Extending Preconditioned GMRES to Nonlinear Optimization

UNIVERSITY OF WATERLOO

uwaterloo.ca

Hans De Sterck

Department of Applied Mathematics
University of Waterloo, Canada

CAIMS Annual Meeting
Quebec, June 2013

nonlinear optimization

- consider smooth nonlinear optimization problem

Optimization Problem

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

First-order optimality equations

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

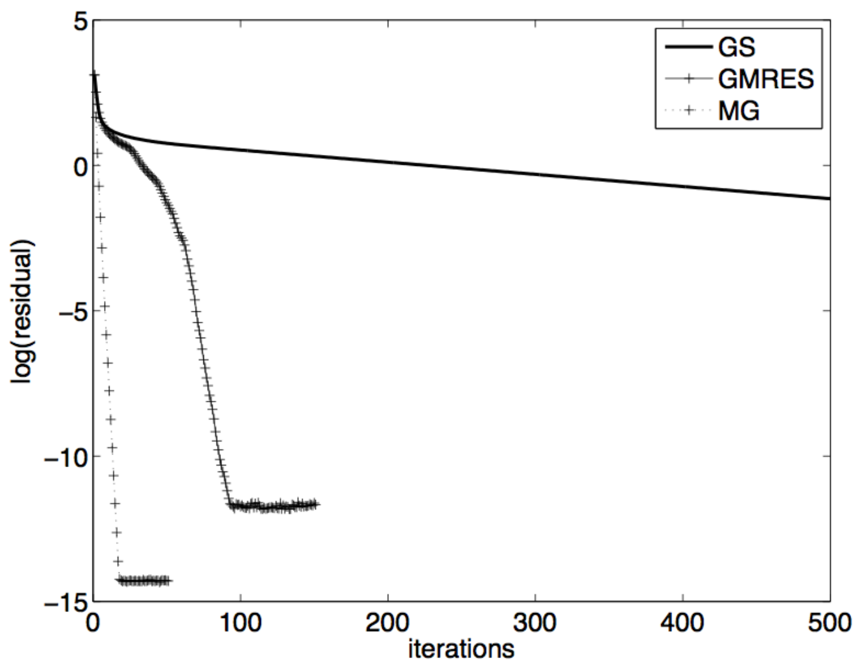
- can we accelerate the convergence of simple iterative optimization methods? ("Alternating Least Squares" (ALS), coordinate descent)

convergence acceleration for linear systems (e.g., FD for elliptic PDE on square domain)

- GS converges slowly
- number of iterations grows as grid is refined
- can we accelerate GS?

  yes!
  - GMRES acceleration (generalized minimal residual method)
  - multigrid, CG

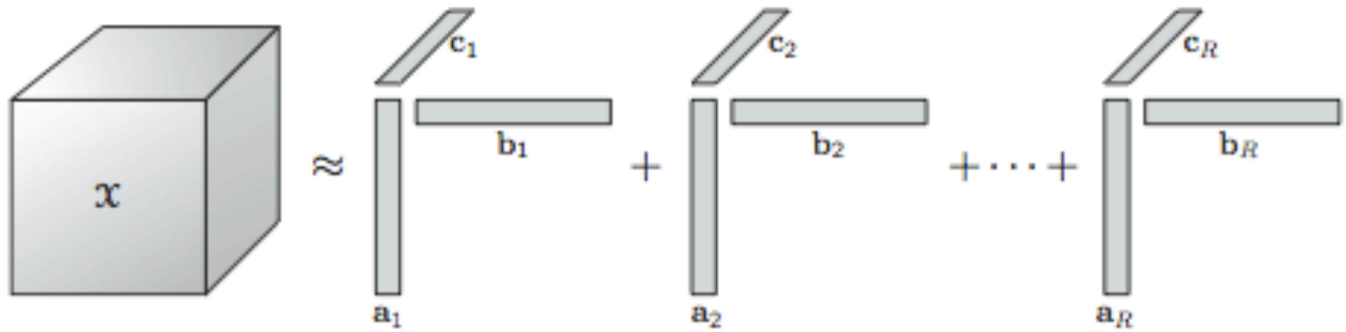
$$\mathbf{A}\,\mathbf{u} = \mathbf{b}$$



UNIVERSITY OF WATERLOO

(1) canonical tensor decomposition

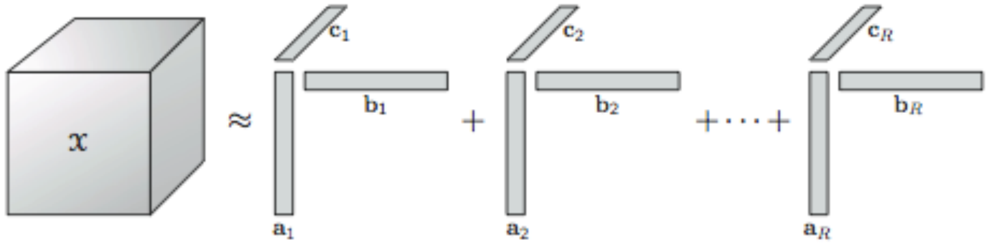
- tensor = element of tensor product of real vector spaces (*N*-dimensional array)
- *N*=3:



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

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

canonical tensor decomposition



OPTIMIZATION PROBLEM

given tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, find rank-$R$ canonical tensor $\mathcal{A}_R \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 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 (ill-posed), ... ; but smooth, and we assume there is a local minimum)

(de Silva and Lim, SIMAX, 2009)

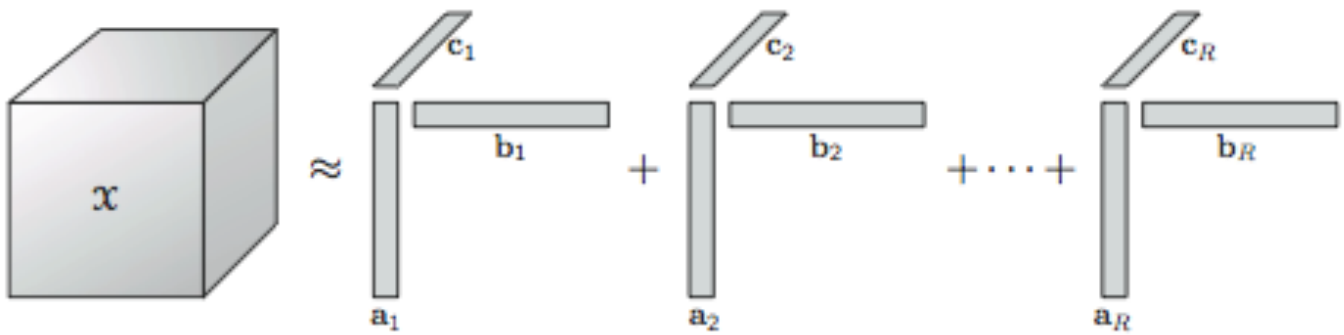
UNIVERSITY OF WATERLOO

link with singular value decomposition

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

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

- canonical decomposition of tensor



a difference with the SVD

truncated SVD is best rank-*R* approximation:

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

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

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

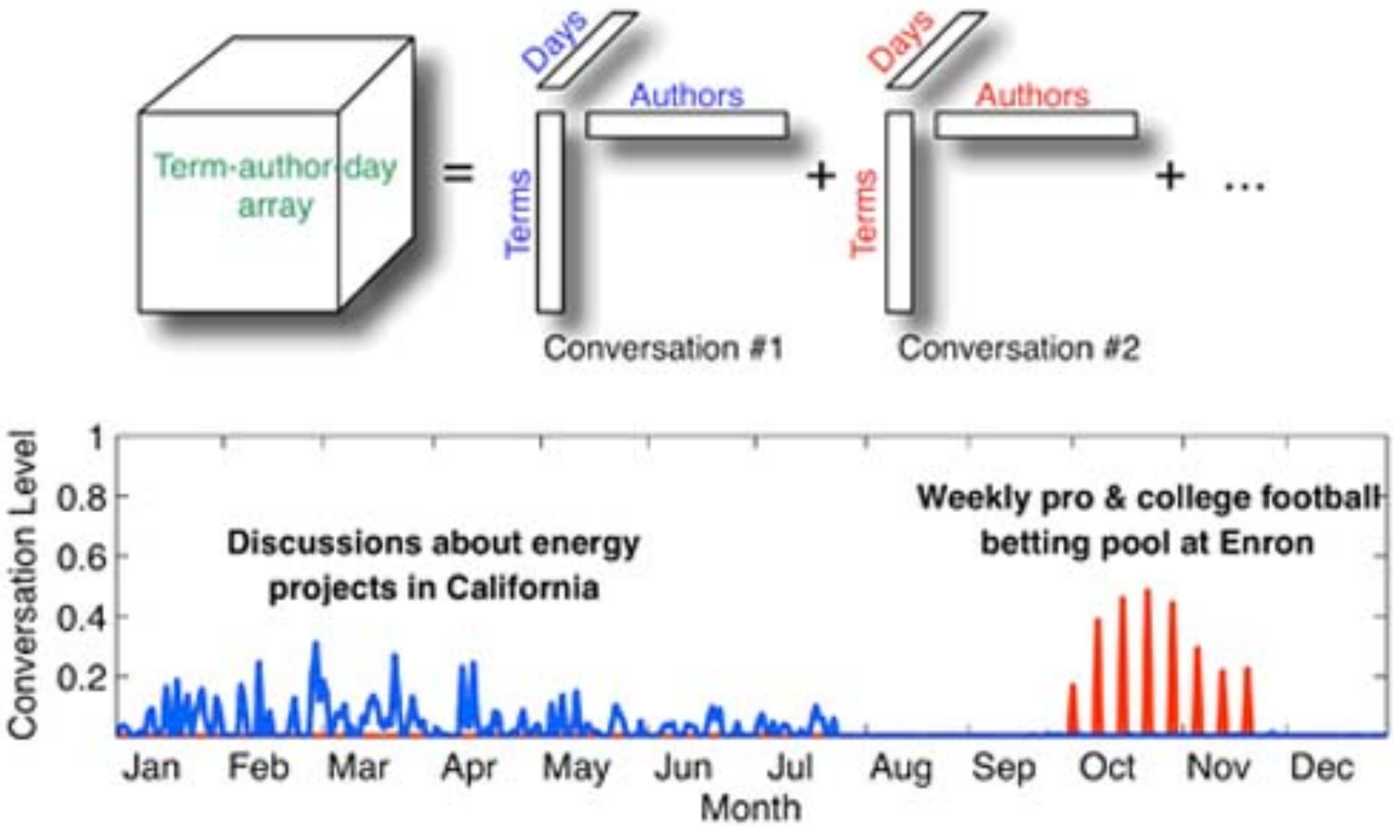
given tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, find rank-$R$ canonical tensor $\mathcal{A}_R \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ that minimizes

