Extending Preconditioned GMRES to Nonlinear Optimization

UNIVERSITY OF WATERLOO

uwaterloo.ca

Hans De Sterck

Department of Applied Mathematics
University of Waterloo, Canada

SIAM Conference on Applied Linear Algebra
Valencia, Spain, June 2012

this talk is about

“accelerating convergence of iterative *nonlinear optimization* methods”

the approach will be to

“extend concepts of preconditioned GMRES for *linear systems* to nonlinear optimization”

(note: fully *nonlinear* preconditioning)

hi Evelyne (born May 30, 2012)

1. background: convergence acceleration for linear systems

- start from simple example: finite difference discretization of Poisson equation on unit square with homogeneous Dirichlet boundary conditions

$$\boxed{\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = b(x,y)}$$

$$\boxed{\frac{u_{i+1,j} - 2\,u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2\,u_{i,j} + u_{i,j-1}}{h^2} = b(x_i, y_j)}$$

- linear system: $\mathbf{A}\,\mathbf{u} = \mathbf{b}$

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:

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

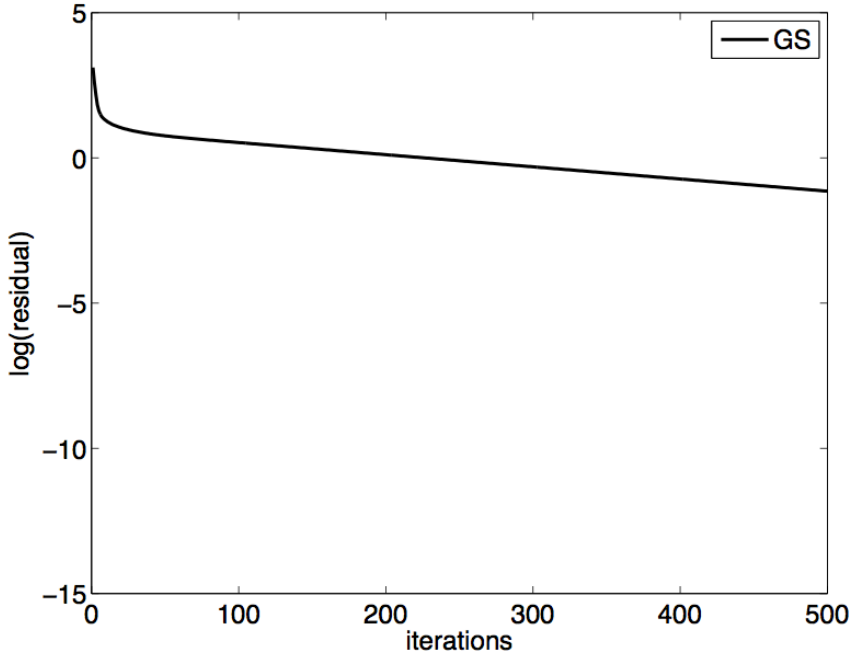
$$\mathbf{M}^{-1}\approx\mathbf{A}^{-1}$$

convergence acceleration for linear systems

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

  yes!

  - GMRES acceleration (generalized minimal residual method)
  - multigrid acceleration



convergence acceleration for linear systems

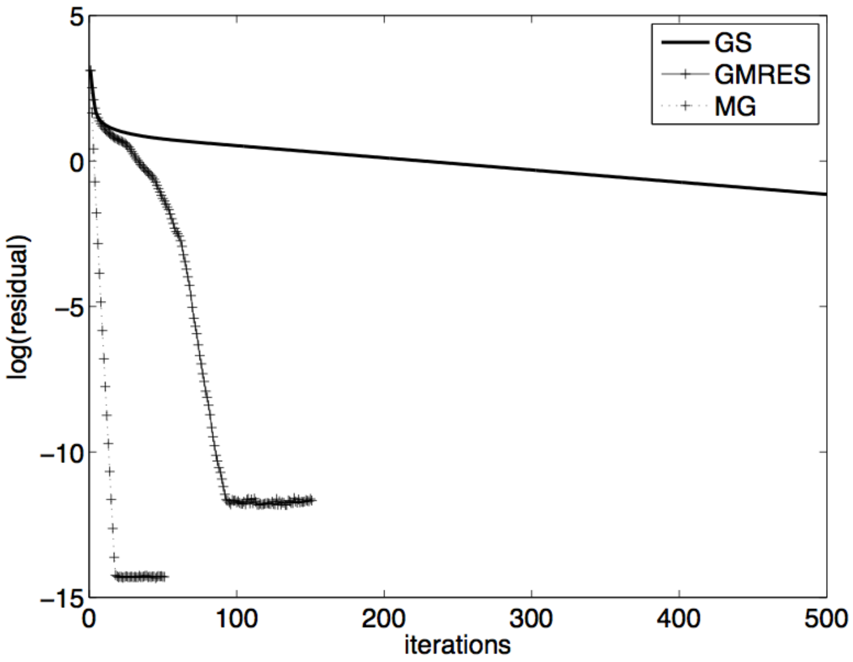
- GS converges slowly
- number of iterations grows as grid is refined
- can we accelerate GS?

  yes!

  – GMRES acceleration

    (generalized minimal residual method)

  – multigrid acceleration



GMRES as an acceleration mechanism

(Washio and Oosterlee, ETNA, 1997)

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

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

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

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

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

GMRES as an acceleration mechanism

(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

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

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

UNIVERSITY OF WATERLOO

GMRES as an acceleration mechanism

(Washio and Oosterlee, ETNA, 1997)

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

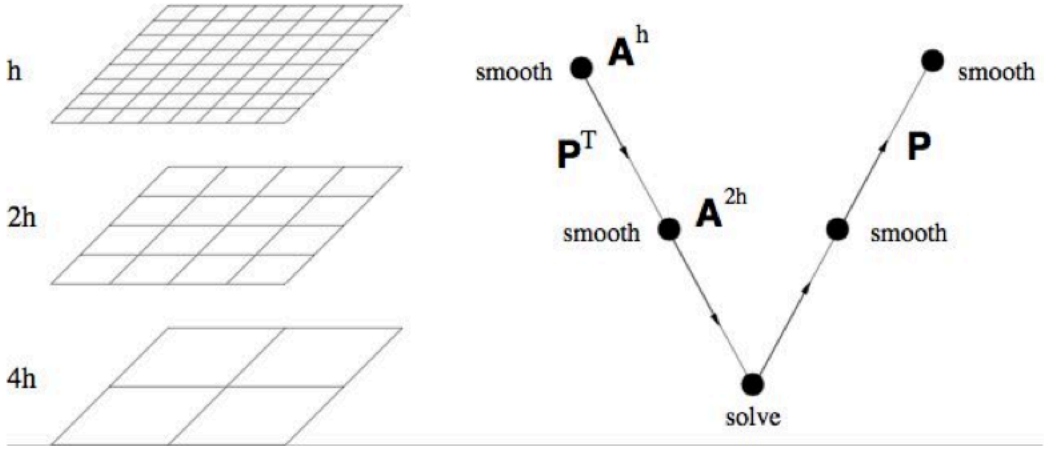
$$\boxed{\mathbf{u}_{i+1} = \mathbf{u}_i + \mathbf{M}^{-1}\,\mathbf{r}_i}$$

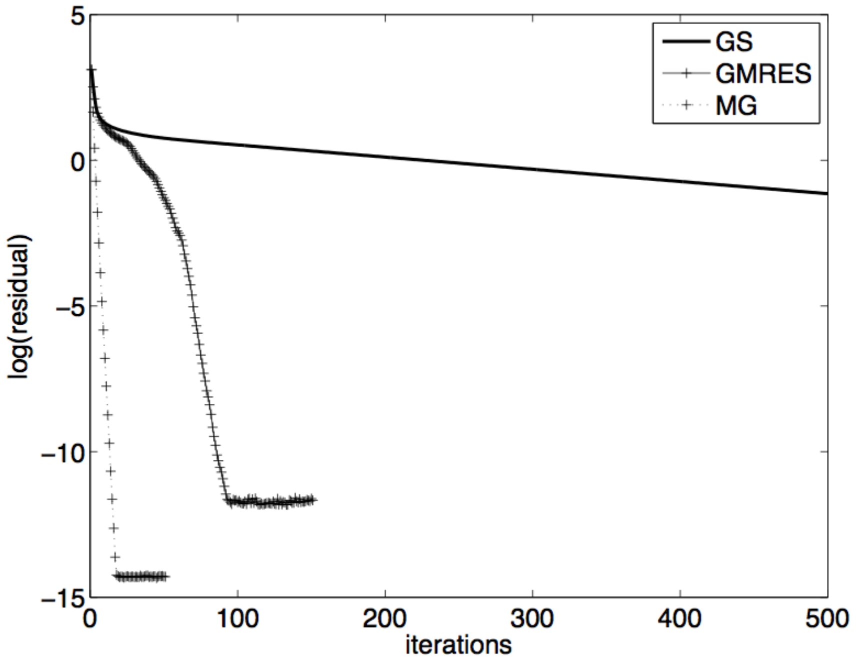
$$\begin{aligned}
V_{1,i+1} &= span\{\mathbf{r}_0, \ldots, \mathbf{r}_i\},\\
V_{2,i+1} &= span\{\mathbf{r}_0, \mathbf{A}\mathbf{M}^{-1}\,\mathbf{r}_0, (\mathbf{A}\mathbf{M}^{-1})^2\,\mathbf{r}_0\}, \ldots, (\mathbf{A}\mathbf{M}^{-1})^i\,\mathbf{r}_0\}\\
&= K_{i+1}(\mathbf{A}\mathbf{M}^{-1}, \mathbf{r}_0),\\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

