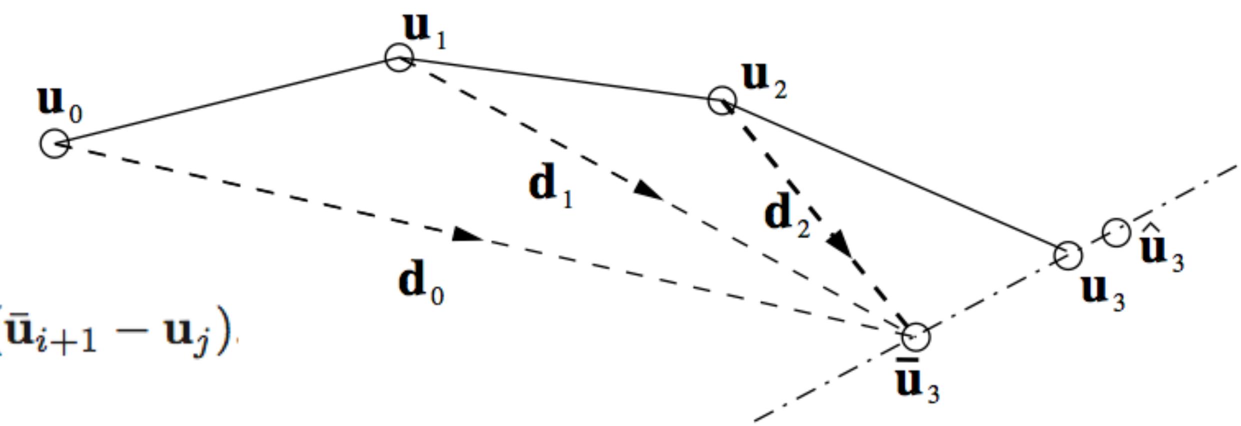
Input: w initial iterates $\mathbf{u}_0, \ldots, \mathbf{u}_{w-1}$.

$i = w - 1$
repeat
  STEP I: *(generate preliminary iterate by one-step update process $M(.)$)*
    $\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$
  STEP II: *(generate accelerated iterate by nonlinear GMRES step)*
    $\hat{\mathbf{u}}_{i+1} = \text{gmres}(\mathbf{u}_{i-w+1}, \ldots, \mathbf{u}_i; \bar{\mathbf{u}}_{i+1})$
  STEP III: *(generate new iterate by line search process)*
    $\mathbf{u}_{i+1} = \text{linesearch}(\bar{\mathbf{u}}_{i+1} + \beta(\hat{\mathbf{u}}_{i+1} - \bar{\mathbf{u}}_{i+1}))$
  $i = i + 1$
until *convergence criterion satisfied*

(Moré-Thuente line search, satisfies Wolfe conditions)

UNIVERSITY OF WATERLOO

step II: N-GMRES acceleration: $\nabla f(\mathcal{A}_R) = \mathbf{g}(\mathcal{A}_R) = 0$



$$\hat{\mathbf{u}}_{i+1} = \bar{\mathbf{u}}_{i+1} + \sum_{j=0}^{i} \alpha_j \left(\bar{\mathbf{u}}_{i+1} - \mathbf{u}_j\right)$$

$$\mathbf{g}(\hat{\mathbf{u}}_{i+1}) \approx \mathbf{g}(\bar{\mathbf{u}}_{i+1}) + \sum_{j=0}^{i} \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\bar{\mathbf{u}}_{i+1}} \alpha_j \left(\bar{\mathbf{u}}_{i+1} - \mathbf{u}_j\right)$$

$$\approx \mathbf{g}(\bar{\mathbf{u}}_{i+1}) + \sum_{j=0}^{i} \alpha_j \left(\mathbf{g}(\bar{\mathbf{u}}_{i+1}) - \mathbf{g}(\mathbf{u}_j)\right)$$

find coefficients $(\alpha_0, \ldots, \alpha_i)$ that minimize

$$\|\mathbf{g}(\bar{\mathbf{u}}_{i+1}) + \sum_{j=0}^{i} \alpha_j \left(\mathbf{g}(\bar{\mathbf{u}}_{i+1}) - \mathbf{g}(\mathbf{u}_j)\right)\|_2.$$

history of nonlinear acceleration mechanism for nonlinear systems (steps I and II)

STEP I: *(generate preliminary iterate by one-step update process $M(.)$)*

$$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$$

STEP II: *(generate accelerated iterate by nonlinear GMRES step)*

$$\hat{\mathbf{u}}_{i+1} = \text{gmres}(\mathbf{u}_{i-w+1}, \ldots, \mathbf{u}_i; \bar{\mathbf{u}}_{i+1})$$

STEP III: *(generate new iterate by line search process)*

$$\mathbf{u}_{i+1} = \text{linesearch}(\bar{\mathbf{u}}_{i+1} + \beta(\hat{\mathbf{u}}_{i+1} - \bar{\mathbf{u}}_{i+1}))$$

$$\nabla f(\mathbf{u}) = \mathbf{g}(\mathbf{u}) = 0$$

$$\hat{\mathbf{u}}_{i+1} = \bar{\mathbf{u}}_{i+1} + \sum_{j=0}^{i} \alpha_j \left(\bar{\mathbf{u}}_{i+1} - \mathbf{u}_j\right)$$

find coefficients $(\alpha_0, \ldots, \alpha_i)$ that minimize

$$\|\mathbf{g}(\bar{\mathbf{u}}_{i+1}) + \sum_{j=0}^{i} \alpha_j \left(\mathbf{g}(\bar{\mathbf{u}}_{i+1}) - \mathbf{g}(\mathbf{u}_j)\right)\|_2.$$

- Washio and Oosterlee, ETNA, 1997
  (FAS multigrid for nonlinear PDEs)
- GMRES, Saad and Schultz, 1986
  (also flexible GMRES, Saad, 1993)
- Anderson mixing, 1965; DIIS (direct inversion in the iterative subspace), Pulay, 1980

UNIVERSITY OF WATERLOO

history of nonlinear acceleration mechanism for nonlinear systems (steps I and II)

STEP I: *(generate preliminary iterate by one-step update process $M(.)$)*

$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$

STEP II: *(generate accelerated iterate by nonlinear GMRES step)*

$\hat{\mathbf{u}}_{i+1} = \text{gmres}(\mathbf{u}_{i-w+1}, \ldots, \mathbf{u}_i; \bar{\mathbf{u}}_{i+1})$

STEP III: *(generate new iterate by line search process)*

$\mathbf{u}_{i+1} = \text{linesearch}(\bar{\mathbf{u}}_{i+1} + \beta(\hat{\mathbf{u}}_{i+1} - \bar{\mathbf{u}}_{i+1}))$

$$\nabla f(\mathbf{u}) = \mathbf{g}(\mathbf{u}) = 0$$

$$\hat{\mathbf{u}}_{i+1} = \bar{\mathbf{u}}_{i+1} + \sum_{j=0}^{i} \alpha_j \left(\bar{\mathbf{u}}_{i+1} - \mathbf{u}_j\right)$$

find coefficients $(\alpha_0, \ldots, \alpha_i)$ that minimize

$$\left\|\mathbf{g}(\bar{\mathbf{u}}_{i+1}) + \sum_{j=0}^{i} \alpha_j \left(\mathbf{g}(\bar{\mathbf{u}}_{i+1}) - \mathbf{g}(\mathbf{u}_j)\right)\right\|_2.$$