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

UNIVERSITY OF WATERLOO

tensor approximation applications

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

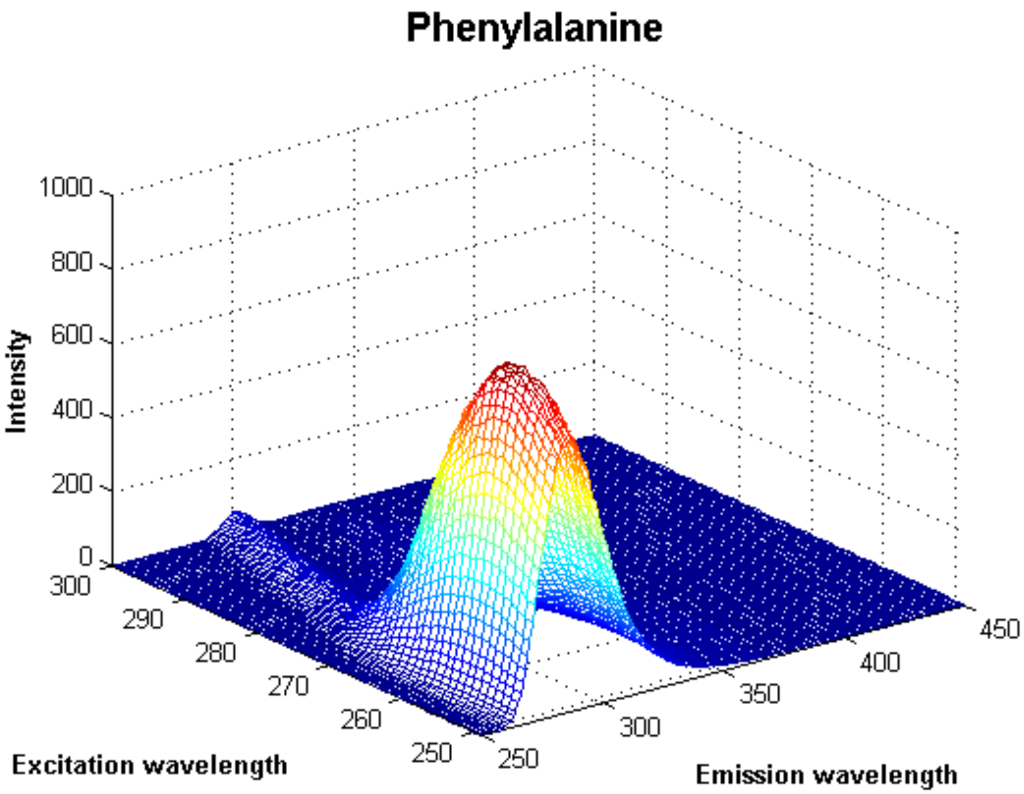


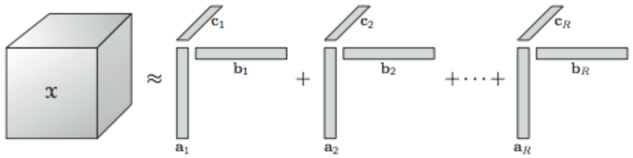
UNIVERSITY OF WATERLOO

tensor approximation applications

(2) 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)

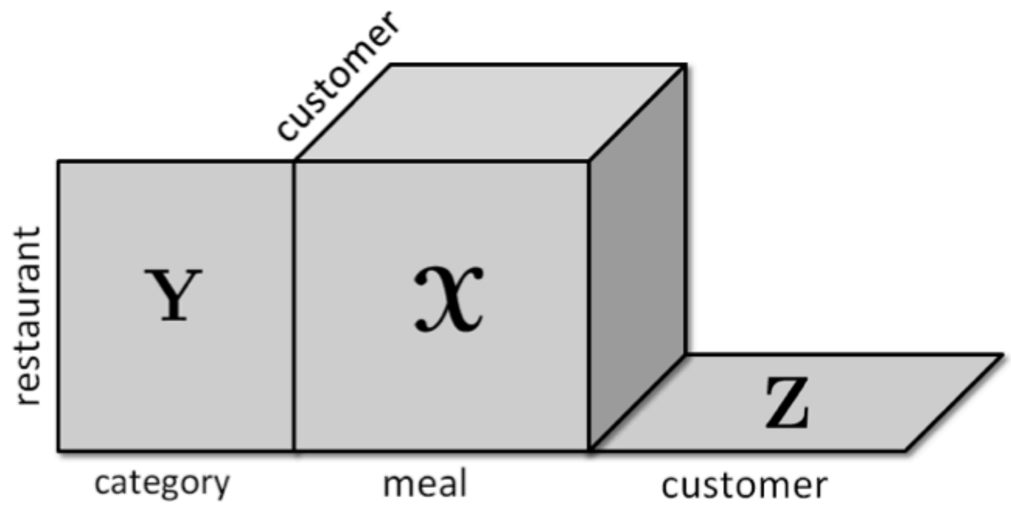


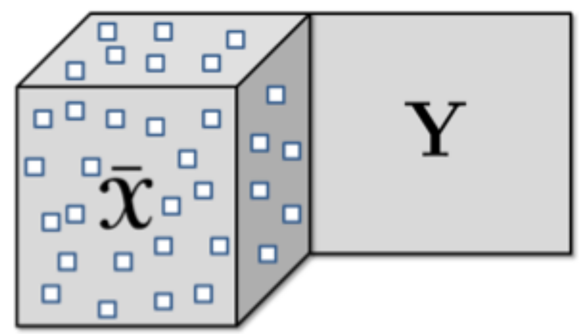


(from [1])

tensor approximation applications

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





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

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

UNIVERSITY OF WATERLOO

(2) 'workhorse' algorithm: alternating least squares (ALS)

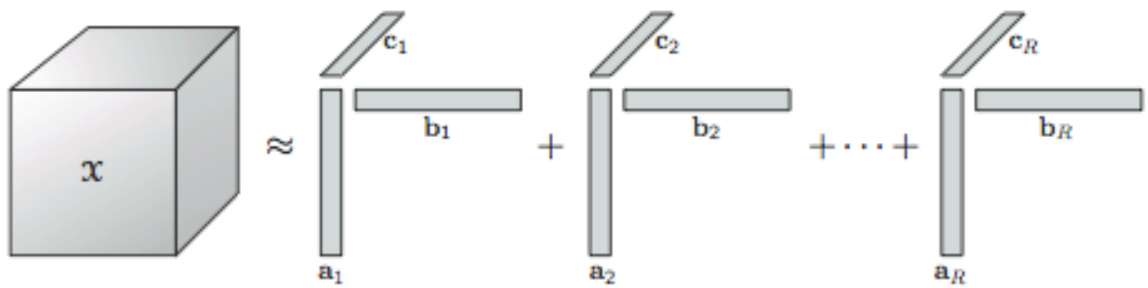
$$f(\mathcal{A}_R) = \frac{1}{2}\left\|\mathcal{T} - \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ a_r^{(3)}\right\|_F^2$$

(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)

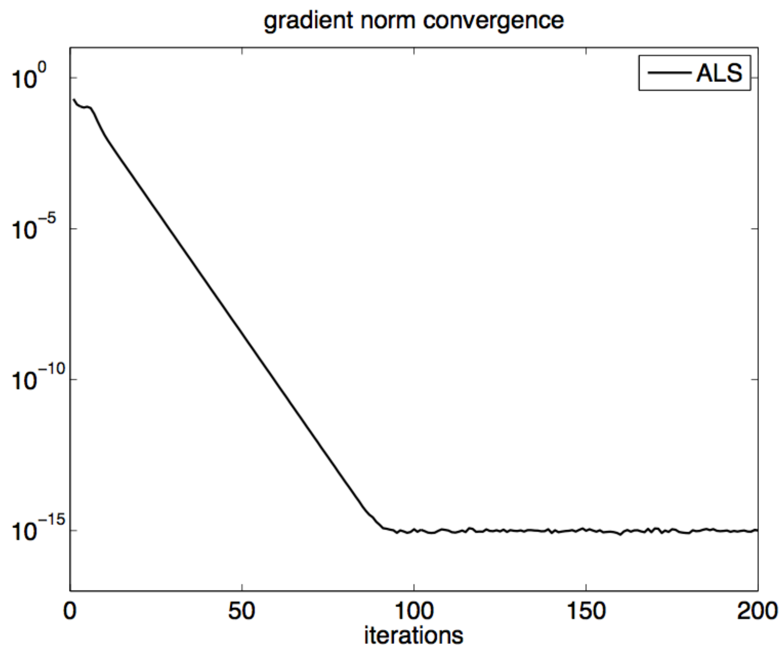
$$f(\mathcal{A}_R) = \frac{1}{2}\left\|\mathcal{T} - \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ a_r^{(3)}\right\|_F^2$$

- "simple iterative optimization method"
- ALS is block nonlinear Gauss-Seidel (other name: coordinate descent)
- ALS is monotone
- ALS is sometimes fast, but can also be extremely slow (depending on problem and initial condition)

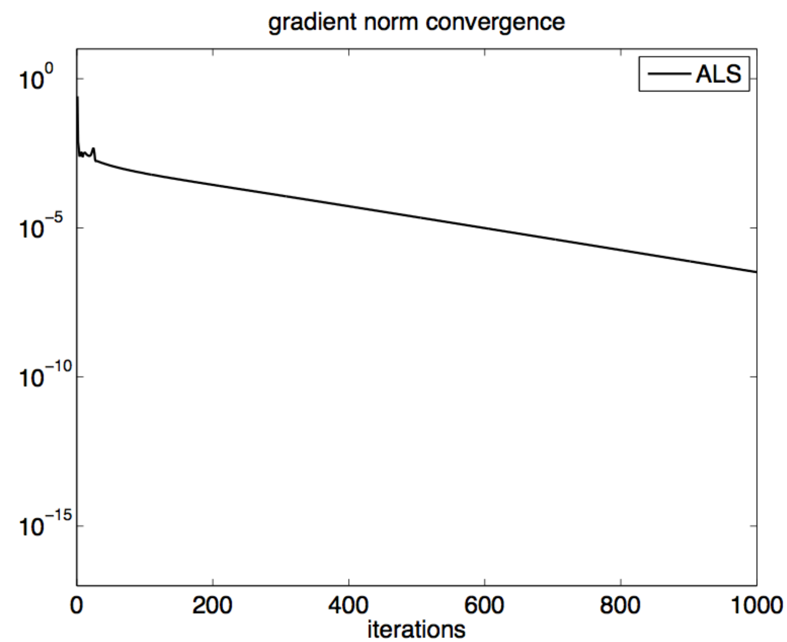
alternating least squares (ALS)

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

fast case



slow case



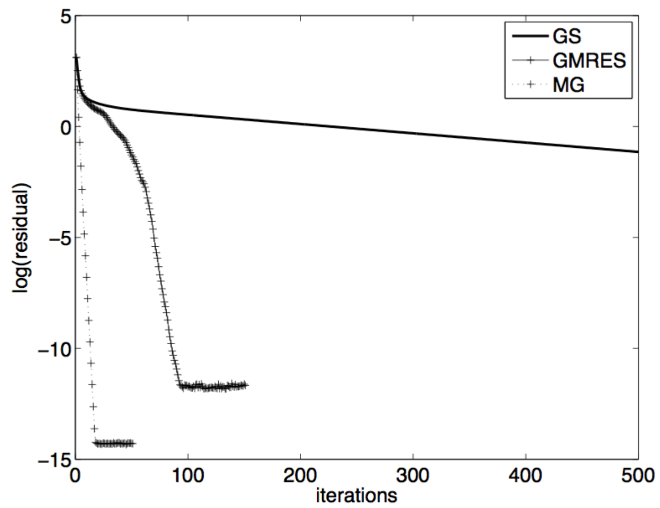
(we used Matlab with Tensor Toolbox (Bader and Kolda) and Poblano Toolbox (Dunlavy et al.) for all computations)