- seek optimal approximation $\mathbf{M}\,(\hat{\mathbf{u}}_{i+1} - \mathbf{u}_i) = \sum_{j=0}^{i} \beta_j\,\mathbf{M}\,(\mathbf{u}_{i+1} - \mathbf{u}_j)$
  $\hat{\mathbf{u}}_{i+1} = \mathbf{u}_i + \sum_{j=0}^{i} \beta_j\,(\mathbf{u}_{i+1} - \mathbf{u}_j)$ that minimizes $\|\hat{\mathbf{r}}_{i+1}\|_2$
- this is *mathematically "essentially equivalent" to right-preconditioned GMRES*
- the stationary iterative method is the preconditioner
- GMRES accelerates the stationary iterative method

UNIVERSITY OF WATERLOO

alternative: multigrid as an acceleration mechanism



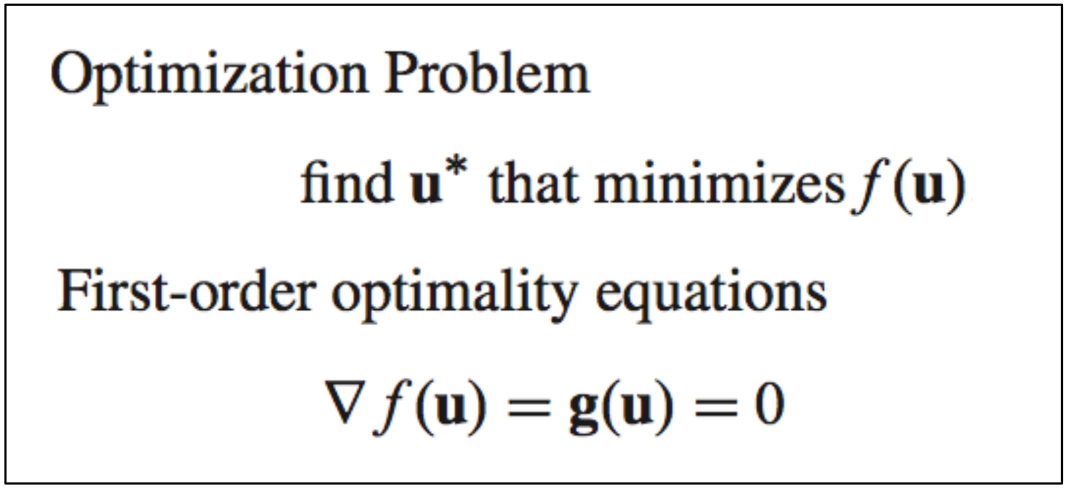


- multigrid accelerates the smoother (Gauss-Seidel)

- GMRES (or: CG, ...) and multigrid are ways to accelerate Gauss-Seidel

2. nonlinear optimization – tensor decomposition

- consider simple iterative optimization methods for smooth nonlinear optimization problem



Optimization Problem

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

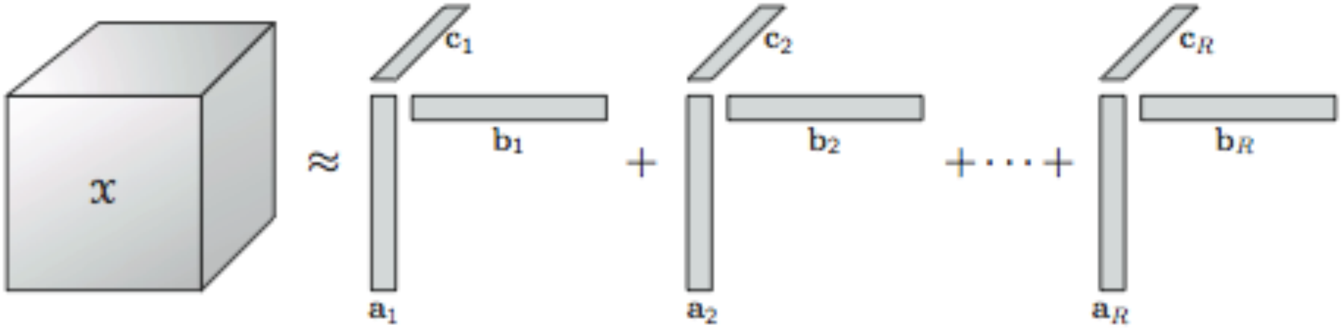
First-order optimality equations

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

- can we accelerate the convergence of simple iterative optimization methods?

application: canonical tensor decomposition

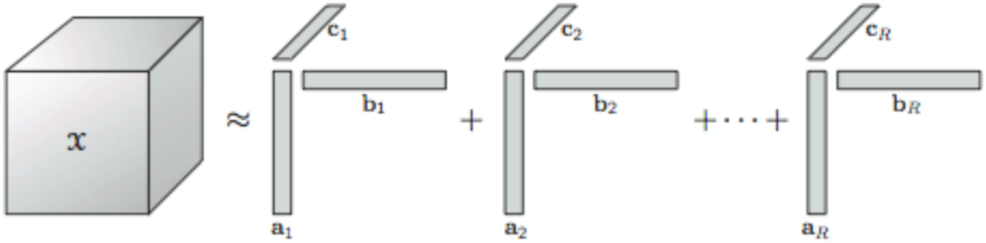
- tensor = element of tensor product of real vector spaces (*N*-dimensional array)
- *N*=3:

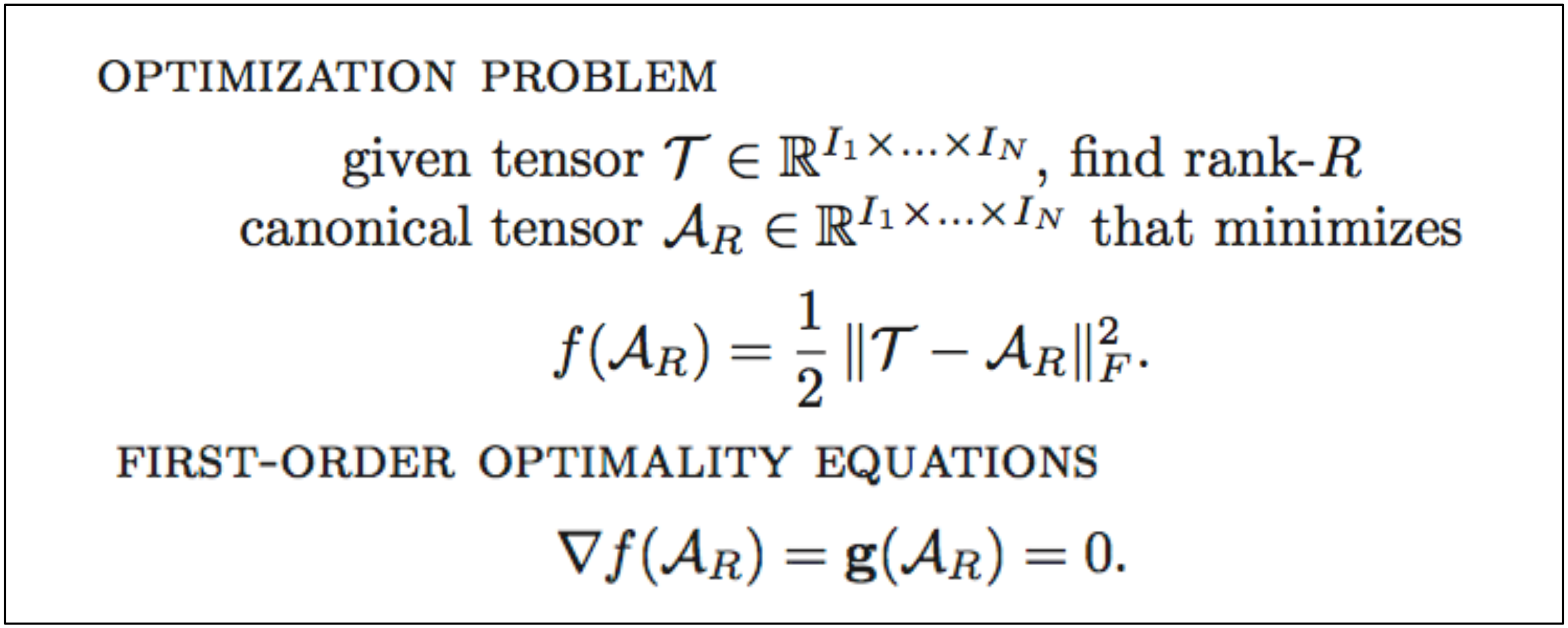


(from “Tensor Decompositions and Applications”, Kolda and Bader, SIAM Rev., 2009 [1])

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

canonical tensor decomposition





OPTIMIZATION PROBLEM

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

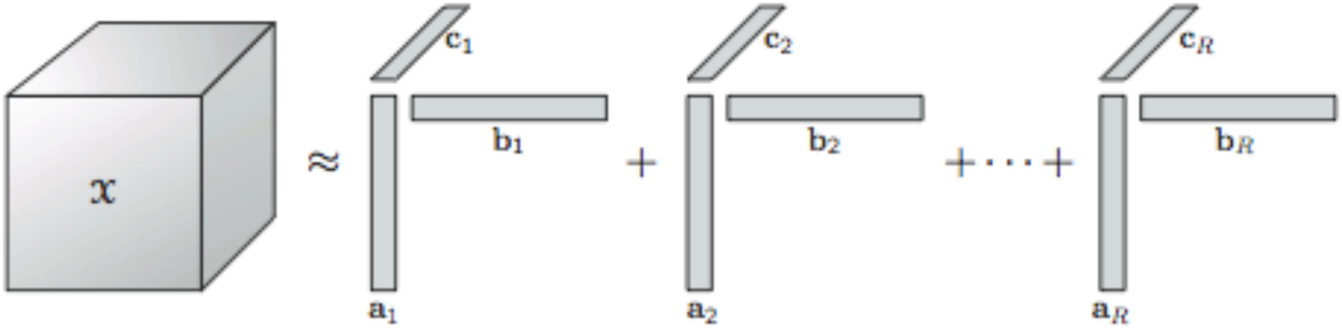
(de Silva and Lim, SIMAX, 2009)

link with singular value decomposition

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

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

- canonical decomposition of tensor



a difference with the SVD

truncated SVD is best rank-*R* approximation:

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

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

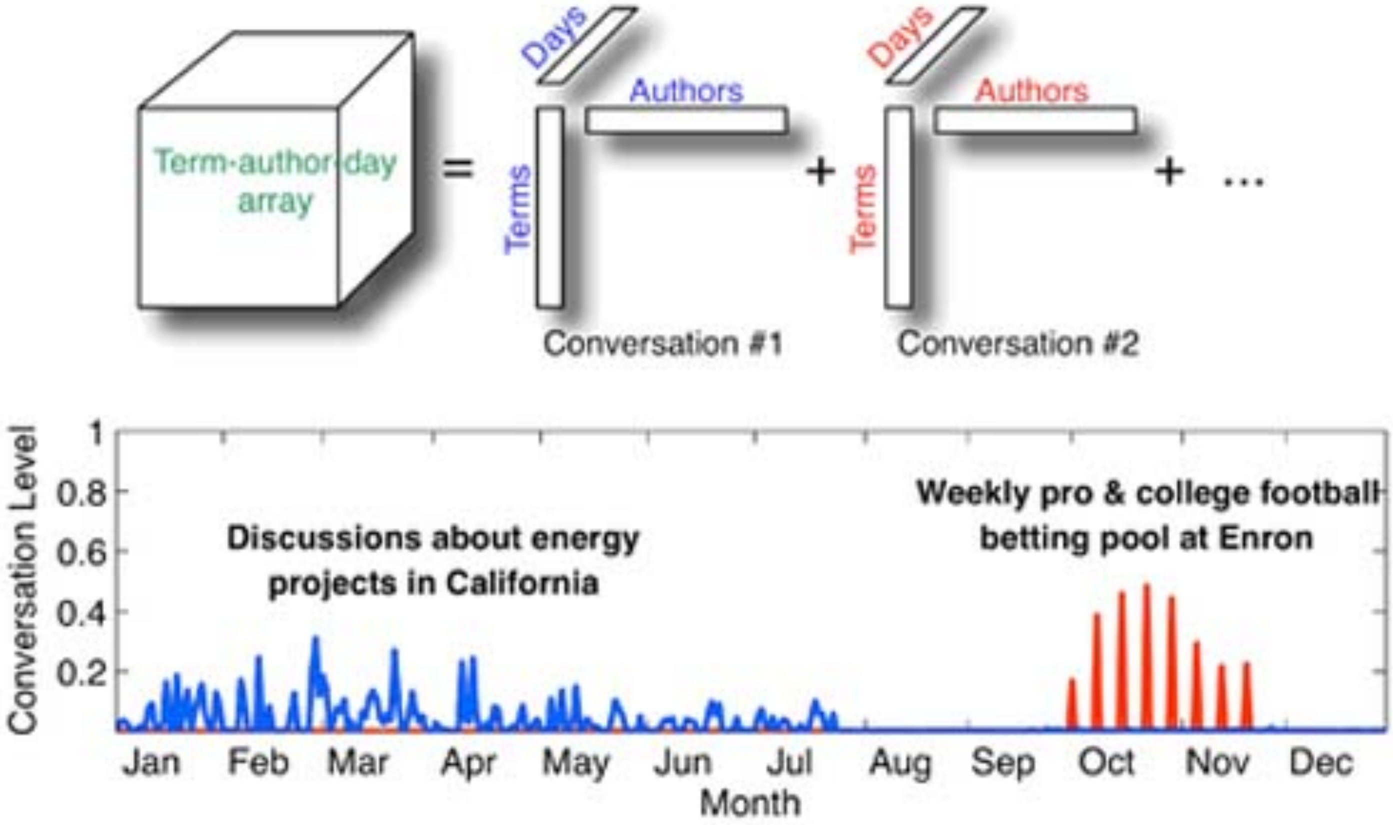
BUT best rank-*R* tensor cannot be obtained by truncation: different optimization problems for different *R*!

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

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

tensor approximation applications

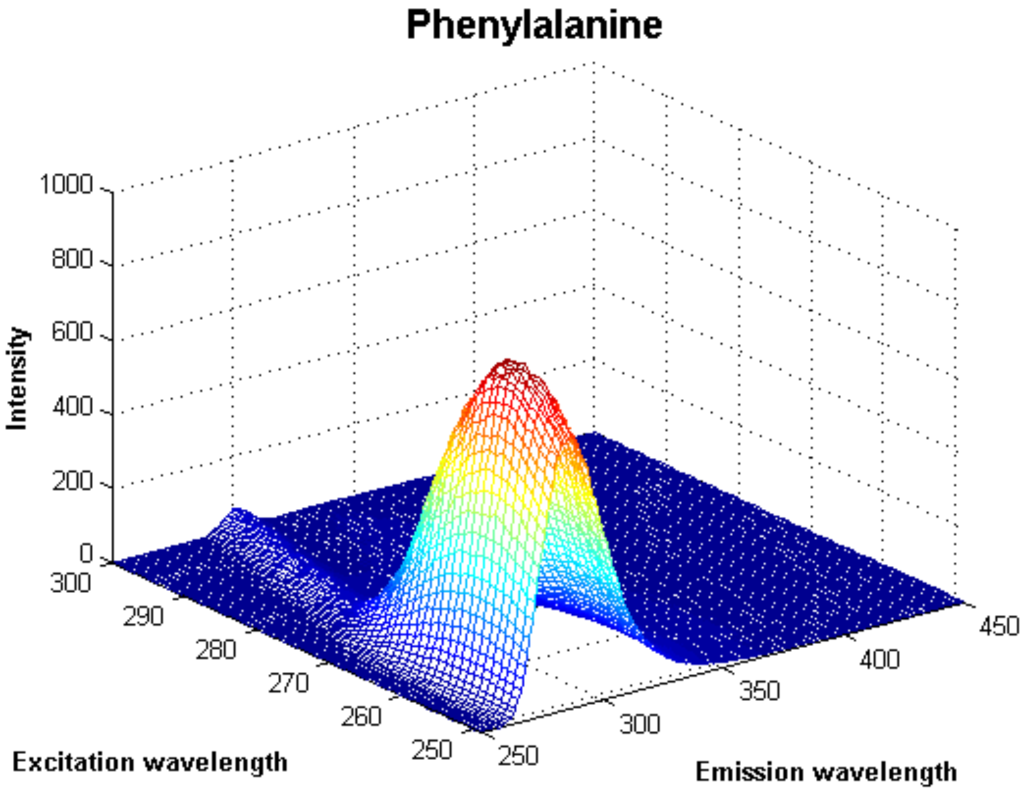
(1) “Discussion Tracking in Enron Email Using PARAFAC” by Bader, Berry and Browne (2008) (sparse, nonnegative)

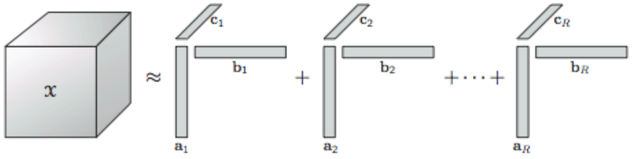


tensor approximation applications

(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])

‘workhorse’ algorithm: alternating least squares (ALS)

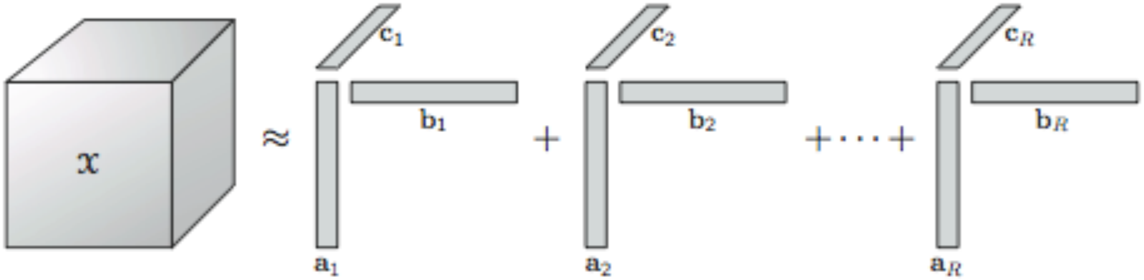
$$f(\mathcal{A}_R) = \frac{1}{2} \left\| \mathcal{T} - \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ a_r^{(3)} \right\|_F^2$$

(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])

alternating least squares (ALS)

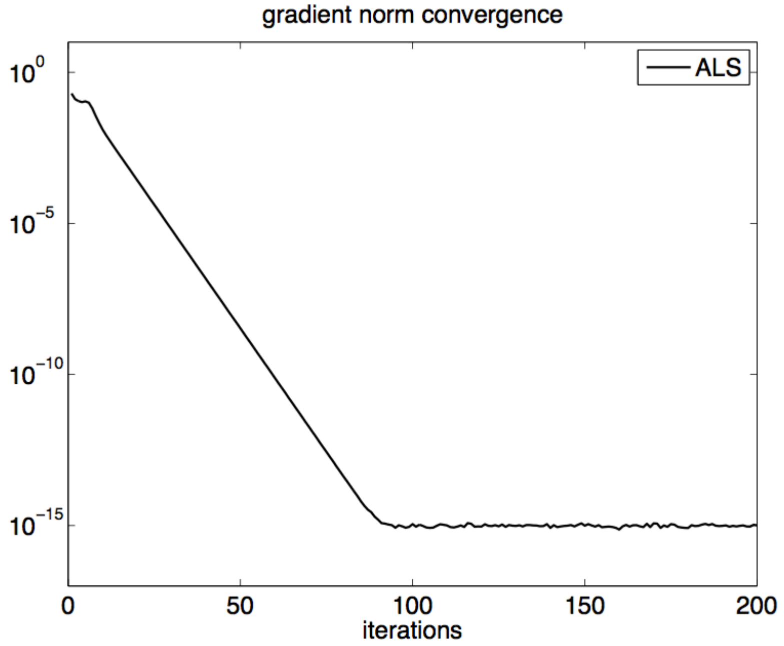
$$f(\mathcal{A}_R) = \frac{1}{2} \left\| \mathcal{T} - \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ a_r^{(3)} \right\|_F^2$$