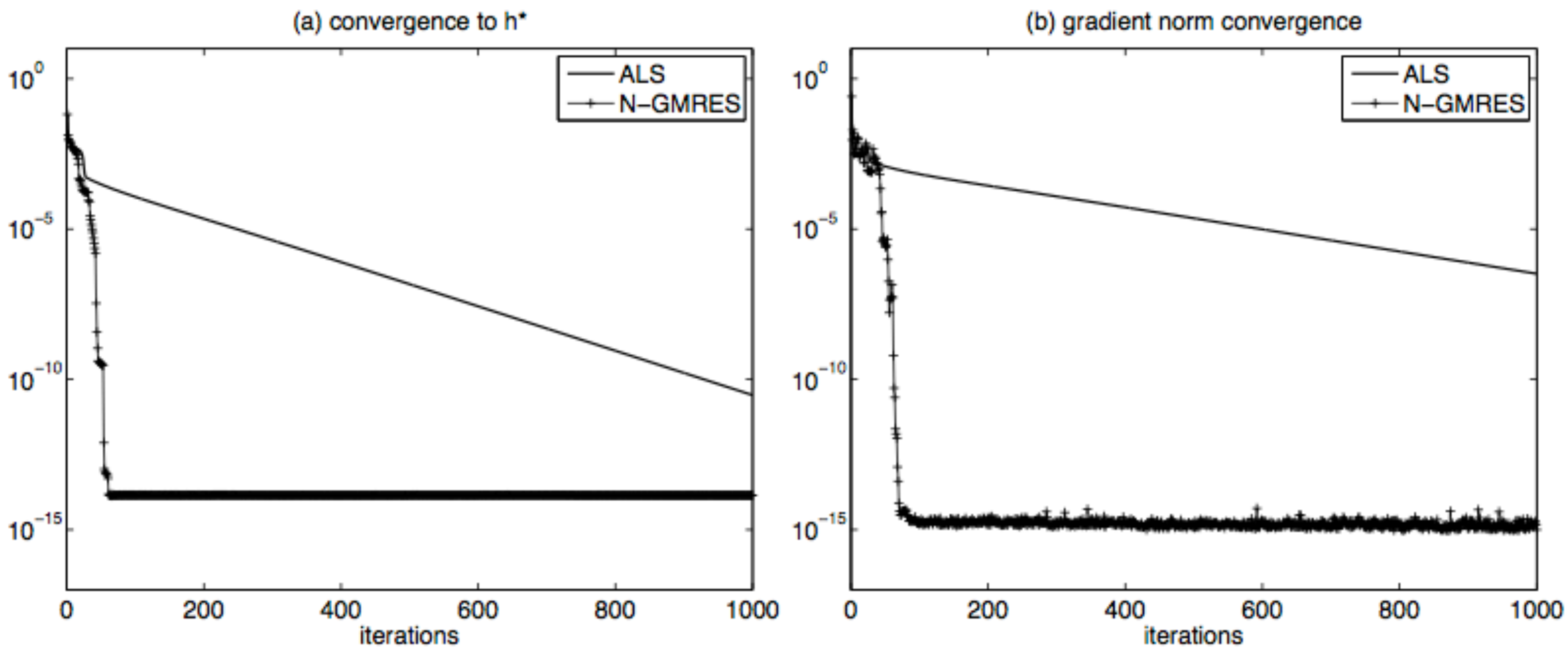
in the context of fixed-point iterations for nonlinear algebraic equations:

- can be interpreted as a specific Broyden-type multi-secant method for $\mathbf{g}(\mathbf{u}) = 0$ (and there are many 'families' of variants) (Fang and Saad, 2009)
- formal equivalence with GMRES, Arnoldi (linear case) (Walker and Ni, 2011)
- BUT: apparently not used systematically yet for optimization (or not common)
- this looks like a generally applicable continuous optimization method ...

UNIVERSITY OF WATERLOO

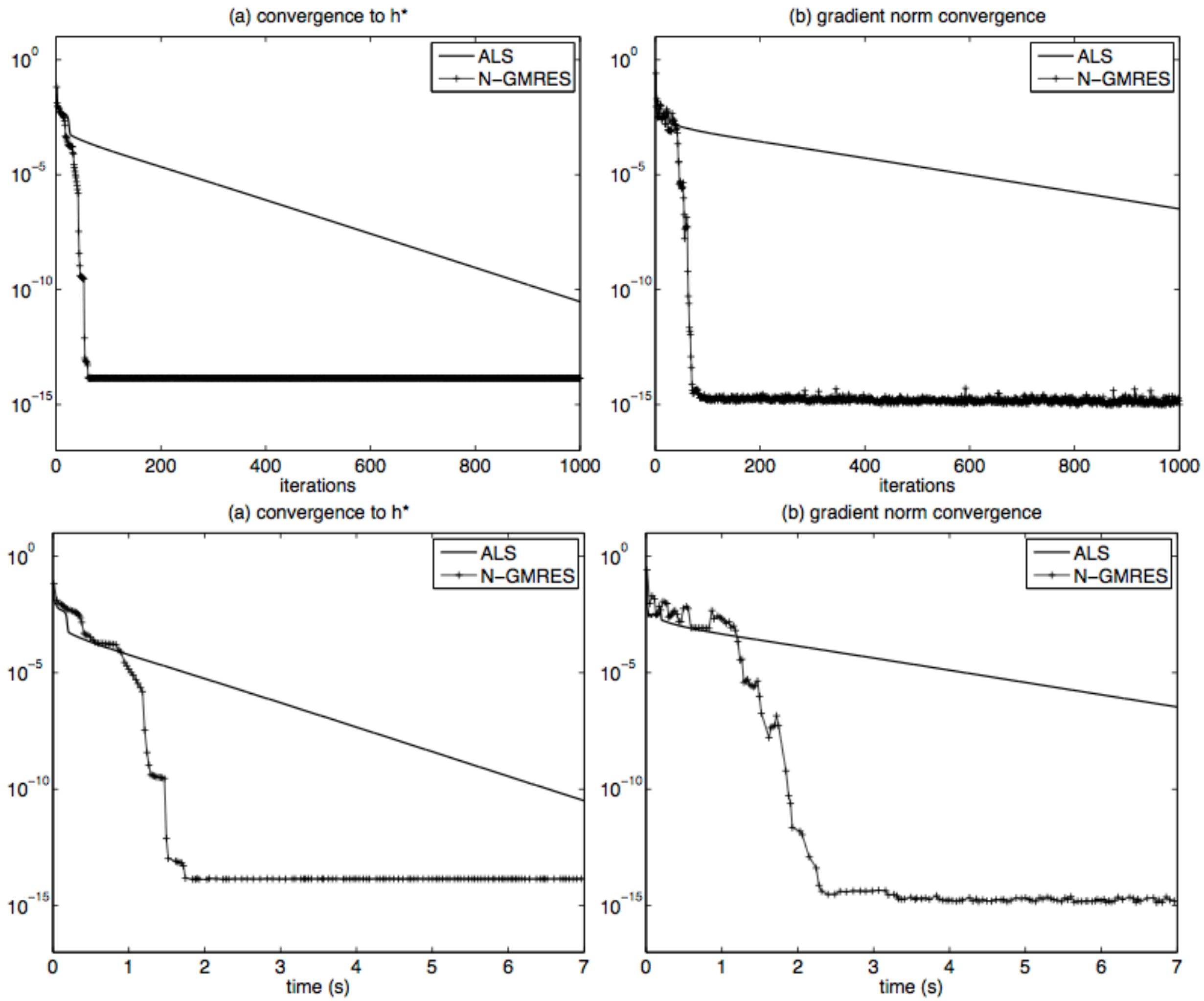
5. numerical results for ALS-preconditioned N-GMRES applied to tensor problem