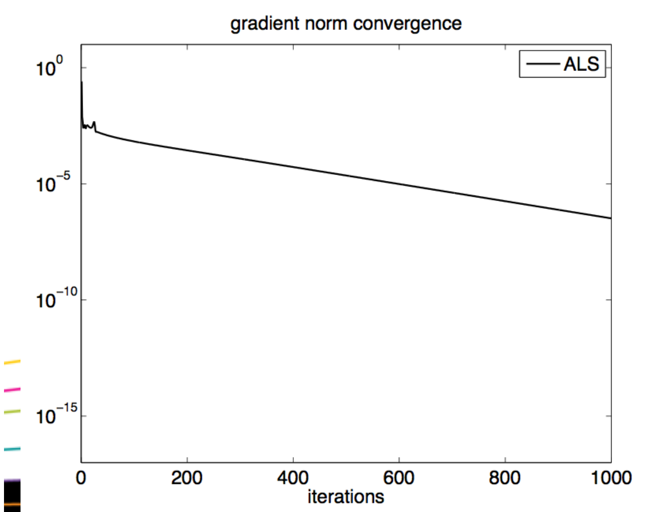
UNIVERSITY OF WATERLOO

convergence acceleration for ALS (nonlinear optimization)

convergence acceleration for linear systems	convergence acceleration for nonlinear optimization		
P-CG	NCG, P? (nonl.)		
P-GMRES	?		
MG	? (MG/OPT)		

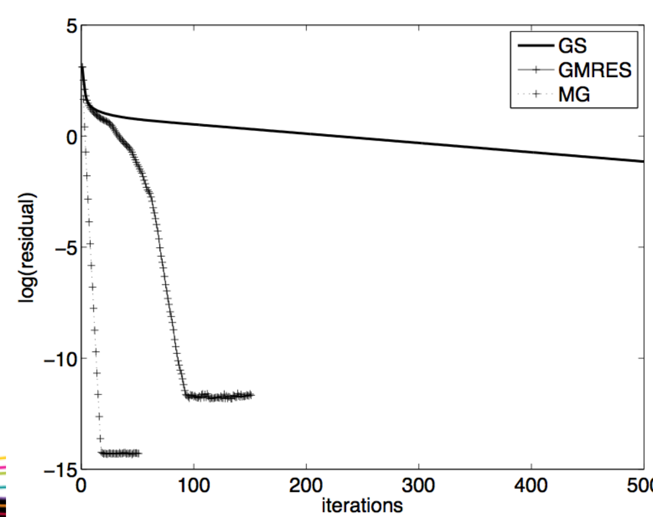
UNIVERSITY OF WATERLOO

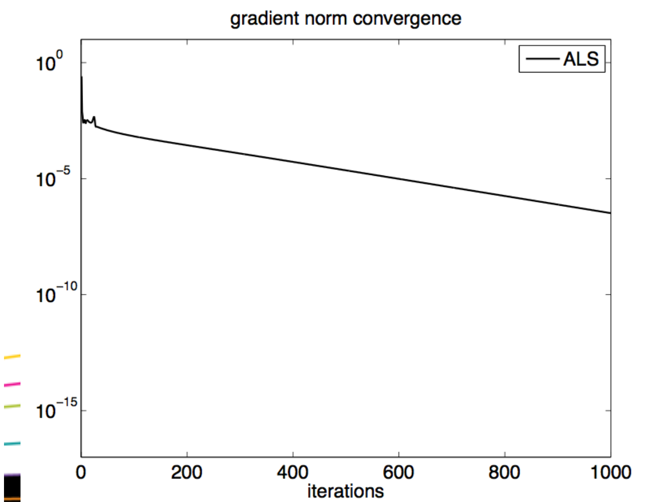




convergence acceleration for ALS (nonlinear optimization)

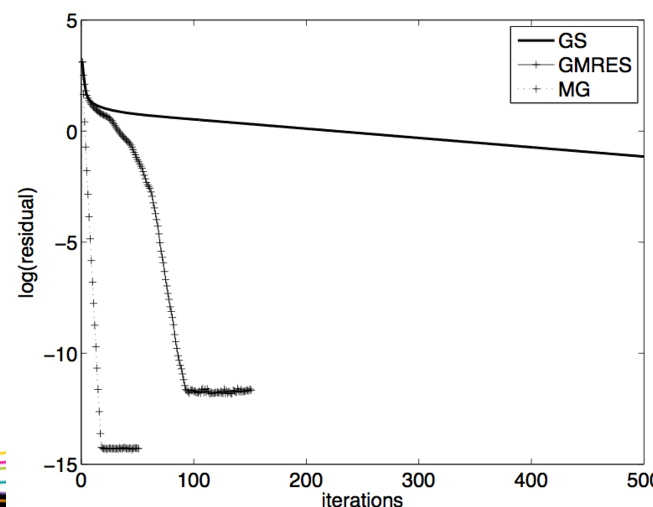
convergence acceleration for linear systems	convergence acceleration for nonlinear optimization	convergence acceleration for nonlinear systems	
P-CG	NCG, P? (nonl.)	NCG, P? (nonl.)	
P-GMRES	?	P-NGMRES (Washio and Oosterlee, Anderson)	
MG	? (MG/OPT)	FAS	

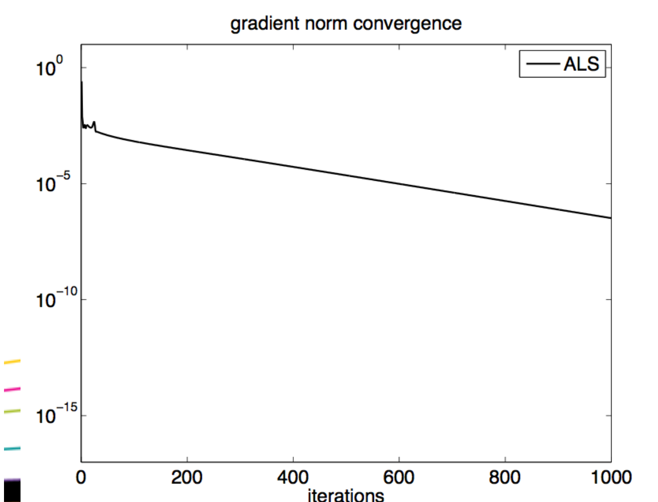




convergence acceleration for ALS (nonlinear optimization)

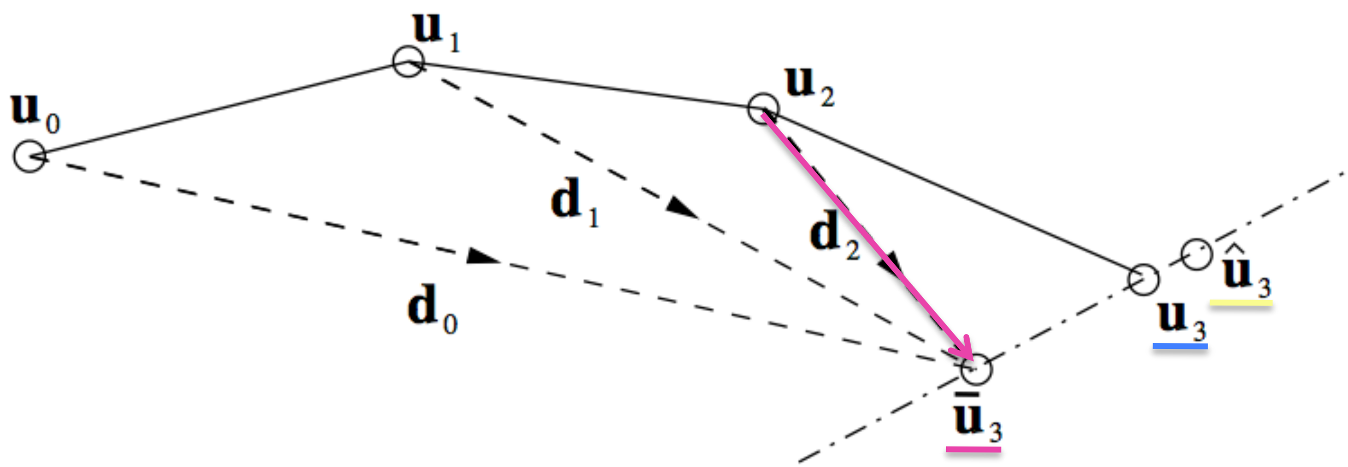
convergence acceleration for linear systems	convergence acceleration for nonlinear optimization	convergence acceleration for nonlinear systems	convergence acceleration for nonlinear optimization
P-CG	NCG, P? (nonl.)	NCG, P? (nonl.)	P-NCG
P-GMRES	?	P-NGMRES (Washio and Oosterlee, Anderson)	P-NGMRES for optimization
MG	?	FAS	adaptive AMG-FAS for optimization





UNIVERSITY OF WATERLOO

(3) nonlinear GMRES optimization method (NGMRES)



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 $M(.)$)* (ALS)

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

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)* (Moré-Thuente line search, satisfies Wolfe conditions)

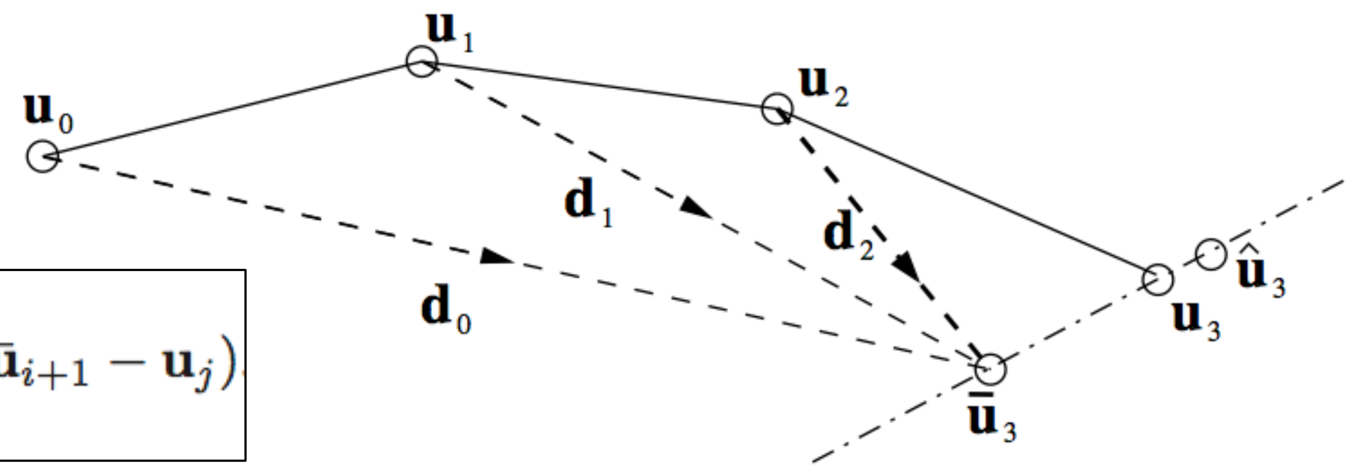
$\mathbf{u}_{i+1} = \text{linesearch}(\bar{\mathbf{u}}_{i+1} + \beta(\hat{\mathbf{u}}_{i+1} - \bar{\mathbf{u}}_{i+1}))$