- “simple iterative optimization method”
- ALS is block nonlinear Gauss-Seidel
- ALS is monotone
- ALS is sometimes fast, but can also be extremely slow (depending on problem and initial condition) (convergence: Uschmajew’s talk)

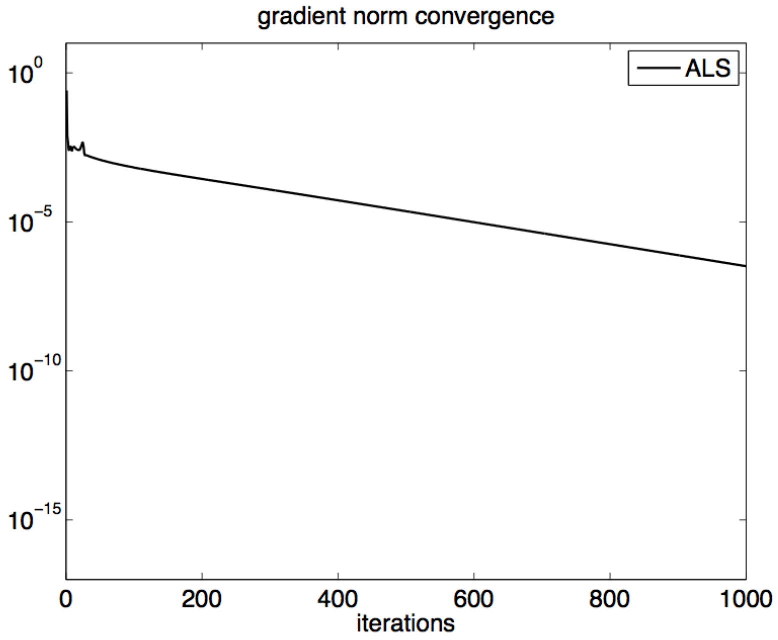
alternating least squares (ALS)

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

fast case



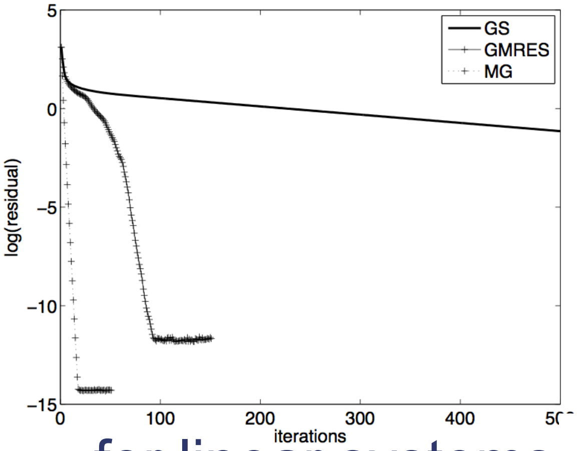
slow case



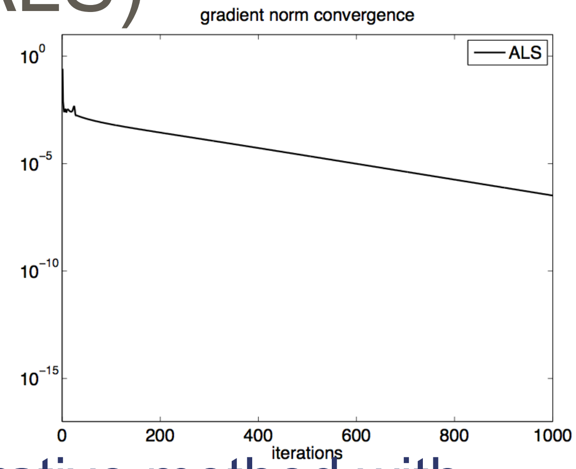
(we used Matlab with Tensor Toolbox (Bader and Kolda) and Poblano Toolbox (Dunlavy et al.) for all computations)

UNIVERSITY OF WATERLOO

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 $\mathbf{A}\,\mathbf{u} = \mathbf{b}$: accelerate simple iterative method with
  - GMRES
  - multigrid
- the simple iterative method is called the ‘preconditioner’
- let’s try to accelerate ALS for the tensor optimization problem
- for nonlinear optimization problems, general approaches to accelerate simple iterative methods (like ALS) appear uncommon (e.g., not in Nocedal and Wright) (“nonlinear preconditioning” does not appear to be a common notion in optimization)
- issues: nonlinear, optimization context

3. existing methods: convergence acceleration for nonlinear systems $\mathbf{g}(\mathbf{u}^*) = 0$

- nonlinear system:

$$\boxed{\mathbf{g}(\mathbf{u}^*) = 0}$$

- equivalent: fixed-point equation

$$\mathbf{u}^* = \mathbf{h}(\mathbf{u}^*)$$

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

(‘simple iterative method’)

$$\hat{\mathbf{g}}(\mathbf{u}^*) = \mathbf{u}^* - \mathbf{h}(\mathbf{u}^*) = 0$$

convergence acceleration for nonlinear systems

- existing convergence acceleration method for nonlinear systems $\mathbf{g}(\mathbf{u}^*) = 0$: ("iterate recombination")
  - Anderson acceleration (1965) (popular in electronic structure calculation)
  - Pulay mixing – direct inversion in the iterative subspace (DIIS) (1980)
  - Washio and Oosterlee (1997): Krylov acceleration for nonlinear multigrid

  (in PETSc)

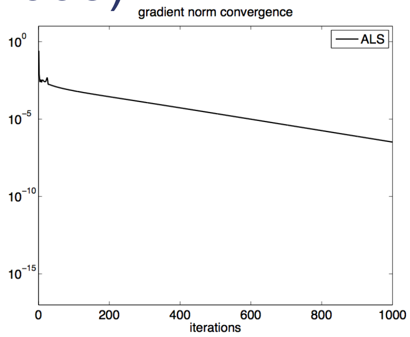
  also related to:

  - GMRES (Saad and Schultz, 1986), flexible GMRES (Saad, 1993)

  recent papers on Anderson mixing/DIIS for nonlinear systems:

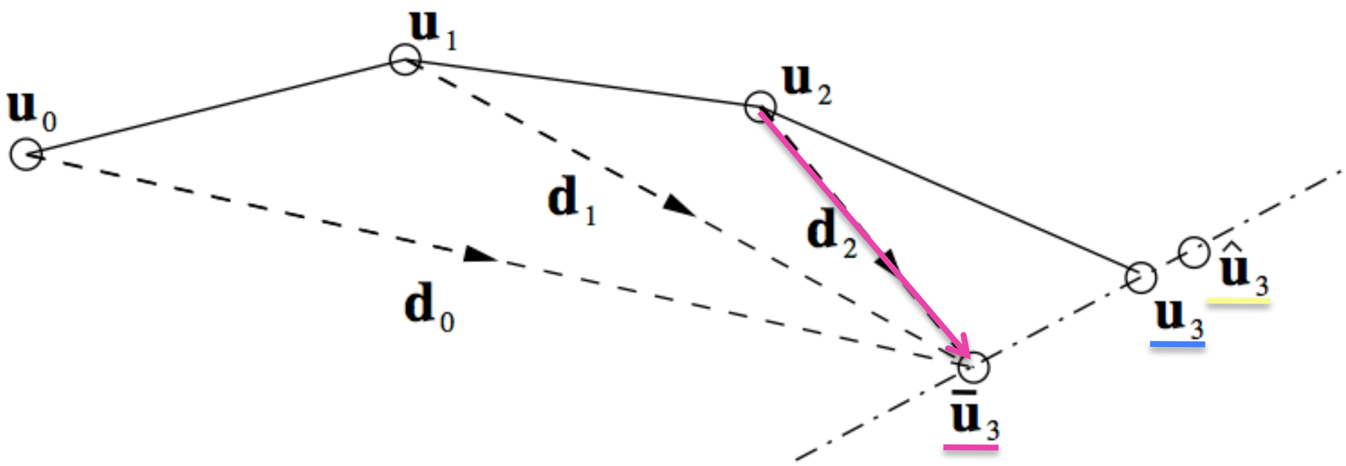
  - Fang and Saad (2009)
  - Walker and Ni (2011)
  - Rohwedder and Schneider (2011)

  plus many other papers with similar ideas and developments



- we will use this type of approach as a building block of our N-GMRES nonlinearly preconditioned optimization algorithm

4. nonlinear GMRES optimization method (N-GMRES)