- dense test problem (from Tomasi and Bro; Acar et al.): random rank-*R* tensor modified to obtain specific column collinearity, with added noise

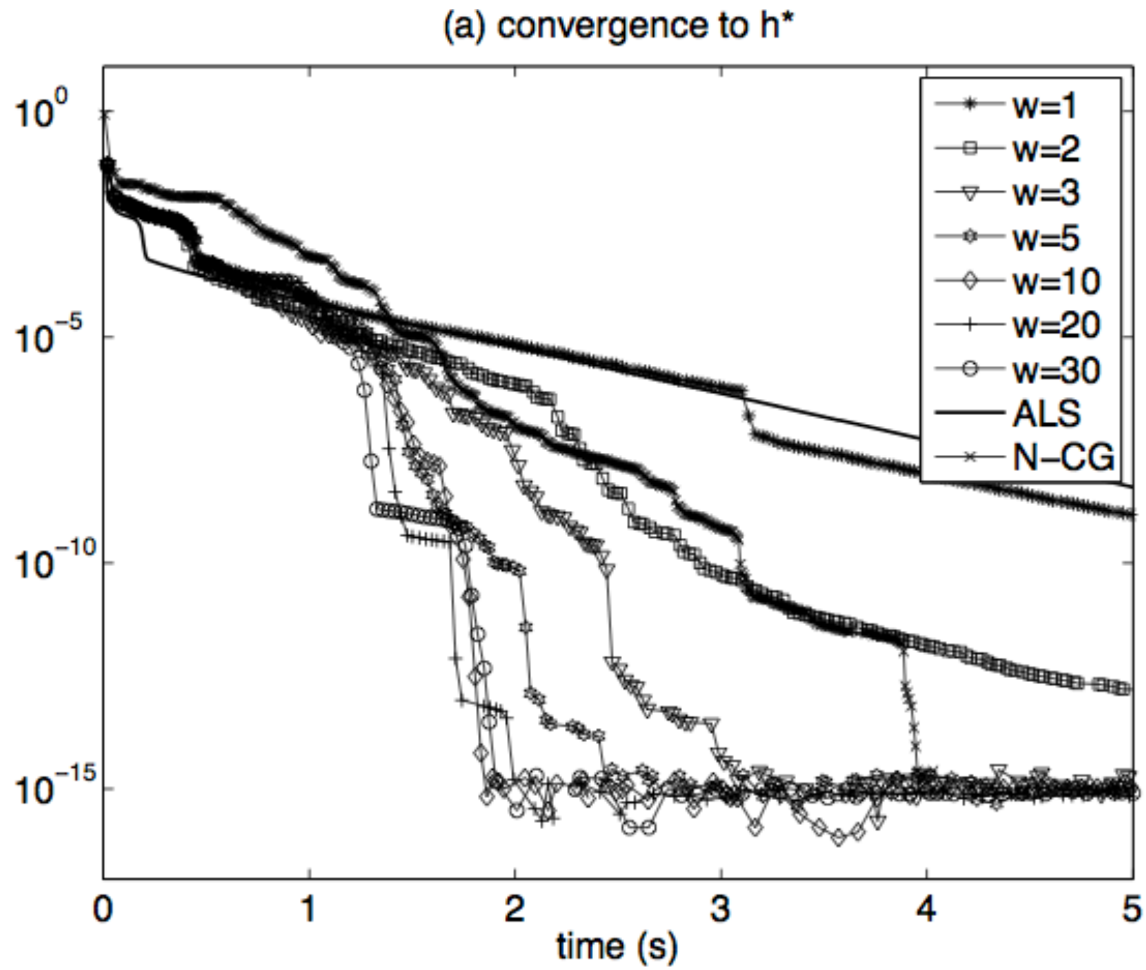


UNIVERSITY OF WATERLOO

numerical results: dense test problem



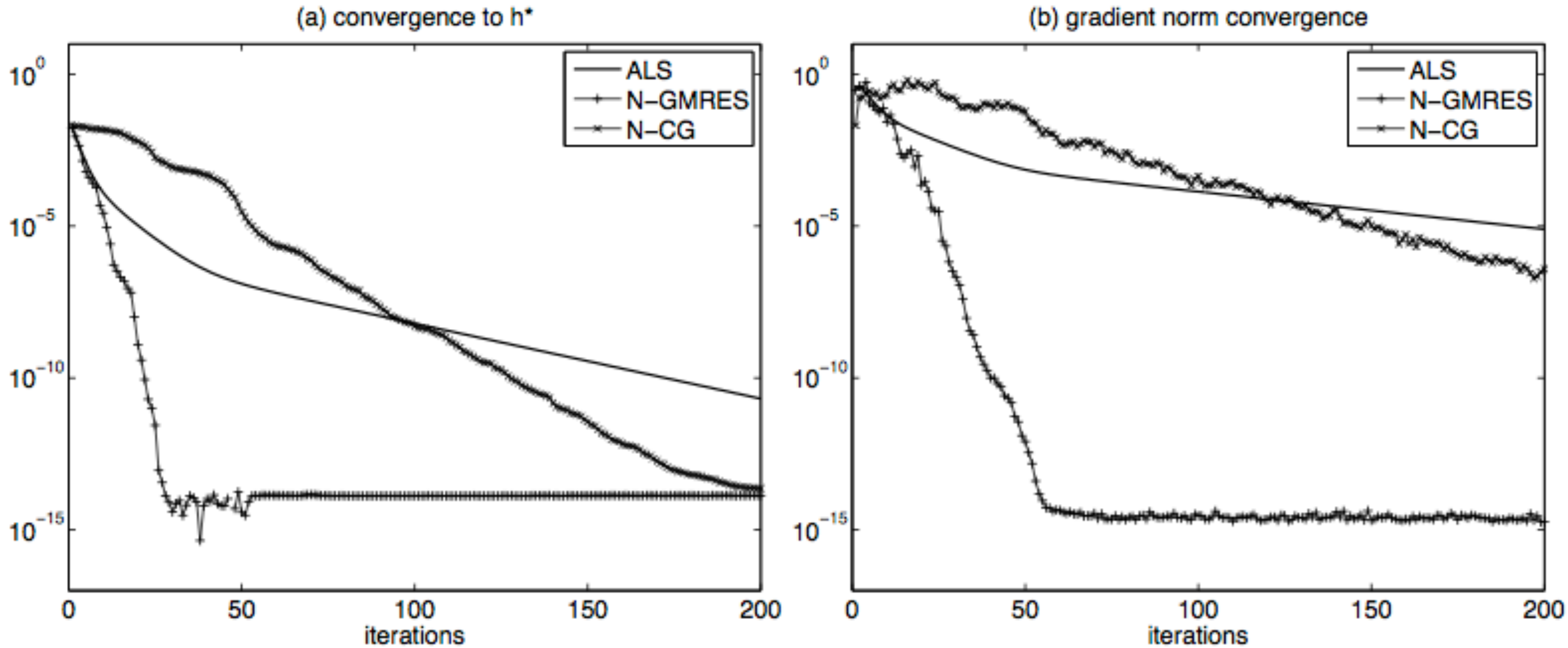
dense test problem: optimal window size



UNIVERSITY OF WATERLOO

numerical results: sparse test problem

- sparse test problem: d-dimensional finite difference Laplacian (2 d-way tensor)