$i = i + 1$

until *convergence criterion satisfied*

UNIVERSITY OF WATERLOO

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



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

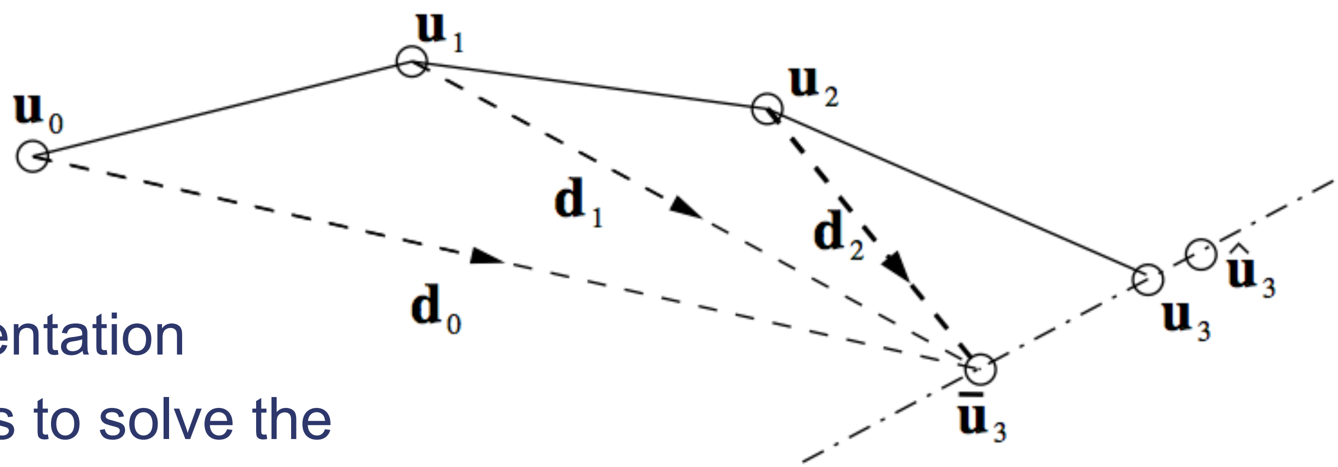
$$\mathbf{g}(\hat{\mathbf{u}}_{i+1}) \approx \mathbf{g}(\bar{\mathbf{u}}_{i+1}) + \sum_{j=0}^{i} \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\bar{\mathbf{u}}_{i+1}} \alpha_j \left(\bar{\mathbf{u}}_{i+1} - \mathbf{u}_j\right)$$

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

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

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

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



- windowed implementation
- several possibilities to solve the small least-squares problems:
  - normal equations (with reuse of scalar products; cost *2nw*) (Washio and Oosterlee, 1997)
  - QR with factor updating (Walker and Ni, 2011)
  - SVD and rank-revealing QR (Fang and Saad, 2009)

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

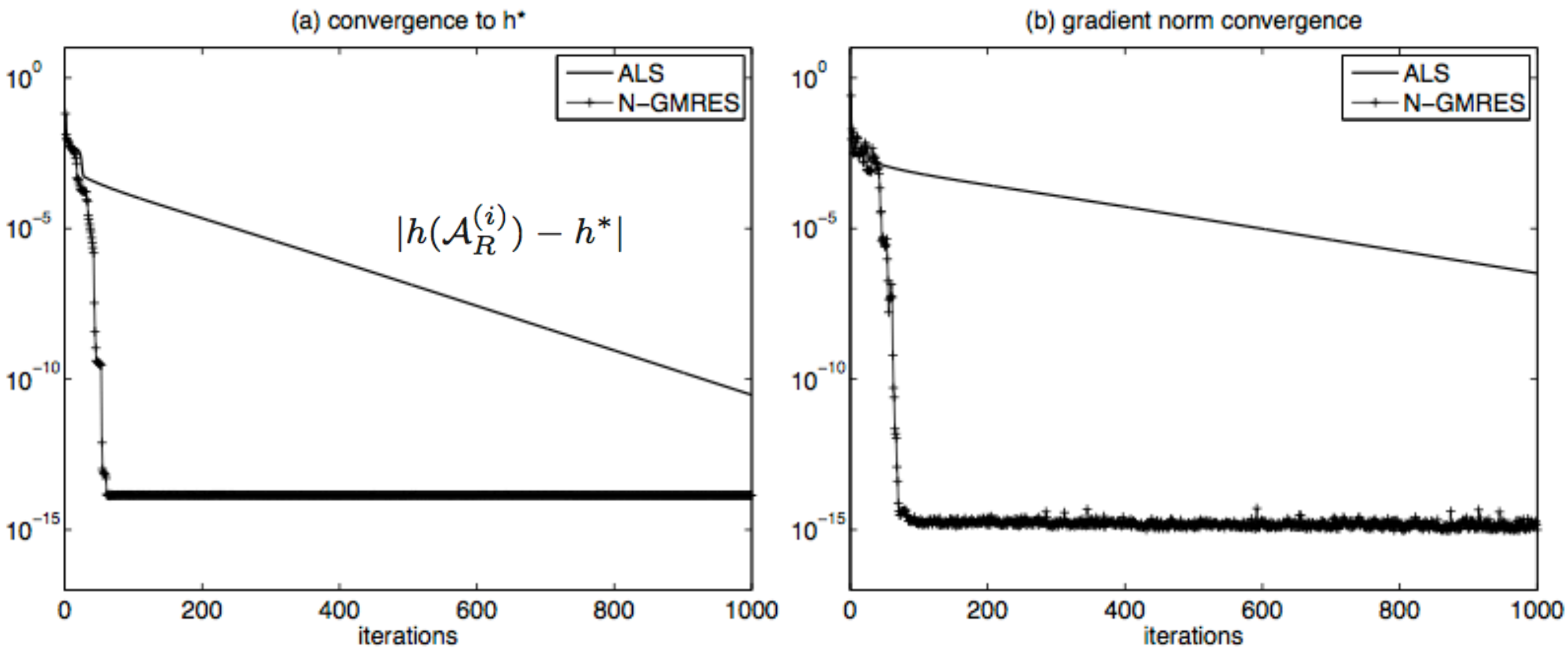
find coefficients $(\alpha_0, \ldots, \alpha_i)$ that minimize

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

UNIVERSITY OF WATERLOO

numerical results for ALS-preconditioned NGMRES 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

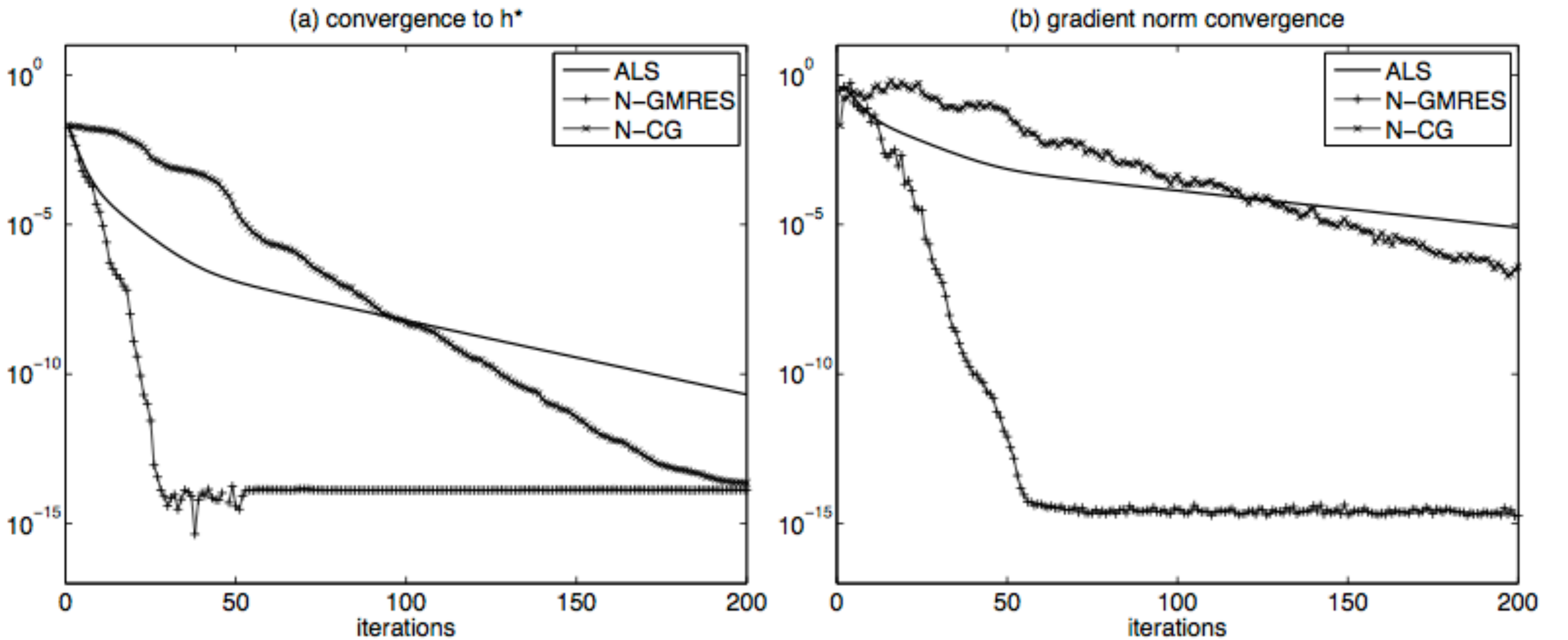


$$h(\mathcal{A}_R^{(i)}) = \frac{\|\mathcal{T} - \mathcal{A}_R^{(i)}\|_F}{\|\mathcal{T}\|_F}$$

UNIVERSITY OF WATERLOO

numerical results: sparse test problem

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



UNIVERSITY OF WATERLOO

(4) comparing NGMRES to GMRES: convergence acceleration for linear systems

- simple iterative method: Gauss-Seidel (GS)

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

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$$

$$(\mathbf{D} - \mathbf{L} - \mathbf{U})\,\mathbf{u} = \mathbf{b}$$

$$(\mathbf{D} - \mathbf{L})\,\mathbf{u}_{i+1} = \mathbf{U}\,\mathbf{u}_i + \mathbf{b}$$

$$\mathbf{r}_i = \mathbf{b} - \mathbf{A}\,\mathbf{u}_i$$

$$\boxed{\mathbf{u}_{i+1} = \mathbf{u}_i + (\mathbf{D} - \mathbf{L})^{-1}\,\mathbf{r}_i}$$

- stationary iterative method:

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

$$\mathbf{M}^{-1} \approx \mathbf{A}^{-1}$$

UNIVERSITY OF WATERLOO

comparing NGMRES to GMRES

(Washio and Oosterlee, ETNA, 1997) ("Anderson mixing")

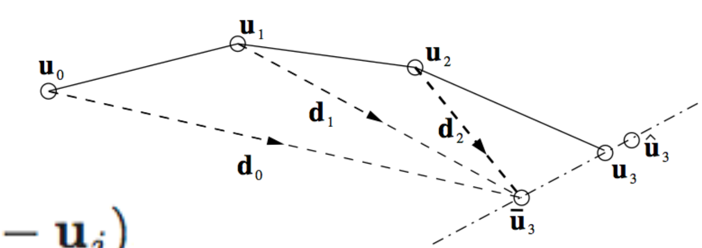
$$\mathbf{A}\,\mathbf{u} = \mathbf{b}$$

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

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

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

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



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

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

UNIVERSITY OF WATERLOO

(5) NGMRES as a general 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. preconditioned NGMRES as a general optimization method?