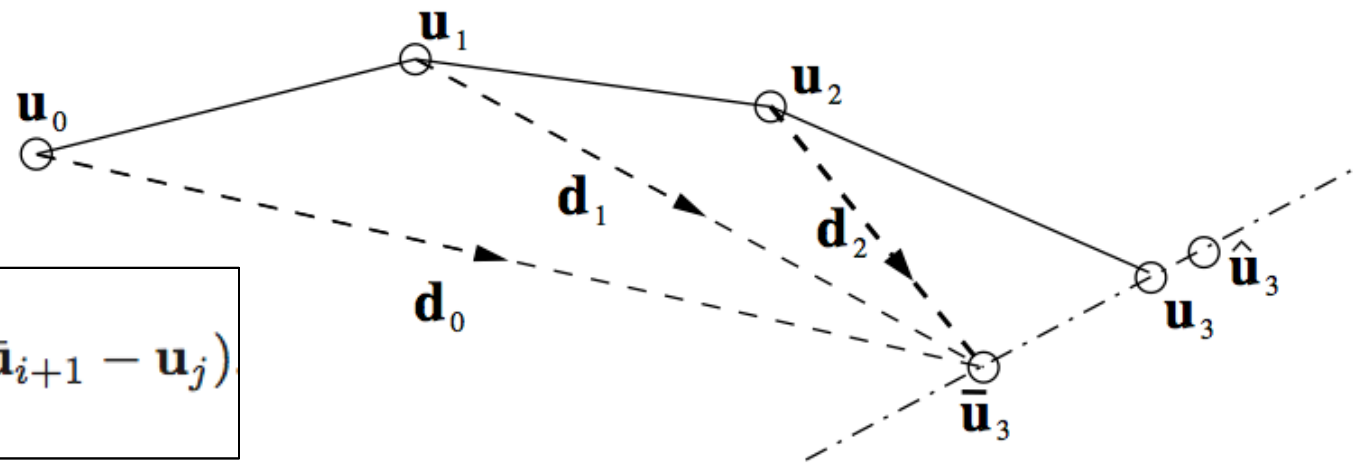
Algorithm 1: N-GMRES optimization algorithm (window size w)

```
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
```

(ALS)

(Moré-Thuente line search, satisfies Wolfe conditions)

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



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

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

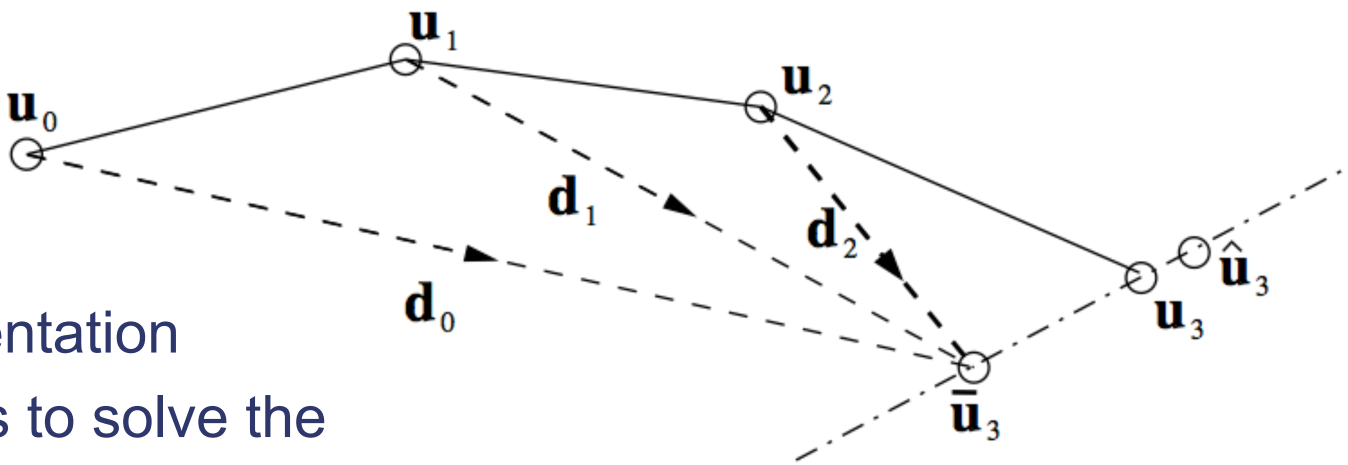
$$\approx \mathbf{g}(\bar{\mathbf{u}}_{i+1}) + \sum_{j=0}^{i} \alpha_j \left(\mathbf{g}(\bar{\mathbf{u}}_{i+1}) - \mathbf{g}(\mathbf{u}_j)\right)$$

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

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

UNIVERSITY OF WATERLOO

step II: N-GMRES 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)

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

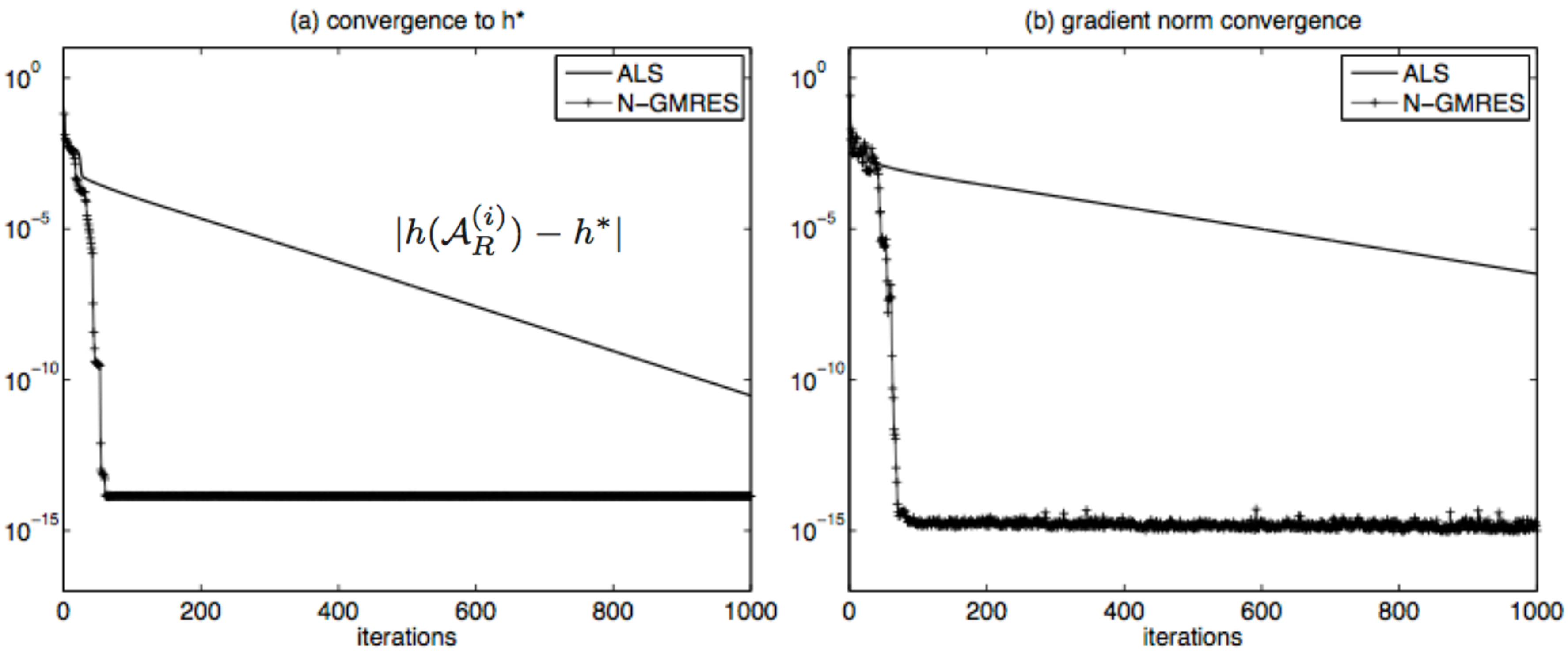
find coefficients $(\alpha_0, \ldots, \alpha_i)$ that minimize

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

UNIVERSITY OF WATERLOO

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

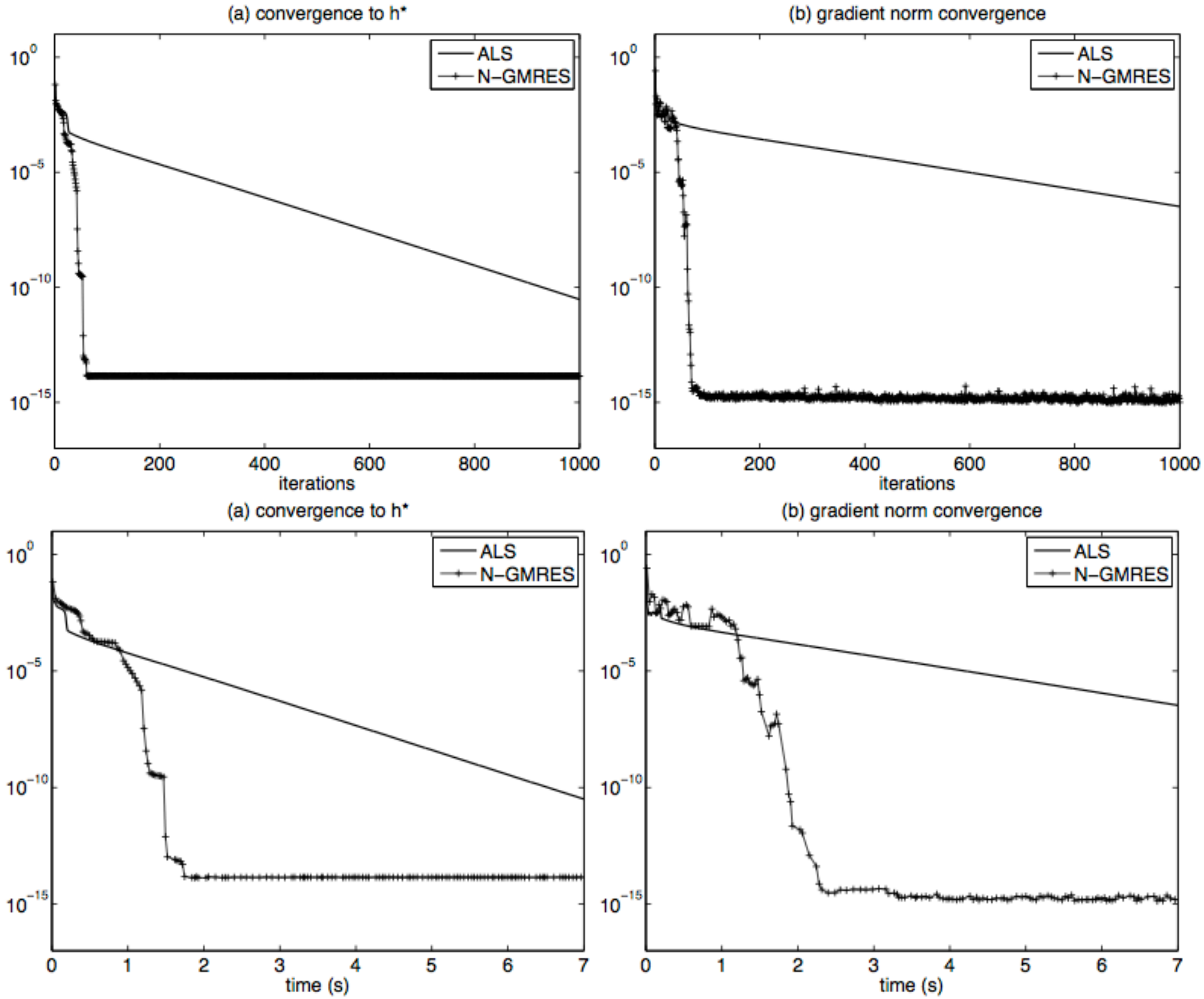
- dense test problem (from Tomasi and Bro; Acar et al.):
  - random rank-*R* tensor modified to obtain specific column collinearity, with added noise