UNIVERSITY OF WATERLOO

6. why does this work: linear case

GMRES for linear systems: $\mathbf{A}\,\mathbf{u} = \mathbf{b}$

- stationary iterative method $\mathbf{u}_{i+1} = \mathbf{u}_i + \mathbf{M}^{-1}\,\mathbf{r}_i$
- generates residuals recursively: $\mathbf{r}_i = \mathbf{b} - \mathbf{A}\,\mathbf{u}_i$

$$= (\mathbf{I} - \mathbf{A}\mathbf{M}^{-1})\,\mathbf{r}_{i-1}$$

$$= (\mathbf{I} - \mathbf{A}\mathbf{M}^{-1})^i\,\mathbf{r}_0.$$

- define Krylov space $K_{i+1}(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0)$

$$V_{1,i+1} = span\{\mathbf{r}_0, \ldots, \mathbf{r}_i\},$$

$$V_{2,i+1} = span\{\mathbf{r}_0, \mathbf{A}\mathbf{M}^{-1}\,\mathbf{r}_0, (\mathbf{A}\mathbf{M}^{-1})^2\,\mathbf{r}_0\}, \ldots, (\mathbf{A}\mathbf{M}^{-1})^i\,\mathbf{r}_0\}$$

$$= K_{i+1}(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0),$$

$$V_{3,i+1} = span\{\mathbf{M}\,(\mathbf{u}_1 - \mathbf{u}_0), \mathbf{M}\,(\mathbf{u}_2 - \mathbf{u}_1), \ldots, \mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_i)\},$$

$$V_{4,i+1} = span\{\mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_0), \mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_1), \ldots, \mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_i)\}$$

(Washio and Oosterlee, ETNA, 1997)

LEMMA 2.1. $V_{1,i+1} = V_{2,i+1} = V_{3,i+1} = V_{4,i+1}$

comparing N-GMRES to GMRES

(Washio and Oosterlee, ETNA, 1997)

GMRES for linear systems: $\mathbf{A}\,\mathbf{u} = \mathbf{b}$

- stationary iterative process $\mathbf{u}_{i+1} = \mathbf{u}_i + \mathbf{M}^{-1}\,\mathbf{r}_i$

  generates preconditioned residuals that build Krylov space

$$
\begin{aligned}
V_{1,i+1} &= span\{\mathbf{r}_0, \ldots, \mathbf{r}_i\}, \\
V_{2,i+1} &= span\{\mathbf{r}_0, \mathbf{AM}^{-1}\,\mathbf{r}_0, (\mathbf{AM}^{-1})^2\,\mathbf{r}_0\}, \ldots, (\mathbf{AM}^{-1})^i\,\mathbf{r}_0\} \\
&= K_{i+1}(\mathbf{AM}^{-1}, \mathbf{r}_0),
\end{aligned}
$$

- GMRES: take optimal linear combination of residuals in Krylov space to minimize the residual $\|\hat{\mathbf{r}}_{i+1}\|_2$

UNIVERSITY OF WATERLOO

comparing N-GMRES to GMRES

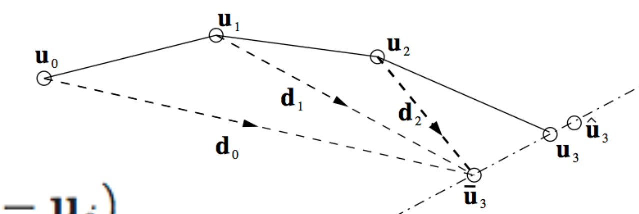
$$\mathbf{A}\,\mathbf{u} = \mathbf{b}$$

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \mathbf{M}^{-1}\,\mathbf{r}_i$$

$$\begin{aligned}
V_{1,i+1} &= span\{\mathbf{r}_0, \ldots, \mathbf{r}_i\}, \\
V_{2,i+1} &= span\{\mathbf{r}_0, \mathbf{A}\mathbf{M}^{-1}\,\mathbf{r}_0, (\mathbf{A}\mathbf{M}^{-1})^2\,\mathbf{r}_0\}, \ldots, (\mathbf{A}\mathbf{M}^{-1})^i\,\mathbf{r}_0\} \\
&= K_{i+1}(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0), \\
V_{3,i+1} &= span\{\mathbf{M}\,(\mathbf{u}_1 - \mathbf{u}_0), \mathbf{M}\,(\mathbf{u}_2 - \mathbf{u}_1), \ldots, \mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_i)\}, \\
V_{4,i+1} &= span\{\mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_0), \mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_1), \ldots, \mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_i)\}
\end{aligned}$$

- GMRES: minimize $\|\hat{\mathbf{r}}_{i+1}\|_2$
- seek optimal approximation $\mathbf{M}\,(\hat{\mathbf{u}}_{i+1} - \mathbf{u}_i) = \sum_{j=0}^{i} \beta_j\,\mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_j)$

$$\begin{aligned}
\hat{\mathbf{u}}_{i+1} &= \mathbf{u}_i + \sum_{j=0}^{i} \beta_j\,(\mathbf{u}_{i+1} - \mathbf{u}_j) \\
&= \mathbf{u}_{i+1} - (\mathbf{u}_{i+1} - \mathbf{u}_i) + \sum_{j=0}^{i} \beta_j\,(\mathbf{u}_{i+1} - \mathbf{u}_j)
\end{aligned}$$



$$\hat{\mathbf{u}}_{i+1} = \mathbf{u}_{i+1} + \sum_{j=0} \alpha_j\,(\mathbf{u}_{i+1} - \mathbf{u}_j)$$ same as for N-GMRES

UNIVERSITY OF WATERLOO

convergence speed of GMRES

$$\mathbf{A}\,\mathbf{u} = \mathbf{b}$$

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \mathbf{M}^{-1}\,\mathbf{r}_i$$

$$\begin{aligned}\mathbf{r}_i &= \mathbf{b} - \mathbf{A}\,\mathbf{u}_i \\ &= (\mathbf{I} - \mathbf{A}\mathbf{M}^{-1})\,\mathbf{r}_{i-1} \\ &= (\mathbf{I} - \mathbf{A}\mathbf{M}^{-1})^i\,\mathbf{r}_0.\end{aligned}$$

$$\begin{aligned}V_{1,i+1} &= span\{\mathbf{r}_0, \ldots, \mathbf{r}_i\}, \\ V_{2,i+1} &= span\{\mathbf{r}_0, \mathbf{A}\mathbf{M}^{-1}\,\mathbf{r}_0, (\mathbf{A}\mathbf{M}^{-1})^2\,\mathbf{r}_0\}, \ldots, (\mathbf{A}\mathbf{M}^{-1})^i\,\mathbf{r}_0\} \\ &= K_{i+1}(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0), \\ V_{3,i+1} &= span\{\mathbf{M}\,(\mathbf{u}_1 - \mathbf{u}_0), \mathbf{M}\,(\mathbf{u}_2 - \mathbf{u}_1), \ldots, \mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_i)\}, \\ V_{4,i+1} &= span\{\mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_0), \mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_1), \ldots, \mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_i)\}\end{aligned}$$