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

  = update in direction of steepest descent

  (or: use NGMRES to accelerate steepest descent)

UNIVERSITY OF WATERLOO

steepest-descent preconditioning

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

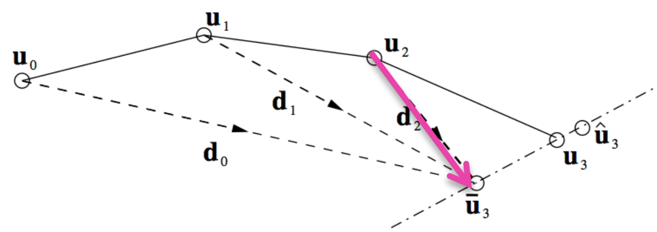
$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$

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

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

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

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



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 NGMRES (note also: link with fixed-point equation)

UNIVERSITY OF WATERLOO

steepest-descent preconditioning

- claim: steepest descent is the 'natural' preconditioner for NGMRES
- example: consider simple quadratic optimization problem

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

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

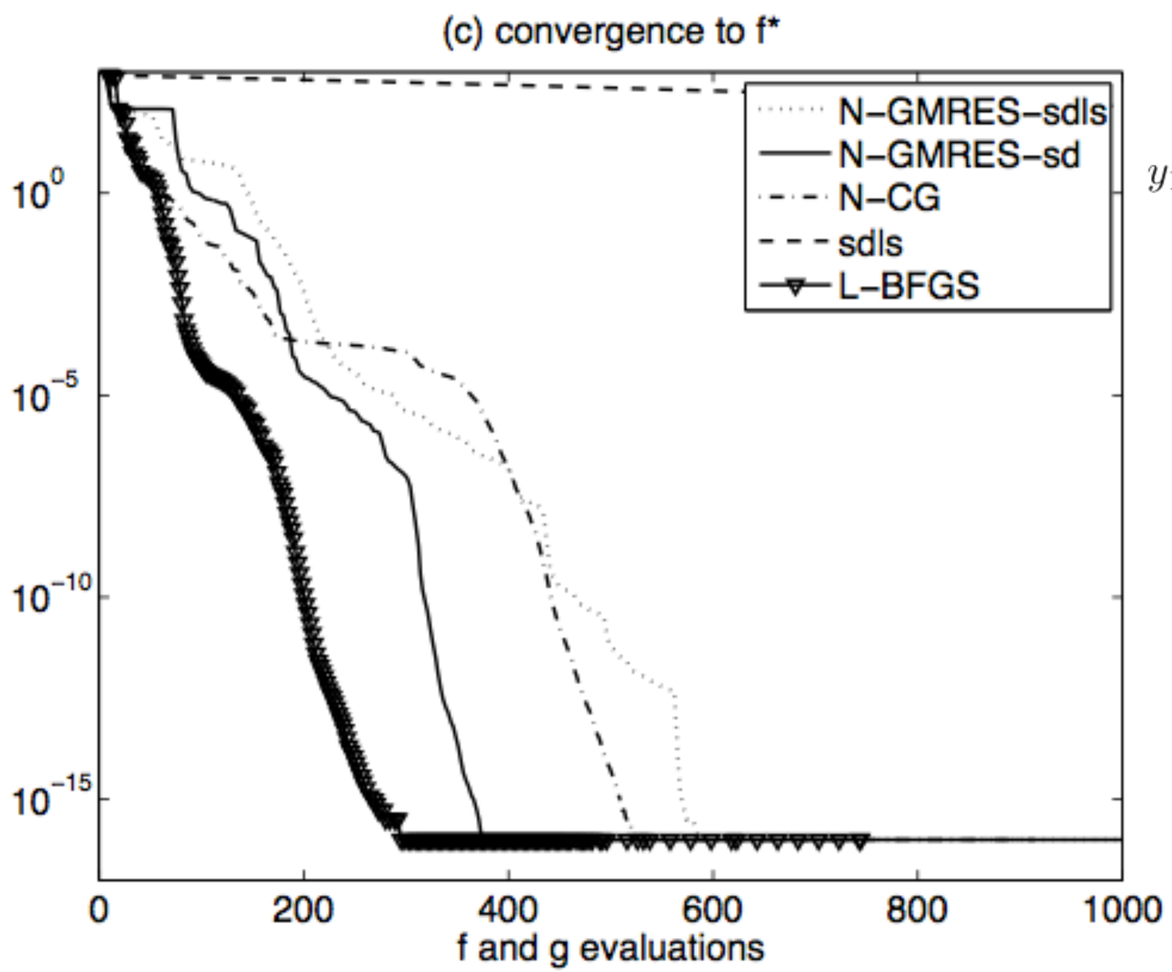
$$\bar{\mathbf{u}}_{i+1} = \mathbf{u}_i - \beta\,\frac{\nabla f(\mathbf{u}_i)}{\|\nabla f(\mathbf{u}_i)\|}$$ becomes $$\boxed{\bar{\mathbf{u}}_{i+1} = \mathbf{u}_i + \beta\,\frac{\mathbf{r}_i}{\|\mathbf{r}_i\|}}$$

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

(and: small step is sufficient)

UNIVERSITY OF WATERLOO

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

UNIVERSITY OF WATERLOO

convergence of steepest-descent preconditioned N-GMRES optimization

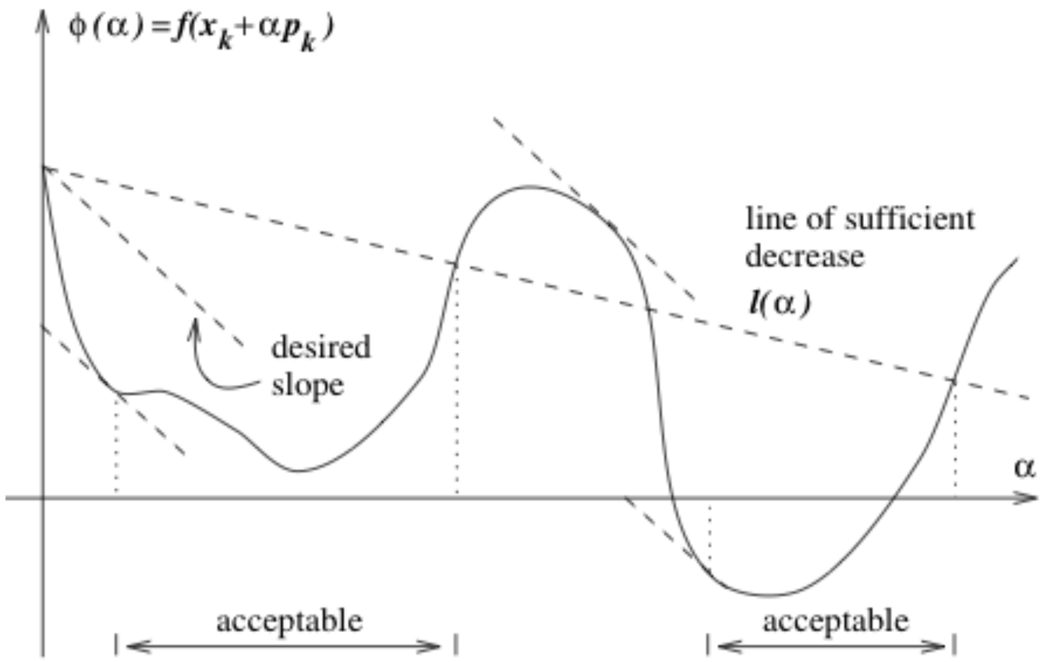
- assume line searches give solutions that satisfy Wolfe conditions:

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:

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



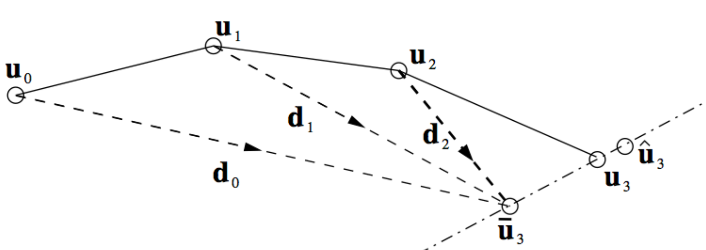
(Nocedal and Wright, 2006)

UNIVERSITY OF WATERLOO

convergence of steepest-descent preconditioned NGMRES 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*

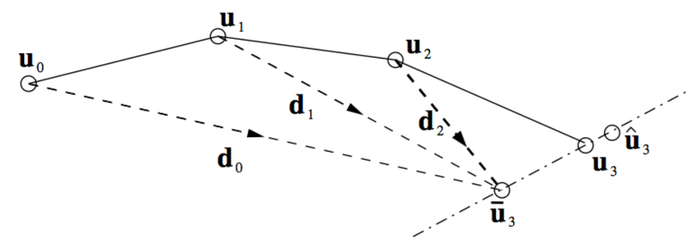
$$\lim_{i\to\infty} \|\nabla f(\mathbf{u}_i)\| = 0. \tag{2.13}$$



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

(6) conclusions



- NGMRES optimization method:
  - extends concept of preconditioned GMRES to nonlinear optimization (*nonlinear* preconditioning)
  - uses iterate recombination (like Anderson acceleration) as an important building block

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)*

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

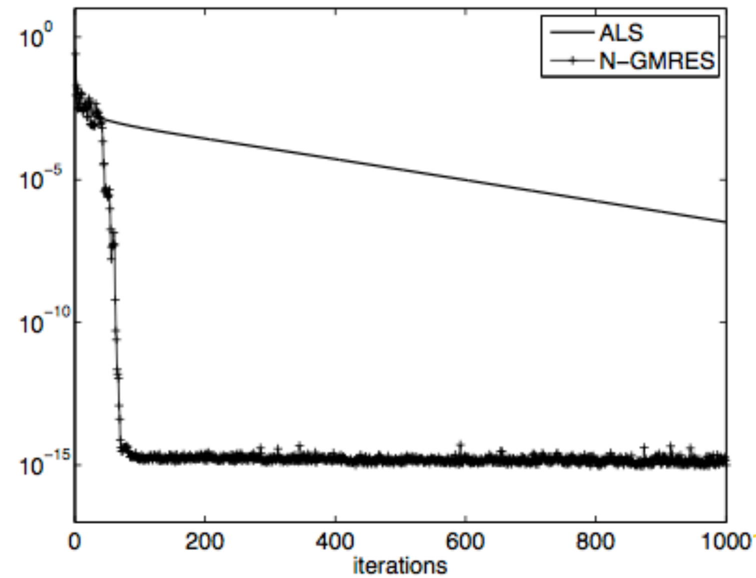
STEP III: *(generate new iterate by line search process)*