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

UNIVERSITY OF WATERLOO

numerical results: dense test problem



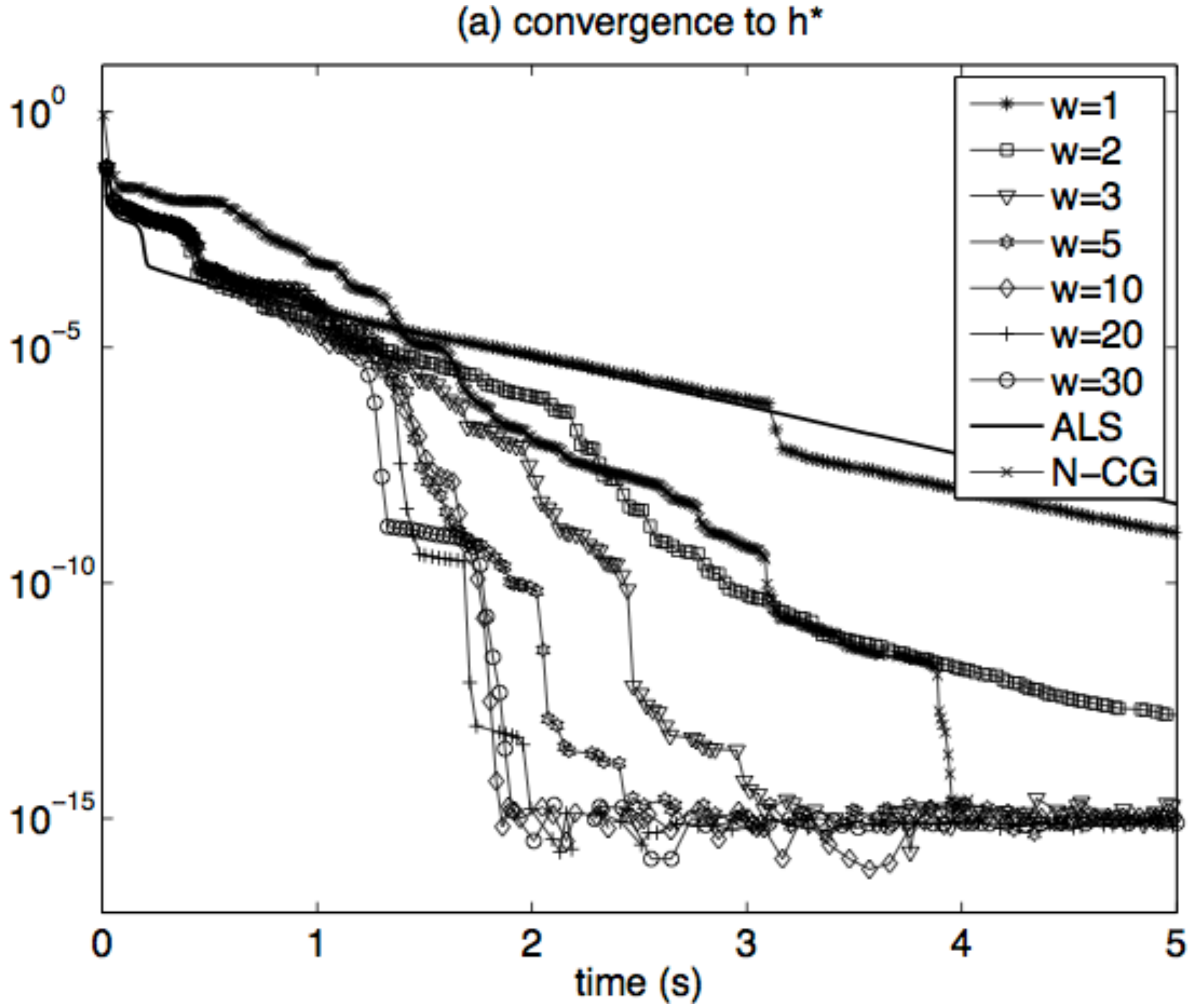
(a) convergence to h*

(b) gradient norm convergence

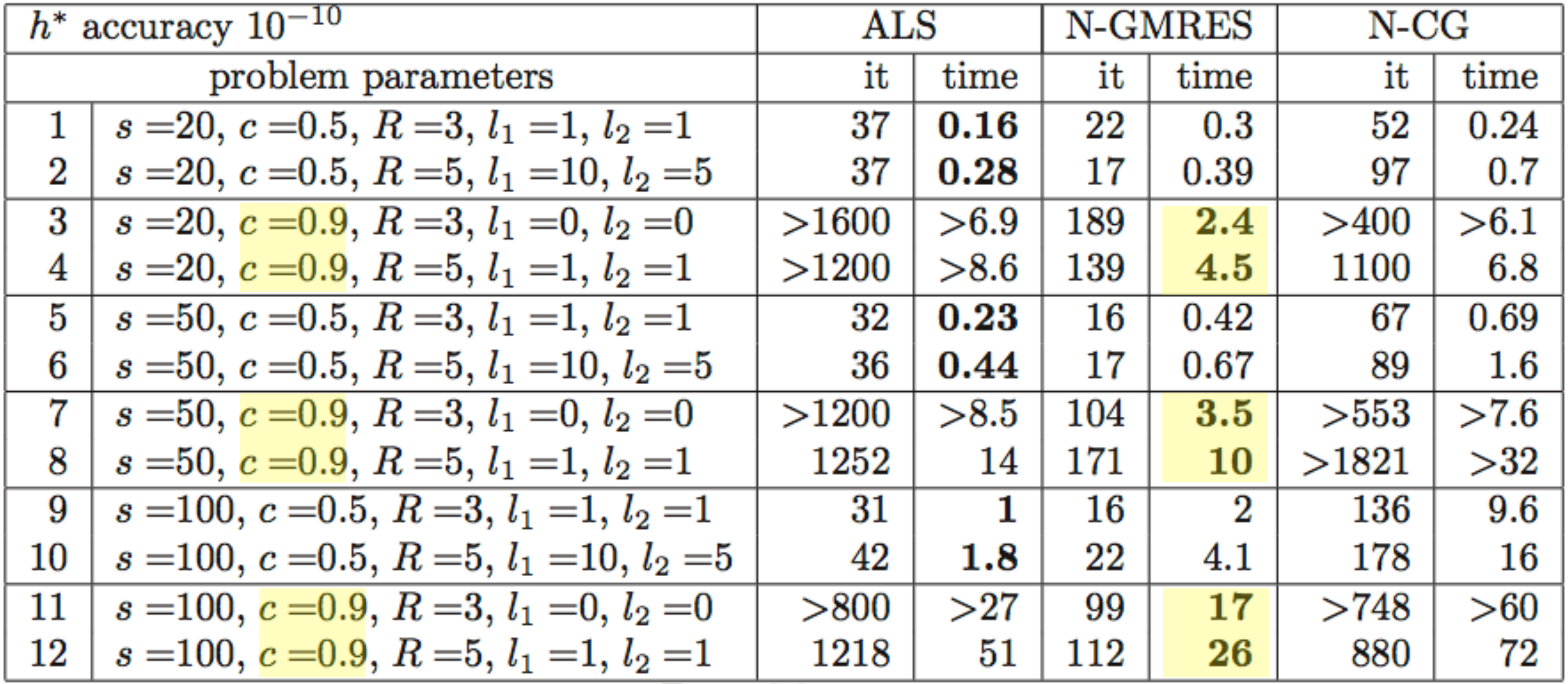
(a) convergence to h*

(b) gradient norm convergence

dense test problem: optimal window size



dense test problem: comparison



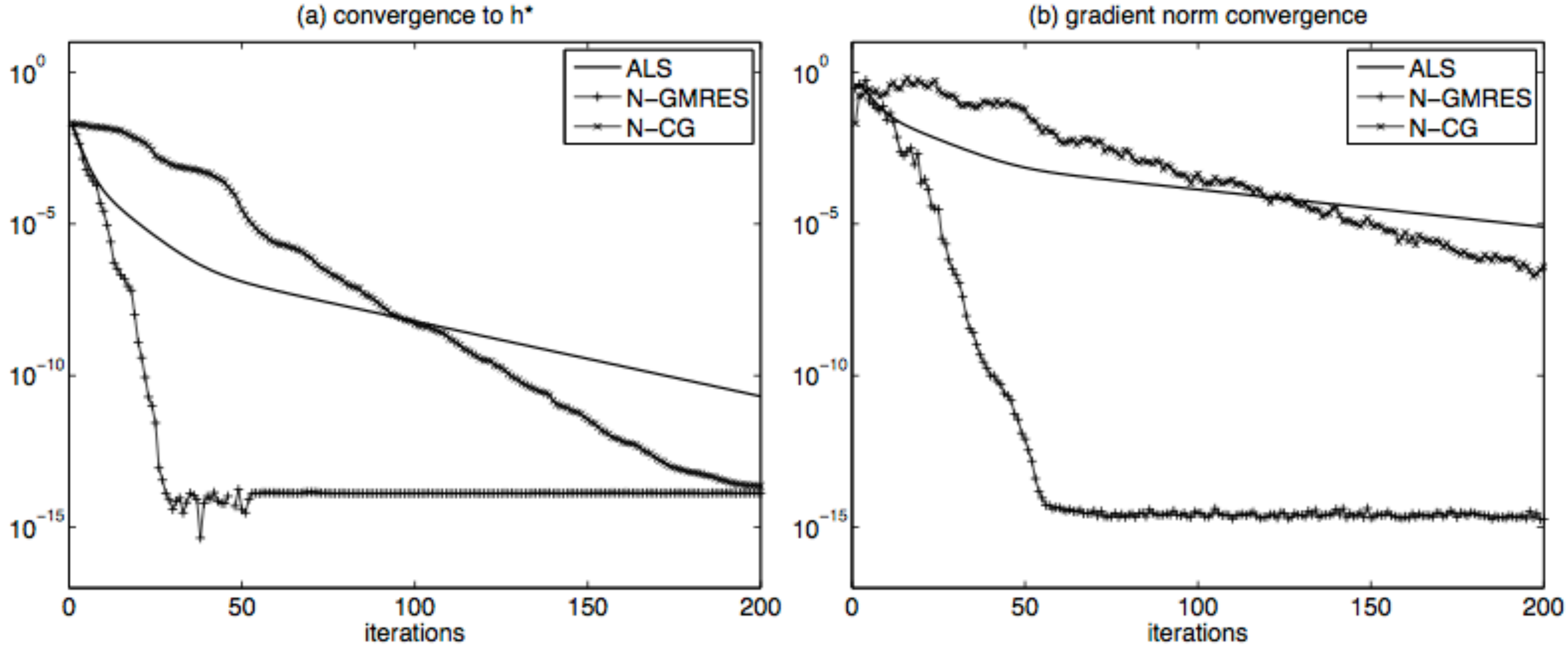
h^* accuracy 10^{-10}		ALS		N-GMRES		N-CG	
problem parameters		it	time	it	time	it	time
1	$s=20$, $c=0.5$, $R=3$, $l_1=1$, $l_2=1$	37	**0.16**	22	0.3	52	0.24
2	$s=20$, $c=0.5$, $R=5$, $l_1=10$, $l_2=5$	37	**0.28**	17	0.39	97	0.7
3	$s=20$, $c=0.9$, $R=3$, $l_1=0$, $l_2=0$	>1600	>6.9	189	**2.4**	>400	>6.1
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)

numerical results: sparse test problem

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



comparing N-GMRES to GMRES

(Washio and Oosterlee, ETNA, 1997)

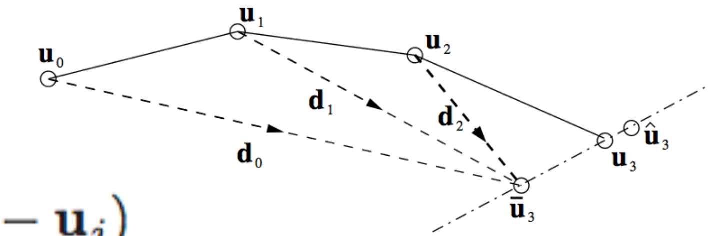
$$\mathbf{A}\,\mathbf{u} = \mathbf{b}$$

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