- GMRES: minimize $\|\hat{\mathbf{r}}_{i+1}\|_2$
- polynomial method: convergence determined by optimal polynomial (for diagonalizable matrix, $A=V\Lambda V^{-1}$)

$$\|r_n\| \le \inf_{p \in P_n} \|p_n(A)\| \le \kappa_2(V) \inf_{p \in P_n} \max_{\lambda \in \sigma(A)} |p(\lambda)|$$

UNIVERSITY OF WATERLOO

convergence speed of N-GMRES

STEP I: *(generate preliminary iterate by one-step update process $M(.)$)*

$$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$$

STEP II: *(generate accelerated iterate by nonlinear GMRES step)*

$$\hat{\mathbf{u}}_{i+1} = \text{gmres}(\mathbf{u}_{i-w+1}, \ldots, \mathbf{u}_i; \bar{\mathbf{u}}_{i+1})$$

STEP III: *(generate new iterate by line search process)*

$$\mathbf{u}_{i+1} = \text{linesearch}(\bar{\mathbf{u}}_{i+1} + \beta(\hat{\mathbf{u}}_{i+1} - \bar{\mathbf{u}}_{i+1}))$$

find coefficients $(\alpha_0, \ldots, \alpha_i)$ that minimize

$$\|\mathbf{g}(\bar{\mathbf{u}}_{i+1}) + \sum_{j=0}^{i} \alpha_j \left(\mathbf{g}(\bar{\mathbf{u}}_{i+1}) - \mathbf{g}(\mathbf{u}_j)\right)\|_2.$$

- GMRES (linear case): convergence determined by optimal polynomial

- convergence speed of N-GMRES for optimization: open

UNIVERSITY OF WATERLOO

7. general N-GMRES optimization method

general methods for nonlinear optimization (smooth, unconstrained)
("Numerical Optimization", Nocedal and Wright, 2006)

1. steepest descent with line search
2. Newton with line search
3. nonlinear conjugate gradient (N-CG) with line search
4. trust-region methods
5. quasi-Newton methods (includes Broyden–Fletcher–Goldfarb–Shanno (BFGS) and limited memory version L-BFGS)

6. N-GMRES as a general optimization method?

general N-GMRES optimization method

- first question: what would be a general preconditioner?

OPTIMIZATION PROBLEM

$$\text{find } \mathbf{u}^* \text{ that minimizes } f(\mathbf{u})$$

FIRST-ORDER OPTIMALITY EQUATIONS

$$\nabla f(\mathbf{u}) = \mathbf{g}(\mathbf{u}) = 0$$

- idea: general N-GMRES preconditioner $\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$

  = update in direction of steepest descent

  (or: use N-GMRES to accelerate steepest descent)

UNIVERSITY OF WATERLOO

8. steepest-descent preconditioning

STEP I: *(generate preliminary iterate by one-step update process $M(.)$)*

$$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$$

STEP II: *(generate accelerated iterate by nonlinear GMRES step)*

$$\hat{\mathbf{u}}_{i+1} = \text{gmres}(\mathbf{u}_{i-w+1}, \dots, \mathbf{u}_i; \bar{\mathbf{u}}_{i+1})$$

STEP III: *(generate new iterate by line search process)*

$$\mathbf{u}_{i+1} = \text{linesearch}(\bar{\mathbf{u}}_{i+1} + \beta(\hat{\mathbf{u}}_{i+1} - \bar{\mathbf{u}}_{i+1}))$$

STEEPEST DESCENT PRECONDITIONING PROCESS:

$$\bar{\mathbf{u}}_{i+1} = \mathbf{u}_i - \beta \frac{\nabla f(\mathbf{u}_i)}{\|\nabla f(\mathbf{u}_i)\|} \quad \text{with}$$

OPTION A: $\beta = \beta_{sdls}$,

OPTION B: $\beta = \beta_{sd} = \min(\delta, \|\nabla f(\mathbf{u}_i)\|)$

- option A: steepest descent with line search
- option B: steepest descent with predefined small step
- claim: steepest descent is the 'natural' preconditioner for N-GMRES

UNIVERSITY OF WATERLOO

steepest-descent preconditioning

- claim: steepest descent is the ‘natural’ preconditioner for N-GMRES optimization
- example: consider simple quadratic optimization problem

$$f(\mathbf{u}) = \frac{1}{2}\,\mathbf{u}^T A\mathbf{u} - \mathbf{b}^T\mathbf{u} \quad \text{where } A \text{ is SPD}$$

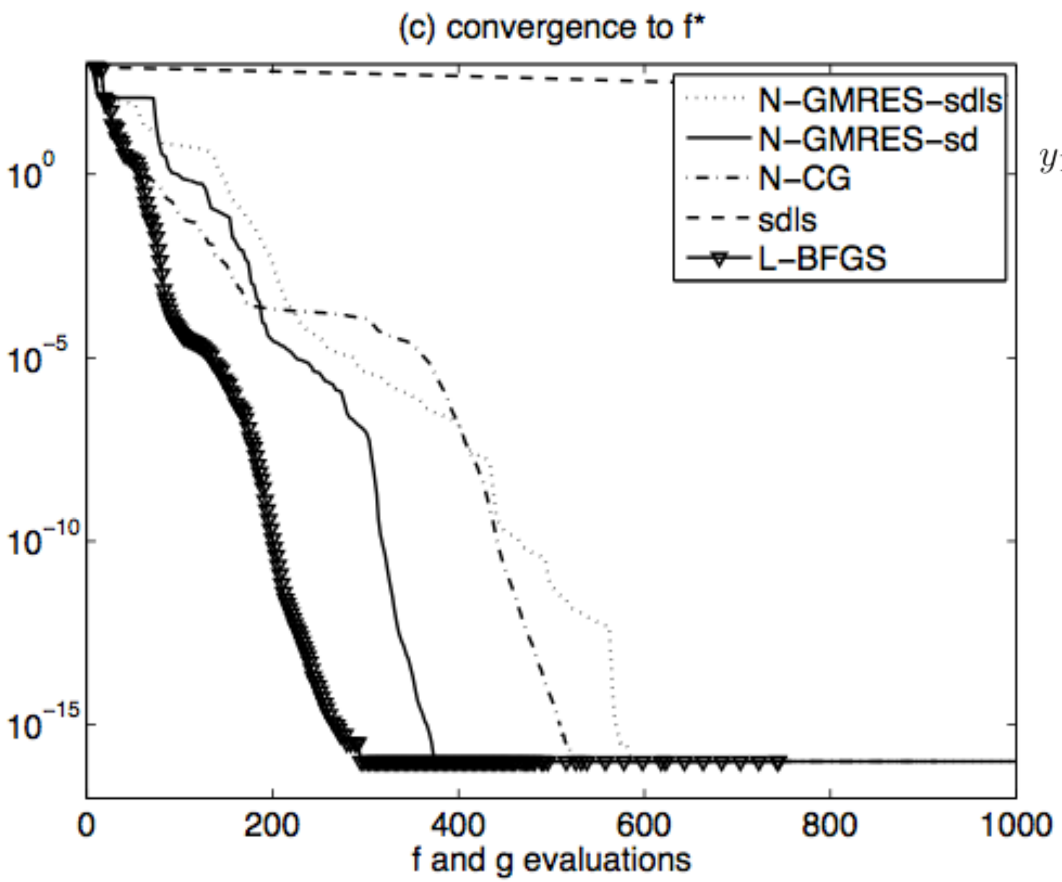
- we know $\nabla f(\mathbf{u}_i) = A\mathbf{u}_i - b = -\mathbf{r}_i$ so