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

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

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

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

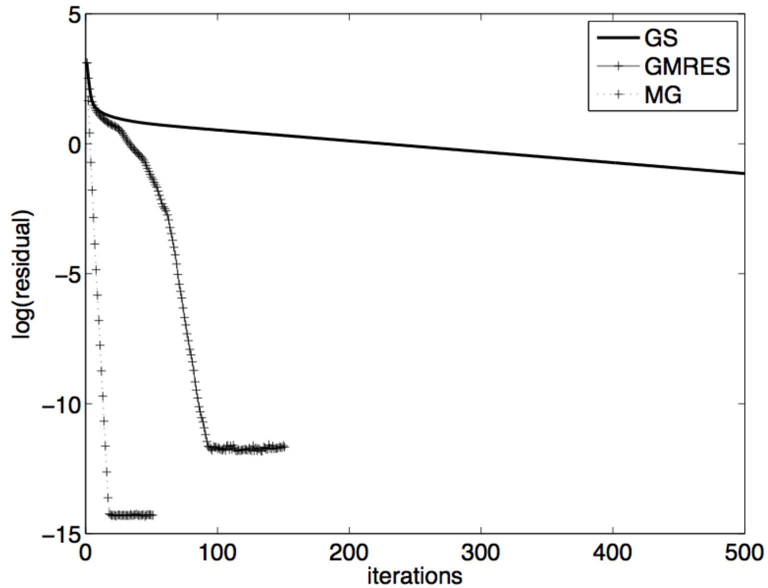


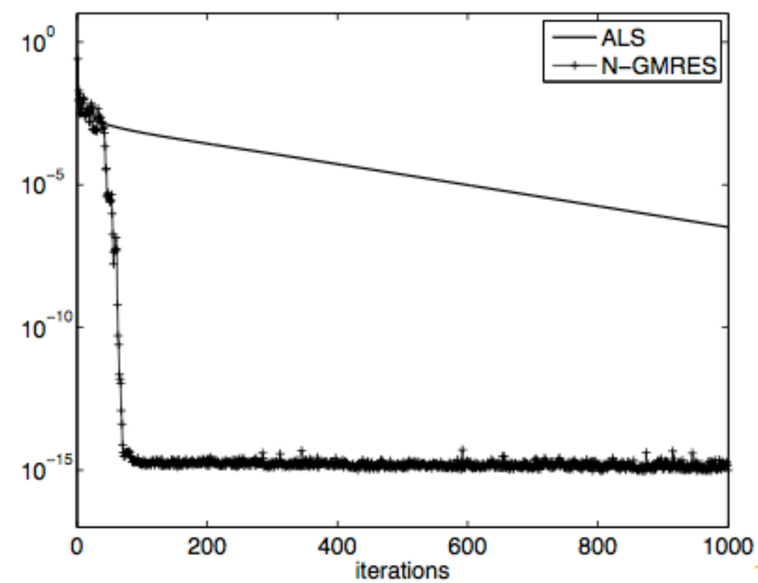
UNIVERSITY OF WATERLOO

conclusions

- NGMRES optimization method:

  accelerates simple iterative optimization method (the nonlinear preconditioner) (ALS)
- just like GMRES accelerates stationary iterative method (preconditioner) for $\mathbf{A}\,\mathbf{u} = \mathbf{b}$



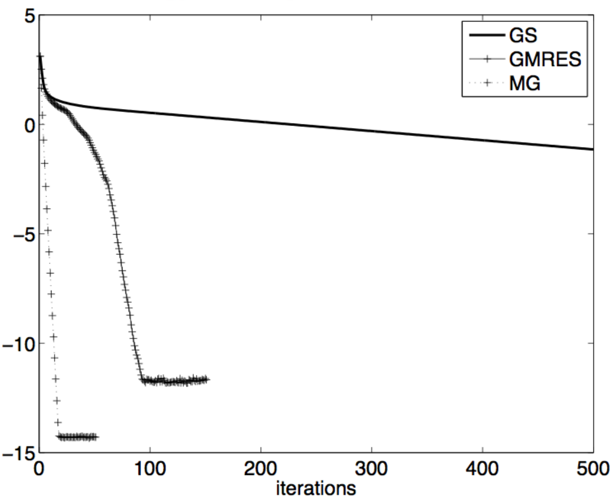


UNIVERSITY OF WATERLOO

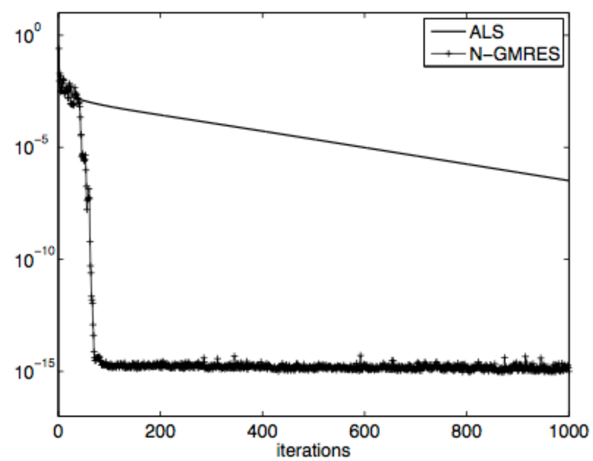
conclusions

- NGMRES (with ALS preconditioning) gives strongly improved solver for canonical tensor decomposition
- note: we can also do nonlinear adaptive algebraic multigrid acceleration for the tensor nonlinear optimization problem, and NCG preconditioned by ALS

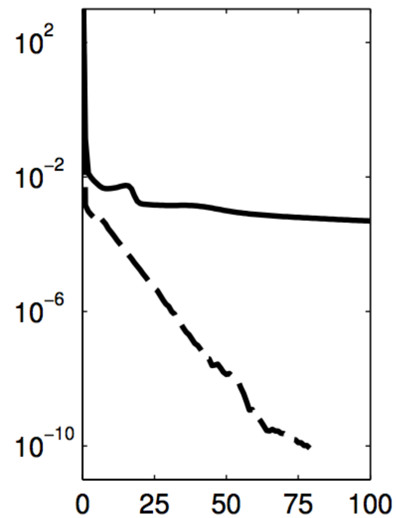
$$\mathbf{A}\,\mathbf{u} = \mathbf{b}$$



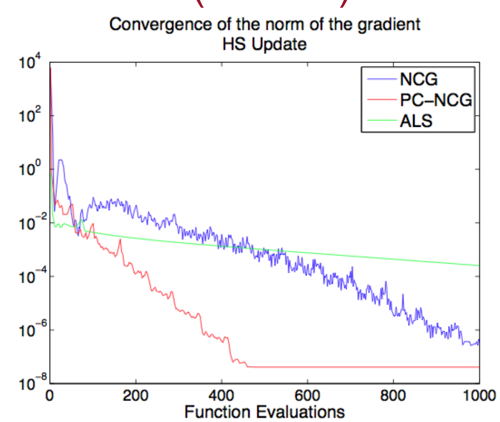
NGMRES



multigrid (Miller)



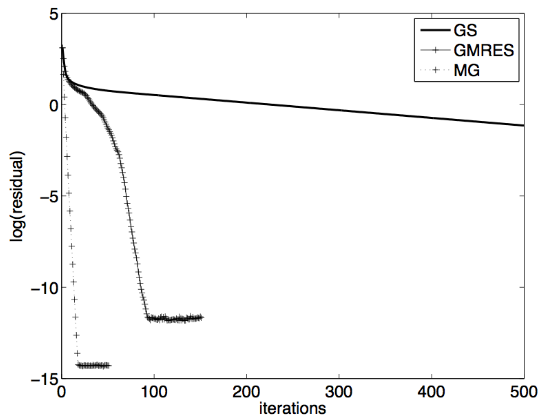
NCG (Winlaw)



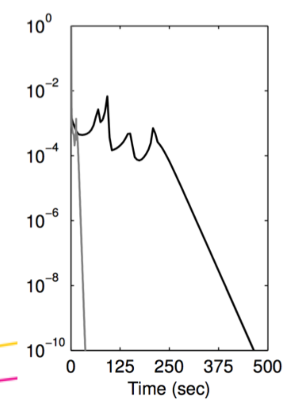
UNIVERSITY OF WATERLOO

conclusions

convergence acceleration for linear systems	convergence acceleration for nonlinear optimization	convergence acceleration for nonlinear systems	convergence acceleration for nonlinear optimization
P-CG	NCG, P? (nonl.)	NCG, P? (nonl.)	P-NCG
P-GMRES	?	P-NGMRES (Washio and Oosterlee, Anderson)	P-NGMRES for optimization
MG	?	FAS	adaptive AMG-FAS for optimization



'nonlinear preconditioning' viewpoint is key



UNIVERSITY OF WATERLOO

- thank you
- questions?

- Hans De Sterck, *'A Nonlinear GMRES Optimization Algorithm for Canonical Tensor Decomposition'*, SIAM J. Sci. Comp., 2012
- Hans De Sterck, *'Steepest Descent Preconditioning for Nonlinear GMRES Optimization'*, Numer. Linear Algebra Appl., 2012
- Hans De Sterck and Killian Miller, *'An Adaptive Algebraic Multigrid Algorithm for Low-Rank Canonical Tensor Decomposition'*, SIAM J. Sci. Comp., 2013

UNIVERSITY OF WATERLOO

a few notes on related methods for $\mathbf{g}(\mathbf{u}^*) = 0$

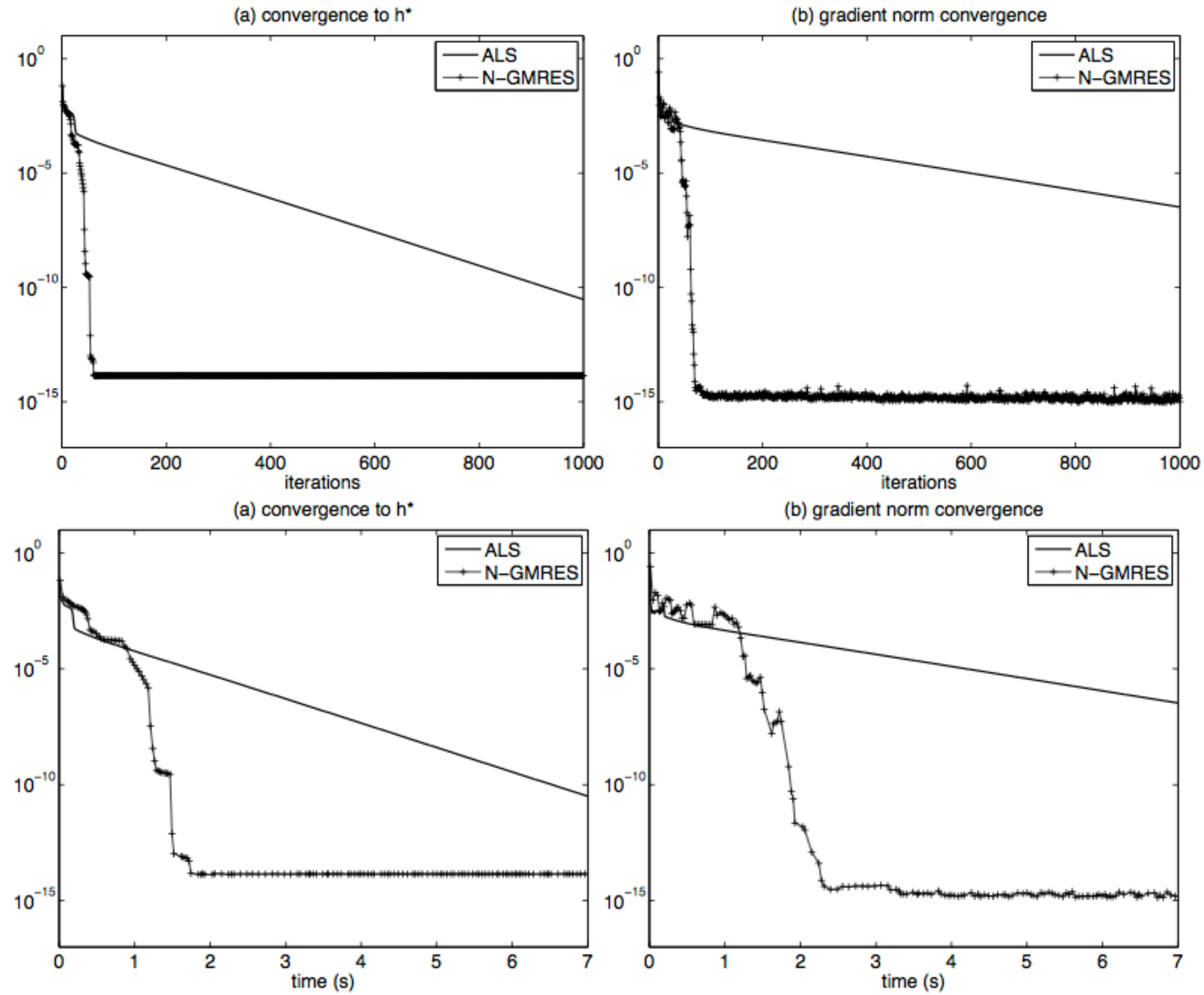
- acceleration step in N-GMRES is similar to Anderson acceleration and DIIS, but not exactly the same
- "mathematical equivalence" with GMRES in the linear case is discussed by Washio and Oosterlee (1997), Walker and Ni (2011), Rohwedder and Schneider (2011), and others
- equivalence of Anderson/DIIS with certain multisecant update methods (Broyden) is discussed by Fang and Saad (2009), Rohwedder and Schneider (2011), and others
- 'nonlinear preconditioning' for N-CG, L-BFGS, multisecant methods is not commonly considered
- 'nonlinear preconditioning' view is natural for the N-GMRES optimization framework