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

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

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



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

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

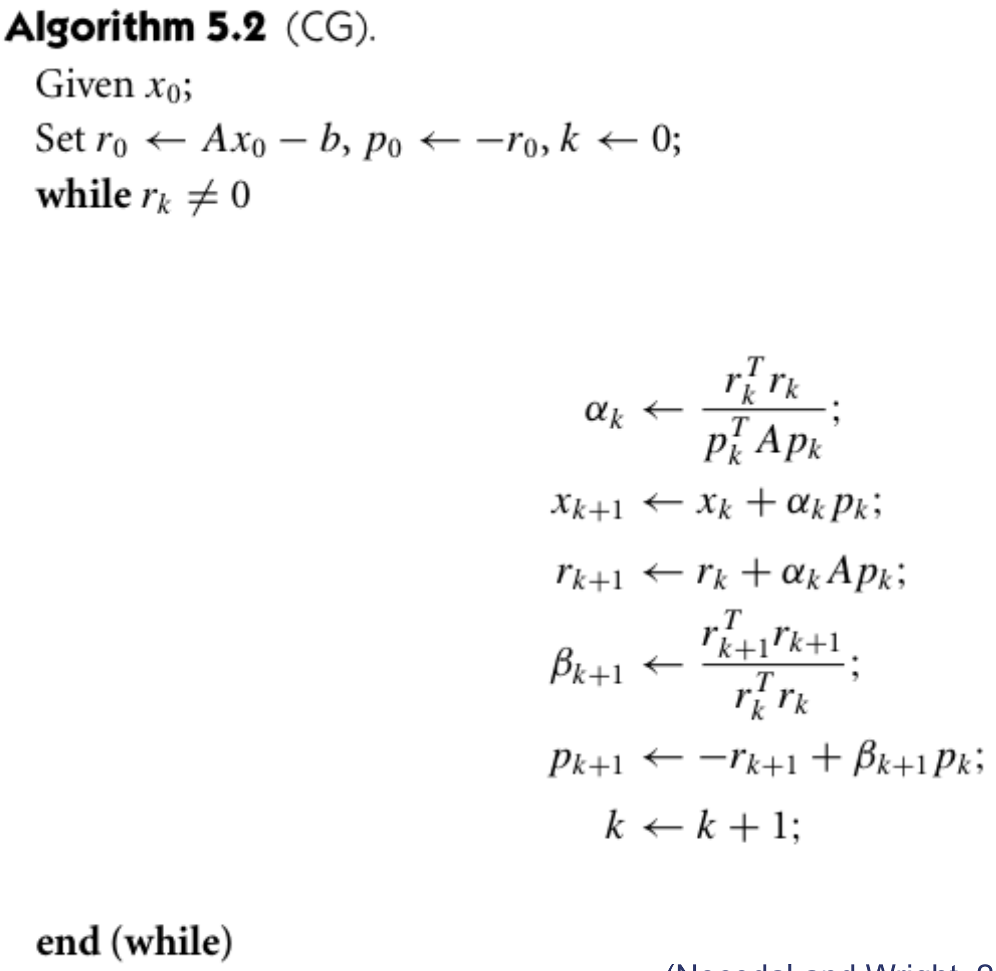
N-GMRES: what's in a name...

- CG for $\boxed{\mathbf{A}\,\mathbf{u} = \mathbf{b}}$

  ↓

  N-CG for $\boxed{\text{find } \mathbf{u}^* \text{ that minimizes } f(\mathbf{u})}$

conjugate gradient (CG) for $\mathbf{A}\,\mathbf{u} = \mathbf{b}$

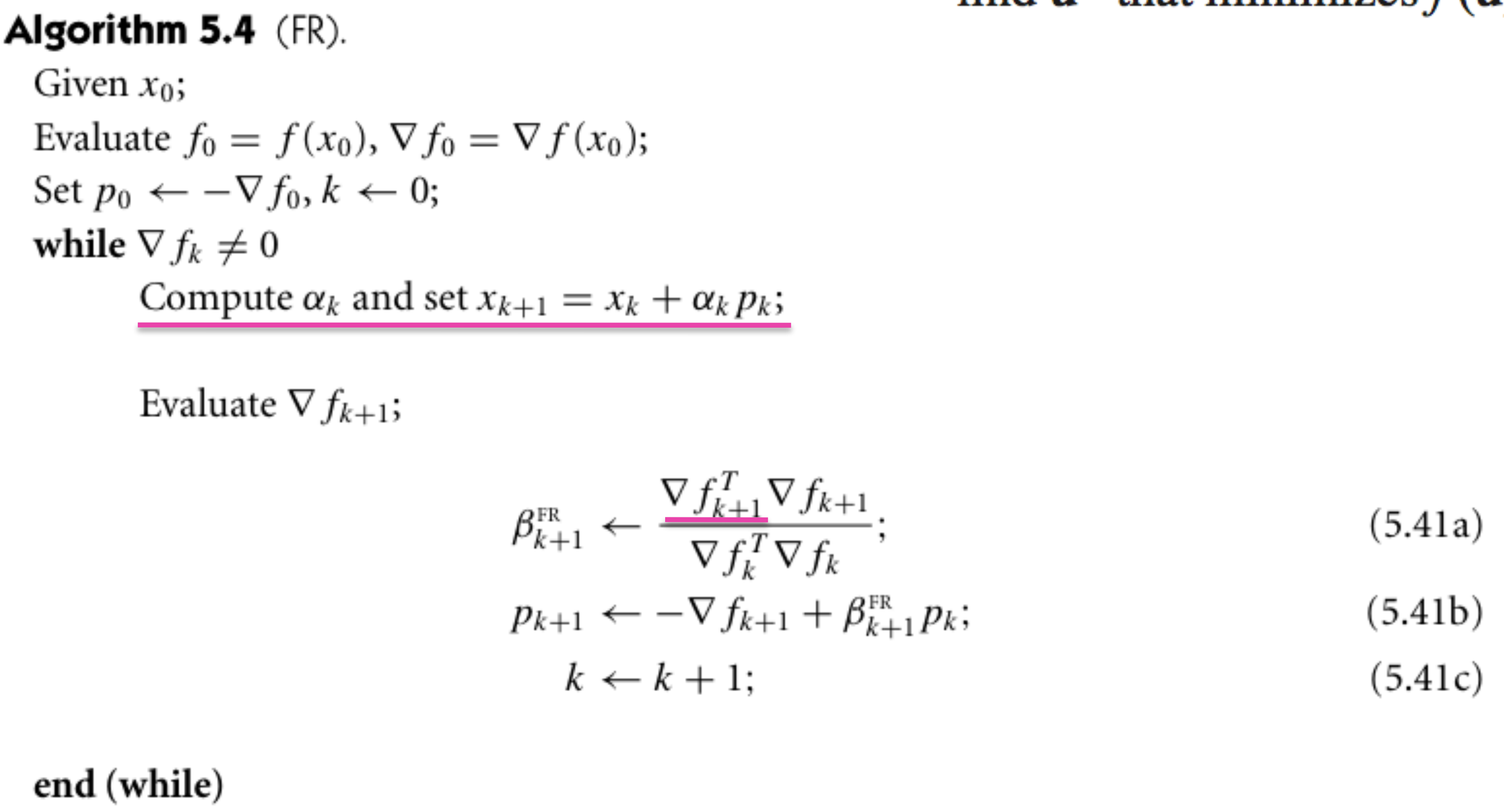


```
Algorithm 5.2 (CG).
  Given $x_0$;
  Set $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;
  while $r_k \neq 0$
      $\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k}$;
      $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
      $r_{k+1} \leftarrow r_k + \alpha_k A p_k$;
      $\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$;
      $p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k$;
      $k \leftarrow k + 1$;
  end (while)
```