$\bar{\mathbf{u}}_{i+1} = \mathbf{u}_i - \beta\,\frac{\nabla f(\mathbf{u}_i)}{\|\nabla f(\mathbf{u}_i)\|}$ becomes $\bar{\mathbf{u}}_{i+1} = \mathbf{u}_i + \beta\,\frac{\mathbf{r}_i}{\|\mathbf{r}_i\|}$

- this gives the same residuals as $\mathbf{u}_{i+1} = \mathbf{u}_i + \mathbf{M}^{-1}\mathbf{r}_i$ with $\mathbf{M} = \mathbf{I}$: steepest-descent N-GMRES preconditioner corresponds to identity preconditioner for linear GMRES

(and: small step is sufficient)

9. numerical results: steepest-descent preconditioning



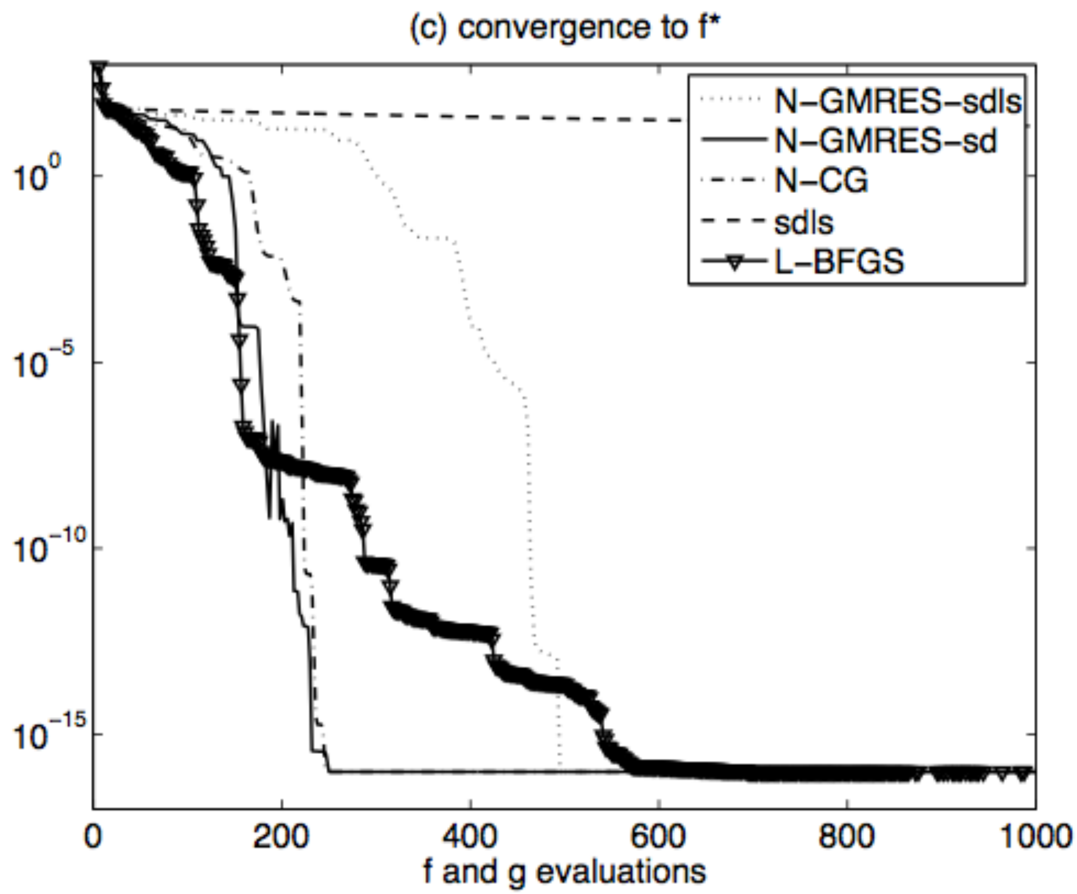
$$f(\mathbf{u}) = \frac{1}{2}\,\mathbf{y}(\mathbf{u}-\mathbf{u}^*)^T D\,\mathbf{y}(\mathbf{u}-\mathbf{u}^*) + 1,$$

with $D = \mathrm{diag}(1, 2, \ldots, n)$ and $\mathbf{y}(\mathbf{x})$ given by $y_1(\mathbf{x}) = x_1$ and $y_i(\mathbf{x}) = x_i - 10\,x_1^2$ $(i = 2, \ldots, n)$.

- steepest descent by itself is slow
- N-GMRES with steepest descent preconditioning is competitive with N-CG and L-BFGS
- option A slower than option B (small step)

UNIVERSITY OF WATERLOO

numerical results: steepest-descent preconditioning



$$f(\mathbf{u}) = \frac{1}{2}\sum_{j=1}^{n} t_j^2(\mathbf{u}), \text{ with } n \text{ even and}$$

$$t_j = 10\,(u_{j+1} - u_j^2) \qquad (j \text{ odd}),$$

$$t_j = 1 - u_{j-1} \qquad (j \text{ even}).$$

- extended Rosenbrock function
- steepest descent by itself is slow
- N-GMRES with steepest descent preconditioning is competitive with N-CG and L-BFGS

UNIVERSITY OF WATERLOO

10. convergence of steepest-descent preconditioned N-GMRES optimization

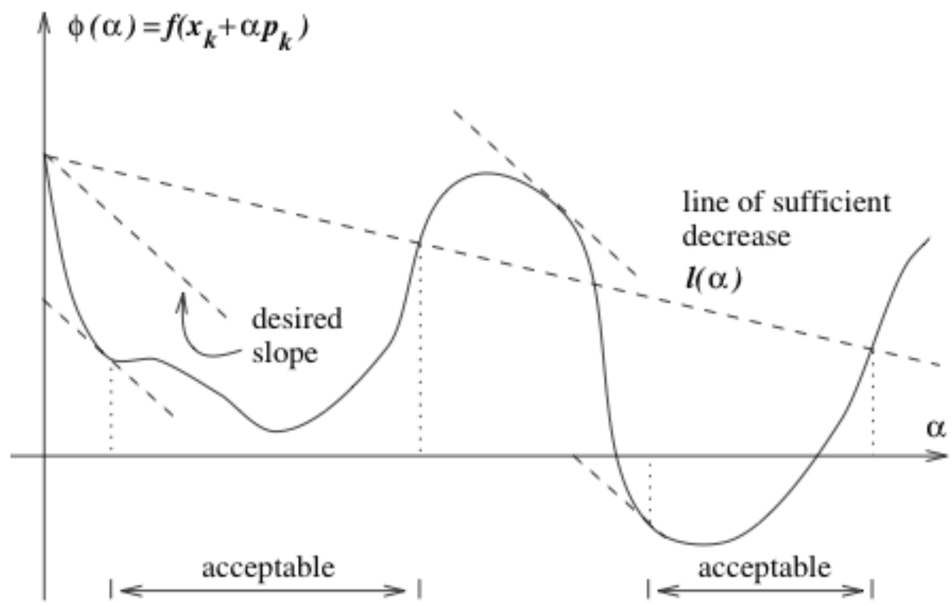
- assume line searches give solutions that satisfy Wolfe conditions:

SUFFICIENT DECREASE CONDITION:

$$f(\mathbf{u}_i + \beta_i \mathbf{p}_i) \leq f(\mathbf{u}_i) + c_1\, \beta_i\, \nabla f(\mathbf{u}_i)^T \mathbf{p}_i,$$

CURVATURE CONDITION:

$$\nabla f(\mathbf{u}_i + \beta_i \mathbf{p}_i)^T\, \mathbf{p}_i \geq c_2\, \nabla f(\mathbf{u}_i)^T \mathbf{p}_i,$$



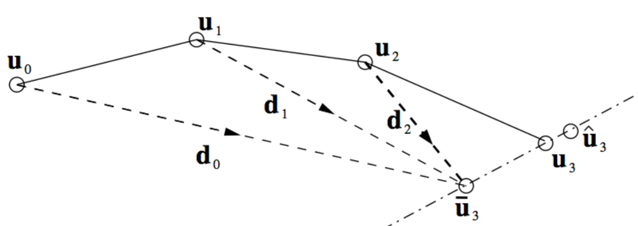
(Nocedal and Wright, 2006)

UNIVERSITY OF WATERLOO

convergence of steepest-descent preconditioned N-GMRES optimization

Theorem 2.1 (Global convergence of N-GMRES optimization algorithm with steepest descent line search preconditioning). *Consider N-GMRES Optimization Algorithm 1 with steepest descent line search preconditioning (2.1) for Optimization Problem I, and assume that all line search solutions satisfy the Wolfe conditions, (2.11) and (2.12). Assume that objective function f is bounded below in $\mathbb{R}^n$ and that f is continuously differentiable in an open set $\mathcal{N}$ containing the level set $\mathcal{L} = \{\mathbf{u} : f(\mathbf{u}) \leq f(\mathbf{u}_0)\}$, where $\mathbf{u}_0$ is the starting point of the iteration. Assume also that the gradient ∇f is Lipschitz continuous on $\mathcal{N}$, that is, there exists a constant L such that $\|\nabla f(\mathbf{u}) - \nabla f(\hat{\mathbf{u}})\| \leq L\|\mathbf{u} - \hat{\mathbf{u}}\|$ for all $\mathbf{u}, \hat{\mathbf{u}} \in \mathcal{N}$. Then the sequence of N-GMRES iterates $\{\mathbf{u}_0, \mathbf{u}_1, \ldots\}$ is convergent to a fixed point of Optimization Problem I in the sense that*

$$\lim_{i \to \infty} \|\nabla f(\mathbf{u}_i)\| = 0. \tag{2.13}$$



UNIVERSITY OF WATERLOO

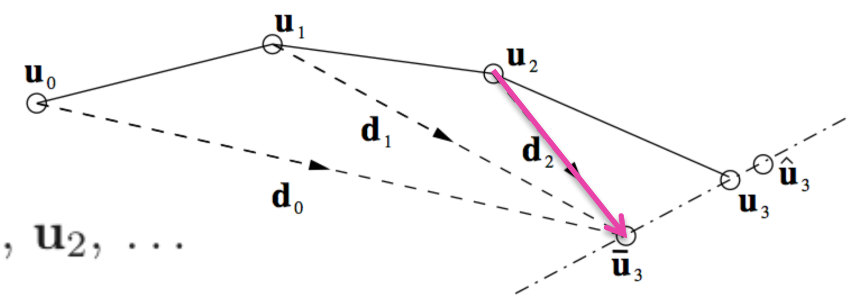
Step I: *(generate preliminary iterate by one-step update process $M(.)$)*
$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$

Step II: *(generate accelerated iterate by nonlinear GMRES step)*
$\hat{\mathbf{u}}_{i+1} = \text{gmres}(\mathbf{u}_{i-w+1}, \ldots, \mathbf{u}_i; \bar{\mathbf{u}}_{i+1})$

Step III: *(generate new iterate by line search process)*
$\mathbf{u}_{i+1} = \text{linesearch}(\bar{\mathbf{u}}_{i+1} + \beta(\hat{\mathbf{u}}_{i+1} - \bar{\mathbf{u}}_{i+1}))$

convergence of steepest-descent preconditioned N-GMRES optimization

sketch of (simple!) proof



- Consider the sequence $\{\mathbf{v}_0, \mathbf{v}_1, \ldots\}$ formed by the iterates $\mathbf{u}_0$, $\bar{\mathbf{u}}_1$, $\mathbf{u}_1$, $\bar{\mathbf{u}}_2$, $\mathbf{u}_2$, ...
- use Zoutendijk's theorem: $\sum_{i=0}^{\infty} \cos^2\theta_i \, \|\nabla f(\mathbf{v}_i)\|^2 < \infty$
  with $\cos\theta_i = \dfrac{-\nabla f(\mathbf{v}_i)^T \mathbf{p}_i}{\|\nabla f(\mathbf{v}_i)\| \, \|\mathbf{p}_i\|}$ and thus $\lim_{i\to\infty} \cos^2\theta_i \, \|\nabla f(\mathbf{v}_i)\|^2 = 0$
- all u_i are followed by a steepest descent step, so

$$\lim_{i\to\infty} \|\nabla f(\mathbf{u}_i)\| = 0.$$

- global convergence to a stationary point for general *f(u)*

UNIVERSITY OF WATERLOO

general N-GMRES optimization method

general methods for nonlinear optimization (smooth, unconstrained)
("Numerical Optimization", Nocedal and Wright, 2006)

1. steepest descent with line search
2. Newton with line search
3. nonlinear conjugate gradient (N-CG) with line search
4. trust-region methods
5. quasi-Newton methods (includes Broyden–Fletcher–Goldfarb–Shanno (BFGS) and limited memory version L-BFGS)

6. N-GMRES as a general optimization method

11. conclusions

- we have proposed the N-GMRES optimization method: a (new?, uncommon) general, convergent method (with steepest-descent preconditioning), appears competitive with N-CG, L-BFGS
- 'preconditioned GMRES' ideas can be extended to nonlinear optimization! (can use powerful nonlinear preconditioners) (ALS in tensor case)