a few more notes on related methods for $\mathbf{g}(\mathbf{u}^*) = 0$

- as mentioned before, there are quite a few other papers that present related ideas for $\mathbf{g}(\mathbf{u}^*) = 0$:
  - Eirola and Nevanlinna, Accelerating with rank-one updates, 1989
  - Brown and Saad, Hybrid Krylov methods for nonlinear systems of equations, 1990
  - Vuik and van der Vorst, A comparison of some GMRES-like methods, 1992
  - Saad (1993): flexible GMRES
  - Fokkema, Sleijpen and van der Vorst, Accelerated Inexact Newton Schemes for Large Systems of Nonlinear Equations, 1998
  - etc.

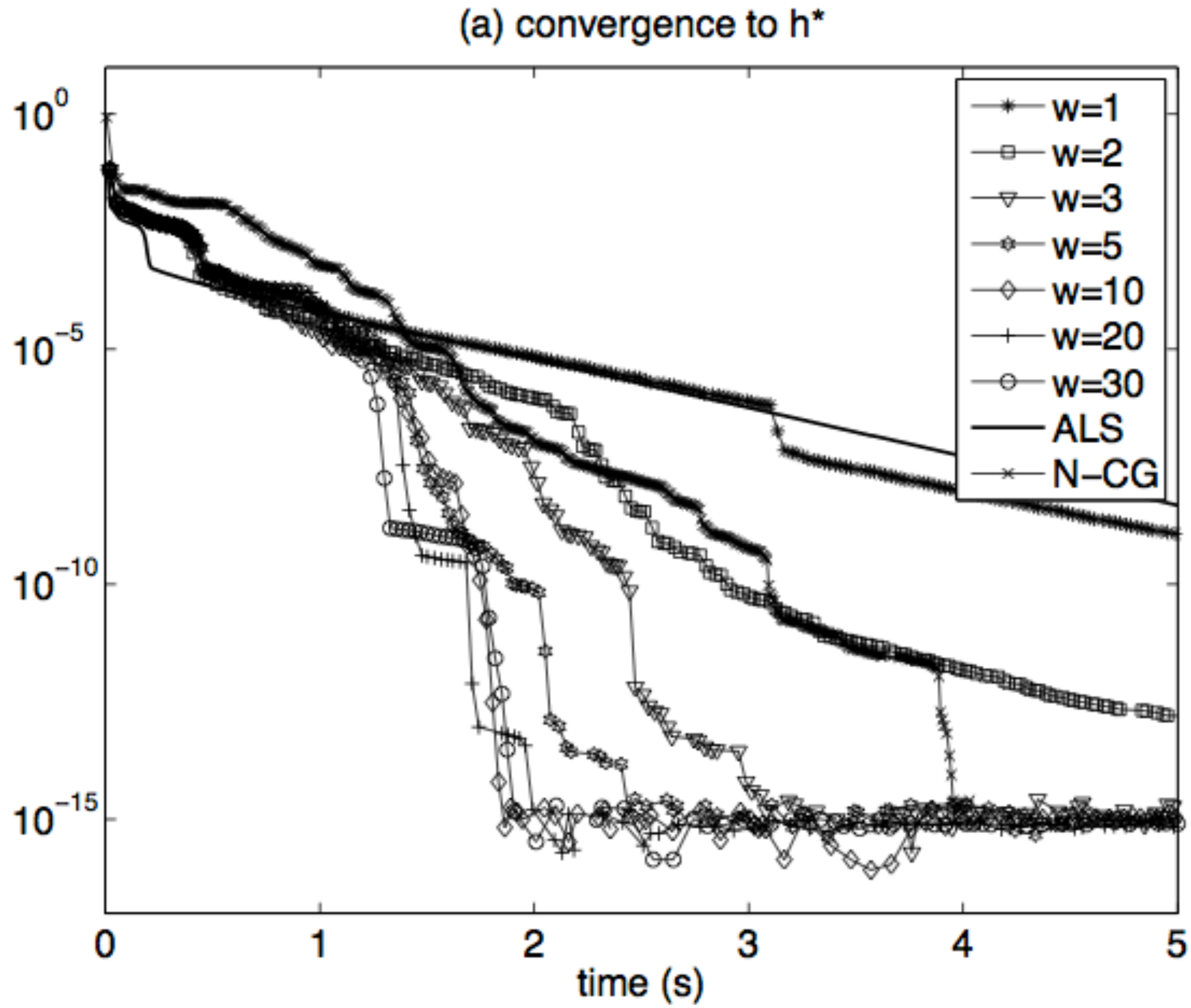
UNIVERSITY OF WATERLOO

numerical results: dense test problem



UNIVERSITY OF
WATERLOO

dense test problem: optimal window size



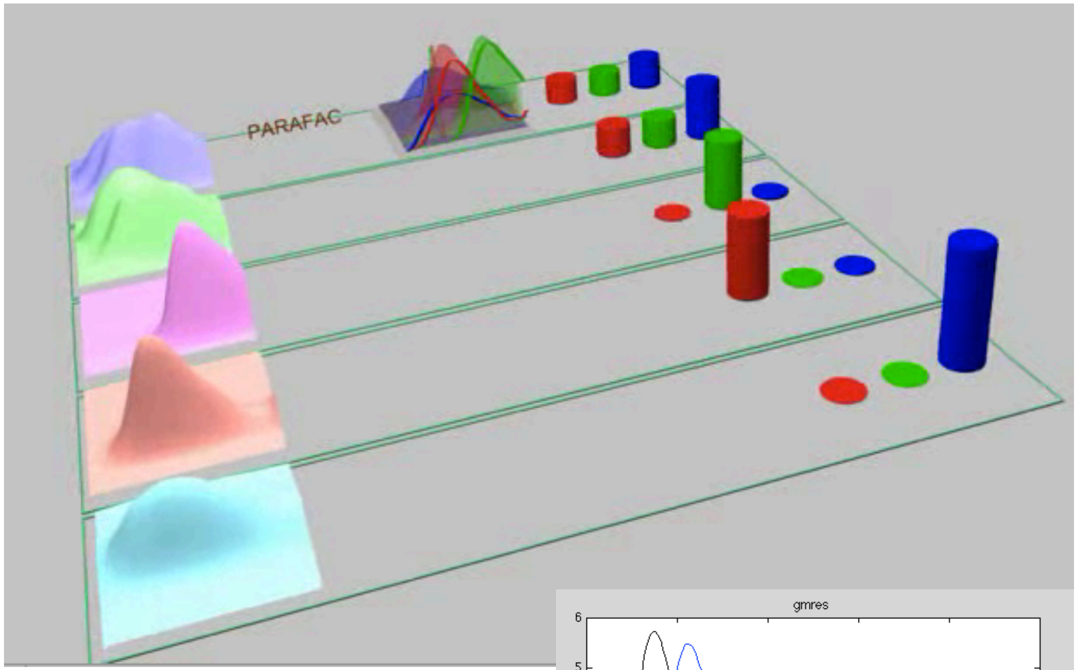
UNIVERSITY OF WATERLOO

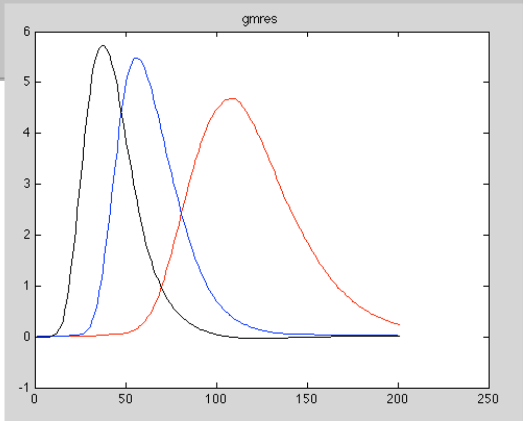
tensor approximation applications

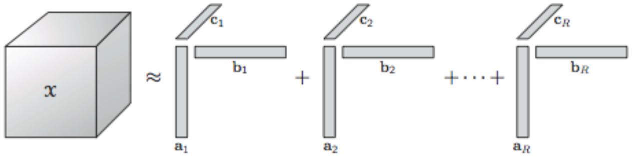
(2) chemometrics: analyze spectrofluorometer data (dense) (Bro et al.,

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

dense test problem: comparison

h^* accuracy 10^{-10}		ALS		N-GMRES		N-CG	
problem parameters		it	time	it	time	it	time
1	$s=20$, $c=0.5$, $R=3$, $l_1=1$, $l_2=1$	37	**0.16**	22	0.3	52	0.24
2	$s=20$, $c=0.5$, $R=5$, $l_1=10$, $l_2=5$	37	**0.28**	17	0.39	97	0.7
3	$s=20$, $c=0.9$, $R=3$, $l_1=0$, $l_2=0$	>1600	>6.9	189	**2.4**	>400	>6.1
4	$s=20$, $c=0.9$, $R=5$, $l_1=1$, $l_2=1$	>1200	>8.6	139	**4.5**	1100	6.8
5	$s=50$, $c=0.5$, $R=3$, $l_1=1$, $l_2=1$	32	**0.23**	16	0.42	67	0.69
6	$s=50$, $c=0.5$, $R=5$, $l_1=10$, $l_2=5$	36	**0.44**	17	0.67	89	1.6
7	$s=50$, $c=0.9$, $R=3$, $l_1=0$, $l_2=0$	>1200	>8.5	104	**3.5**	>553	>7.6
8	$s=50$, $c=0.9$, $R=5$, $l_1=1$, $l_2=1$	1252	14	171	**10**	>1821	>32
9	$s=100$, $c=0.5$, $R=3$, $l_1=1$, $l_2=1$	31	**1**	16	2	136	9.6
10	$s=100$, $c=0.5$, $R=5$, $l_1=10$, $l_2=5$	42	**1.8**	22	4.1	178	16
11	$s=100$, $c=0.9$, $R=3$, $l_1=0$, $l_2=0$	>800	>27	99	**17**	>748	>60
12	$s=100$, $c=0.9$, $R=5$, $l_1=1$, $l_2=1$	1218	51	112	**26**	880	72

TABLE 3.3

(N-CG from Acar, Dunlavy and Kolda, 2011)

UNIVERSITY OF WATERLOO

conclusions

- N-GMRES optimization method:
  - a general, convergent method (with steepest-descent preconditioning)
  - appears competitive with N-CG, L-BFGS
  - ‘nonlinear preconditioning’ viewpoint is key

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

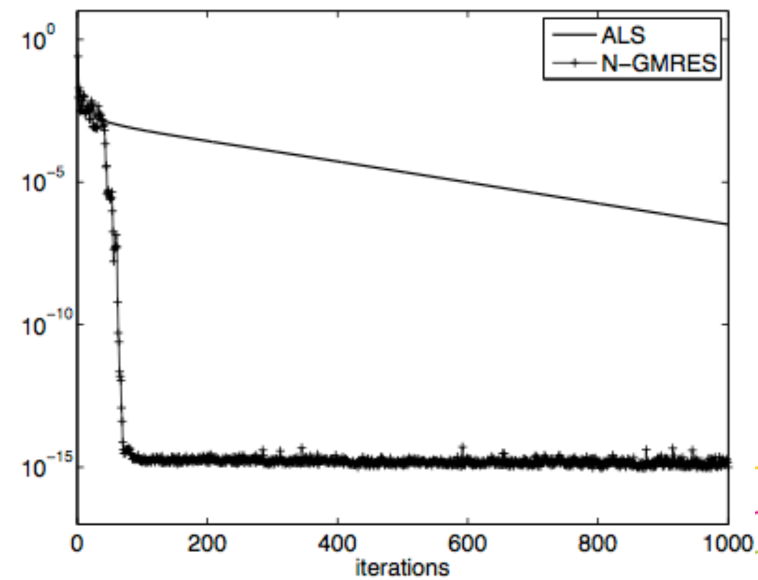
$$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$$

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

open questions: convergence speed of N-GMRES

- GMRES (linear case): convergence rate can be analyzed by considering optimal polynomials etc.
- convergence speed of N-GMRES for optimization: many open questions
  (cfr. Rohwedder and Schneider (2011) for DIIS, superlinear convergence?, multisecant)

- we should try more applications... (to ALS for other tensor decompositions (see e.g. Grasedyck's talk), and to other problems)

conclusions

- real power of N-GMRES: N-GMRES optimization framework can employ sophisticated nonlinear preconditioners (use ALS in tensor case)
- power lies in good preconditioners (like case of GMRES for linear systems)

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

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

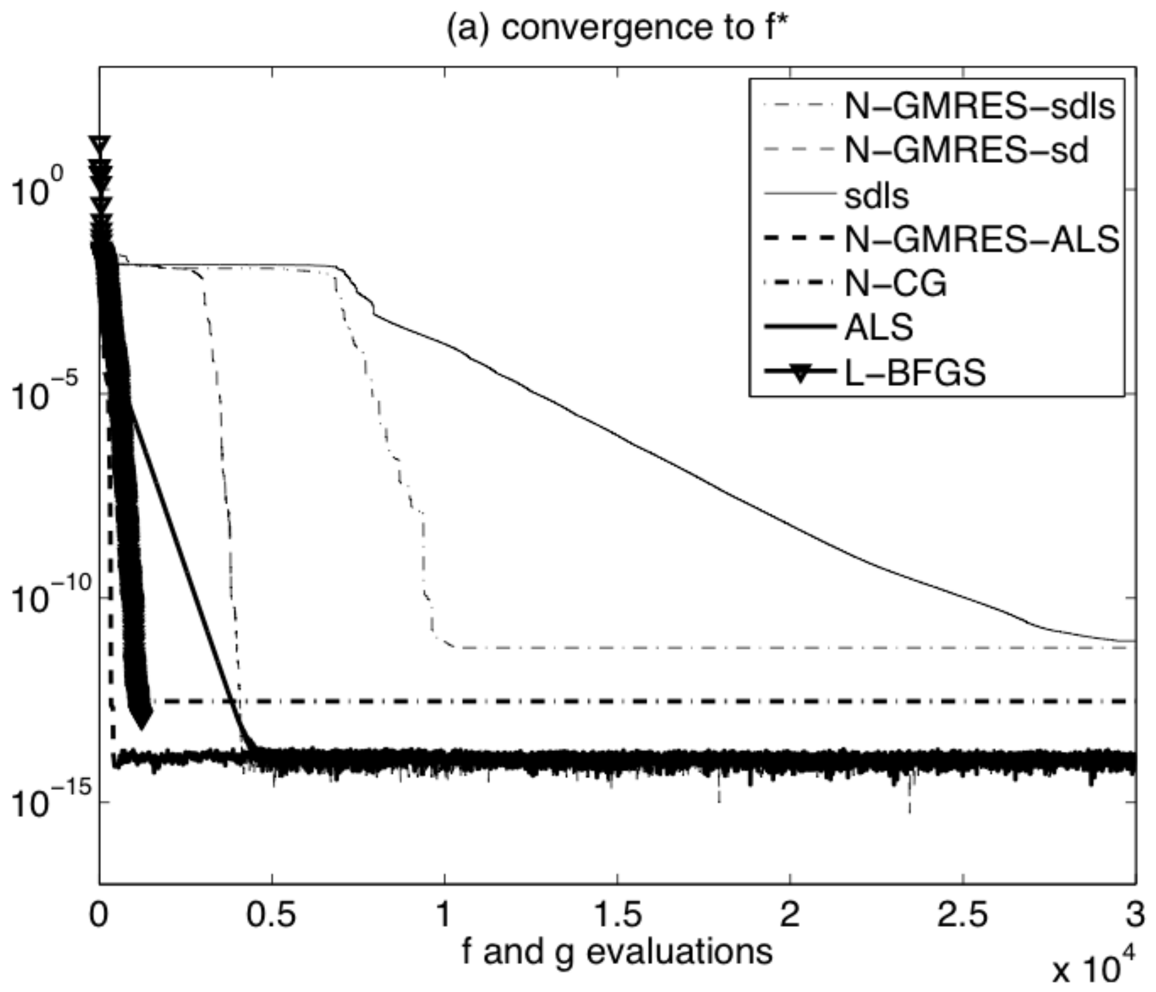
STEP II: *(generate accelerated iterate by nonlinear GMRES step)*

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

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

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

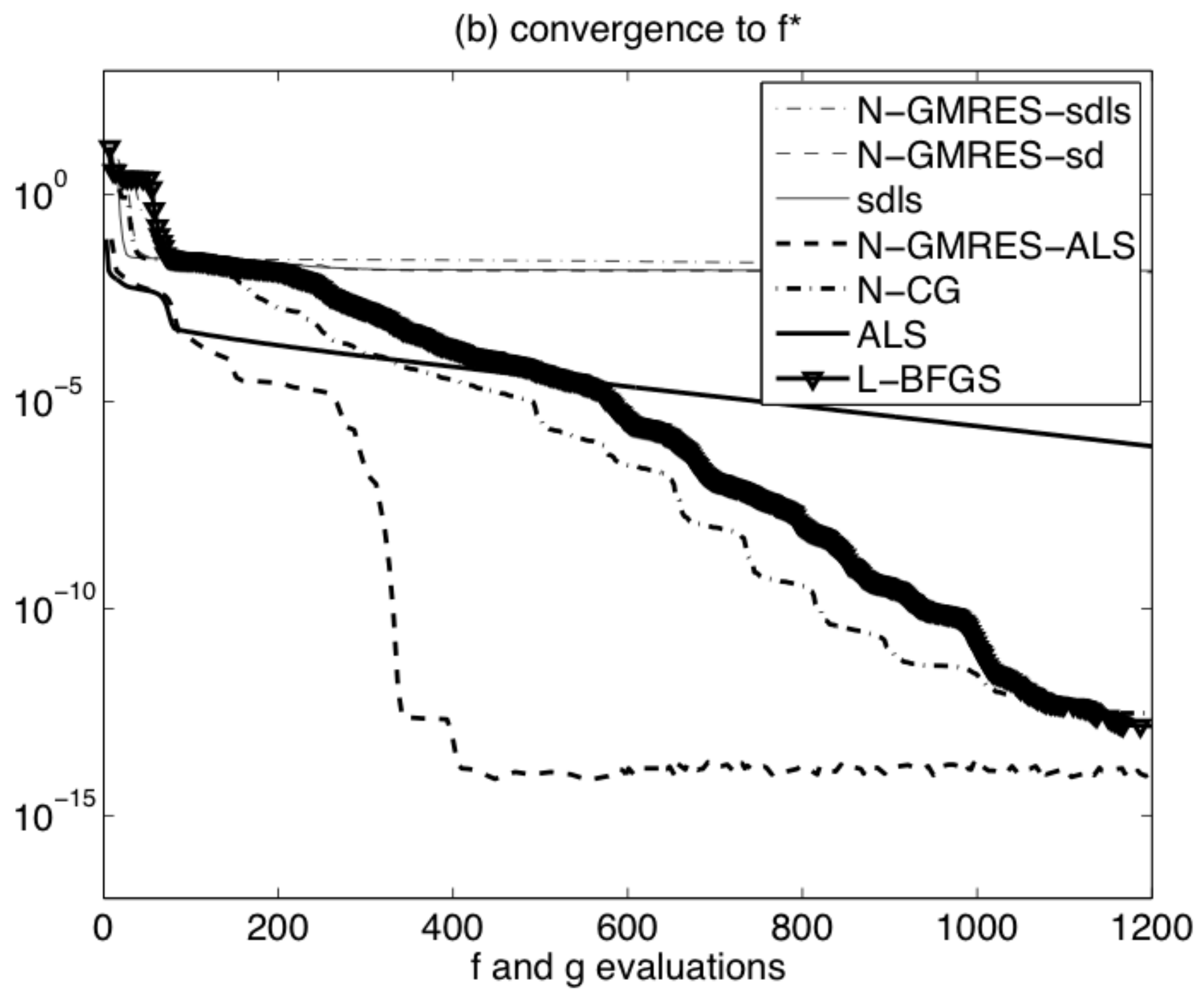
the power of N-GMRES optimization (tensor problem)



steepest descent is not good enough as a preconditioner for N-GMRES (for the tensor problem)

UNIVERSITY OF WATERLOO

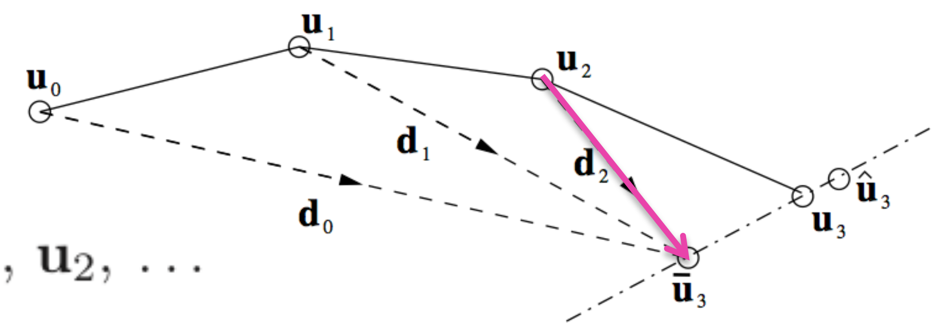
the power of N-GMRES optimization (tensor problem)



ALS is a much better preconditioner!

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 $\mathbf{u}_0,\ \bar{\mathbf{u}}_1,\ \mathbf{u}_1,\ \bar{\mathbf{u}}_2,\ \mathbf{u}_2,\ \ldots$
- 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)*

(note also: Absil and Gallivan, 2009)

UNIVERSITY OF WATERLOO