STEP I: *(generate preliminary iterate by one-step update process $M(.)$)*

$$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$$

STEP II: *(generate accelerated iterate by nonlinear GMRES step)*

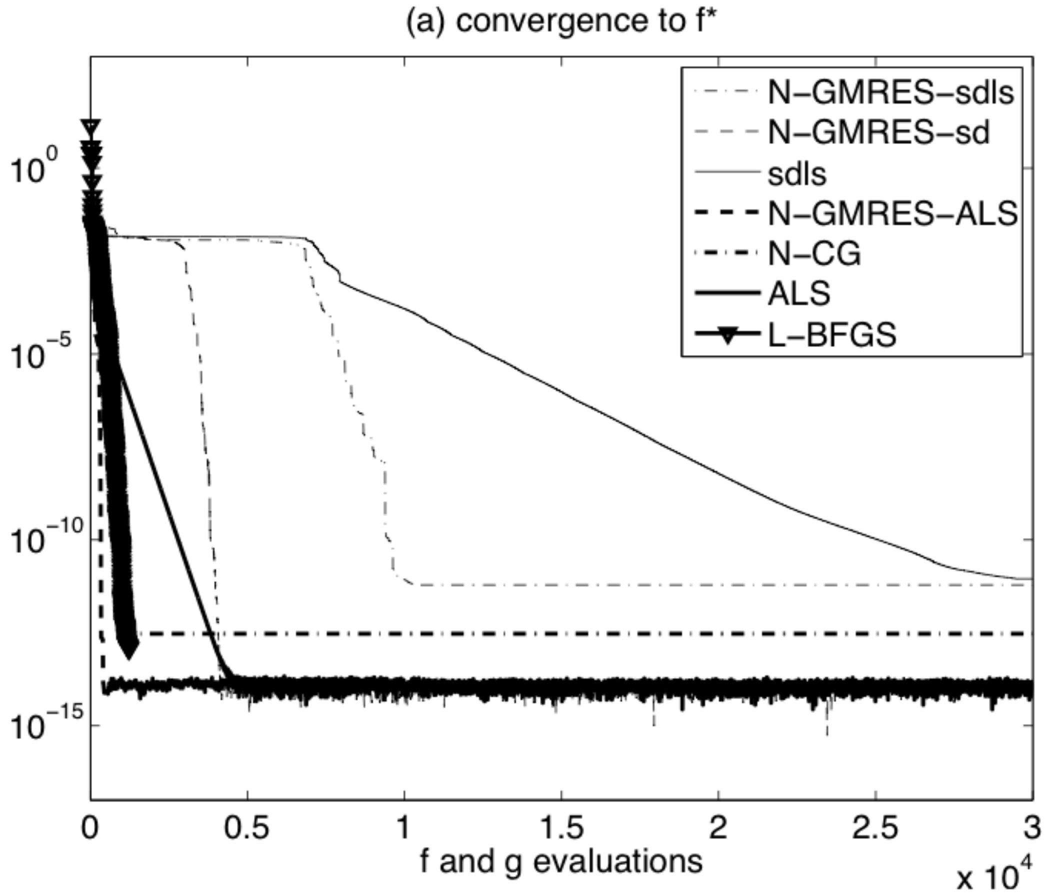
$$\hat{\mathbf{u}}_{i+1} = \text{gmres}(\mathbf{u}_{i-w+1}, \ldots, \mathbf{u}_i; \bar{\mathbf{u}}_{i+1})$$

STEP III: *(generate new iterate by line search process)*

$$\mathbf{u}_{i+1} = \text{linesearch}(\bar{\mathbf{u}}_{i+1} + \beta(\hat{\mathbf{u}}_{i+1} - \bar{\mathbf{u}}_{i+1}))$$

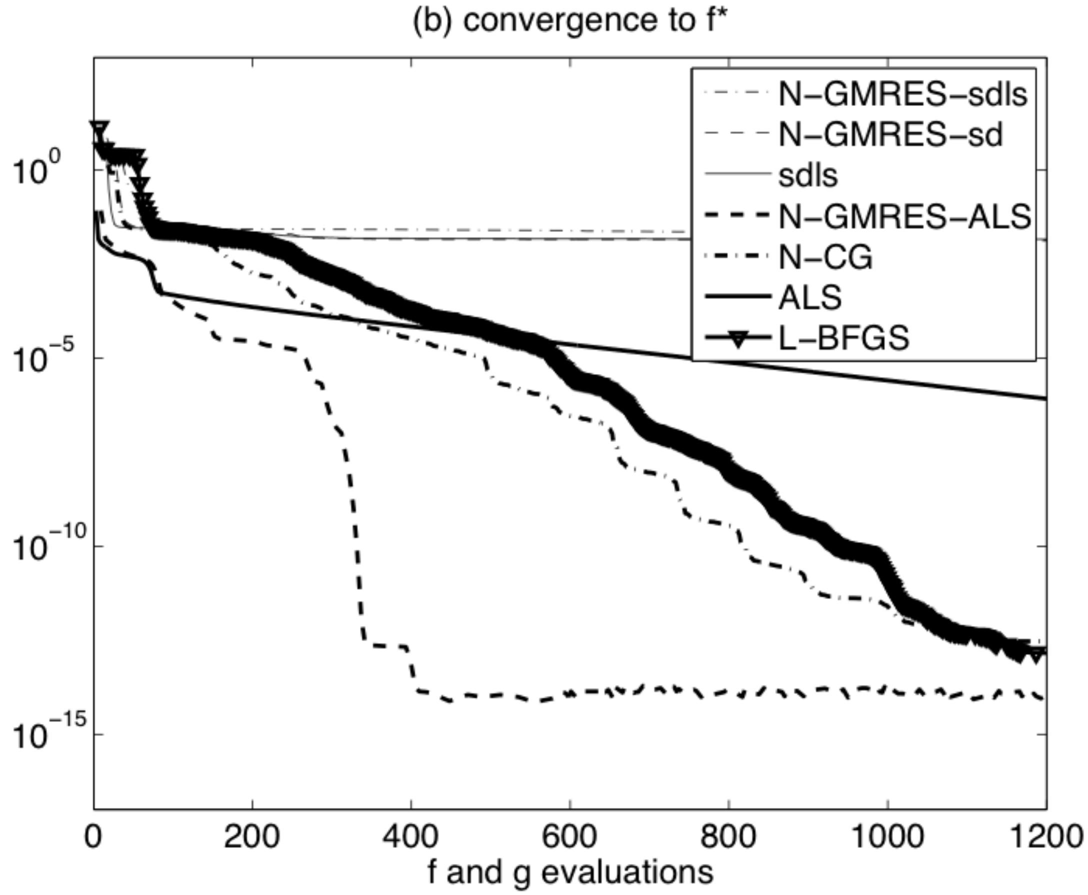
UNIVERSITY OF WATERLOO

the power of N-GMRES optimization (tensor problem)



UNIVERSITY OF WATERLOO

the power of N-GMRES optimization



nonlinear 'preconditioned GMRES' for nonlinear optimization! (using powerful nonlinear preconditioners)

UNIVERSITY OF WATERLOO

- thank you
- questions?

- Hans De Sterck, *'A Nonlinear GMRES Optimization Algorithm for Canonical Tensor Decomposition'*, submitted to SIAM J. Sci. Comp., May 2011, *arXiv:1105.5331*
- Hans De Sterck, *'Steepest Descent Preconditioning for Nonlinear GMRES Optimization'*, NLA, in press, *arXiv:1106.4426*

UNIVERSITY OF WATERLOO