(Nocedal and Wright, 2006)

nonlinear conjugate gradient (N-CG)

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

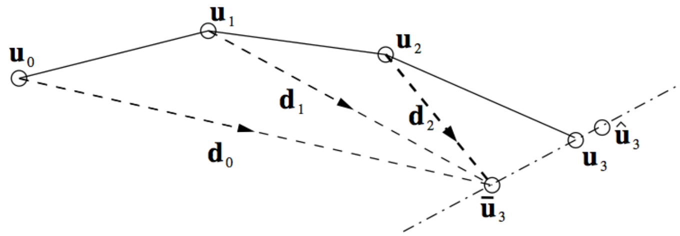


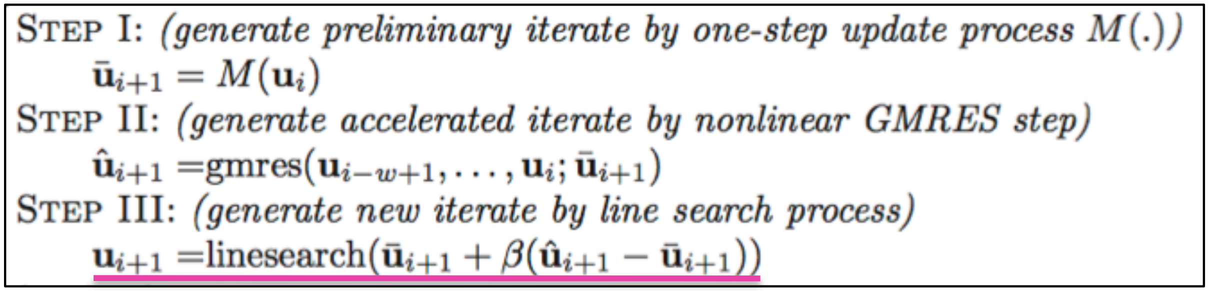
Algorithm 5.4 (FR).

```
Given $x_0$;
Evaluate $f_0 = f(x_0)$, $\nabla f_0 = \nabla f(x_0)$;
Set $p_0 \leftarrow -\nabla f_0$, $k \leftarrow 0$;
while $\nabla f_k \neq 0$
        $\underline{\text{Compute } \alpha_k \text{ and set } x_{k+1} = x_k + \alpha_k p_k;}$

        Evaluate $\nabla f_{k+1}$;

                $\beta_{k+1}^{\text{FR}} \leftarrow \frac{\nabla \underline{f_{k+1}^T} \nabla f_{k+1}}{\nabla f_k^T \nabla f_k};$        (5.41a)
                $p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{\text{FR}} p_k;$        (5.41b)
                $k \leftarrow k + 1;$        (5.41c)

end (while)
```

(Nocedal and Wright, 2006)

N-GMRES




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

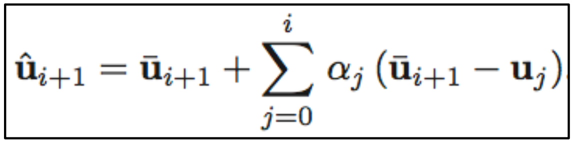
$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$

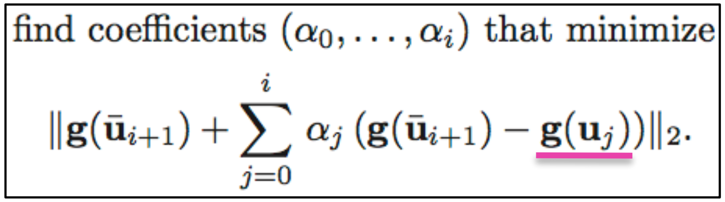
STEP II: *(generate accelerated iterate by nonlinear GMRES step)*

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

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

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


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


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

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

N-GMRES: what’s in a name...

- CG for $\boxed{\mathbf{A}\,\mathbf{u} = \mathbf{b}}$

↓

N-CG for $\boxed{\text{find } \mathbf{u}^* \text{ that minimizes } f(\mathbf{u})}$

- GMRES for $\boxed{\mathbf{A}\,\mathbf{u} = \mathbf{b}}$

↓

N-GMRES for $\boxed{\text{find } \mathbf{u}^* \text{ that minimizes } f(\mathbf{u})}$

(symmetry...)

5. N-GMRES as a general optimization method

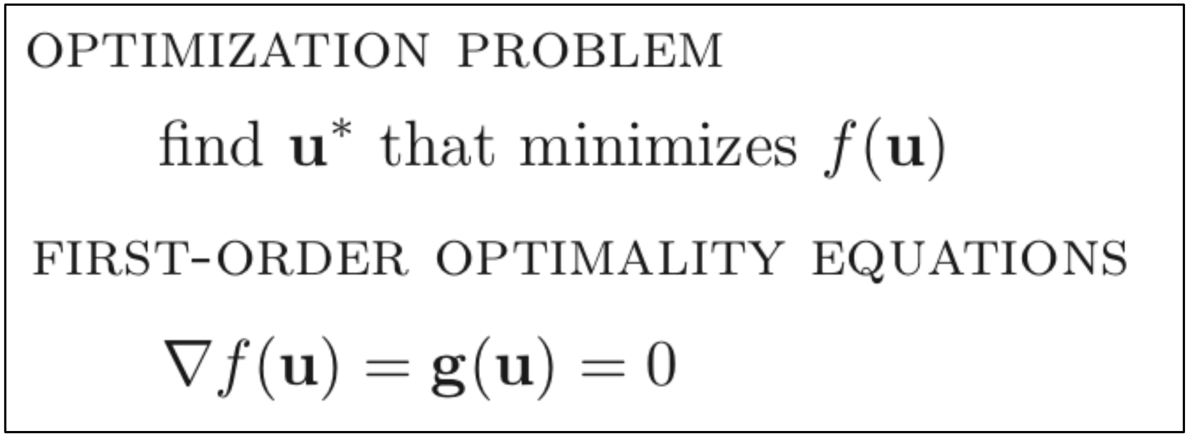
general methods for nonlinear optimization (smooth, unconstrained) ("Numerical Optimization", Nocedal and Wright, 2006)

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 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

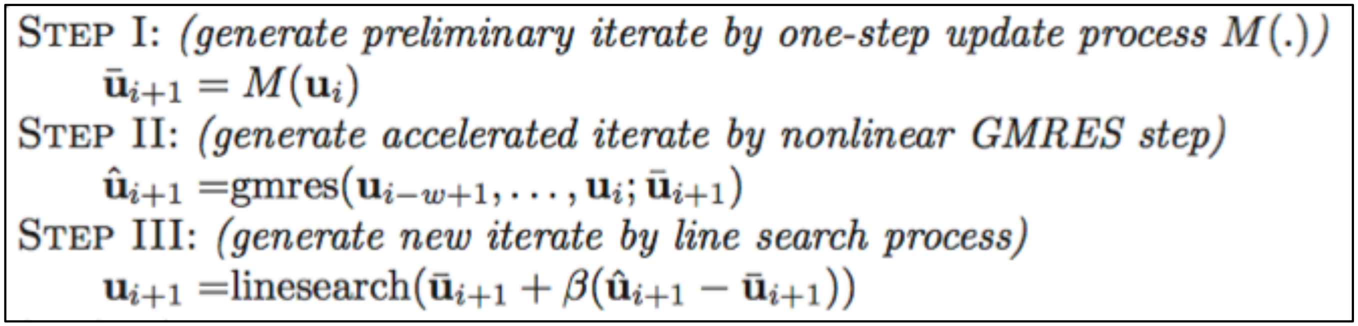
$$\nabla f(\mathbf{u}) = \mathbf{g}(\mathbf{u}) = 0$$

- 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)

steepest-descent preconditioning



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

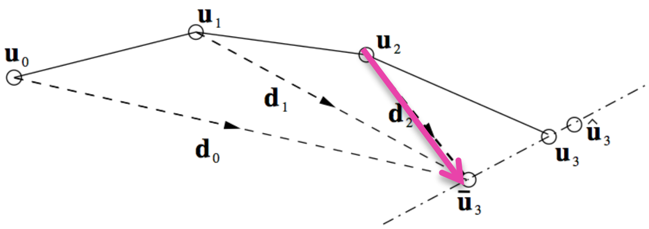
$$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$$

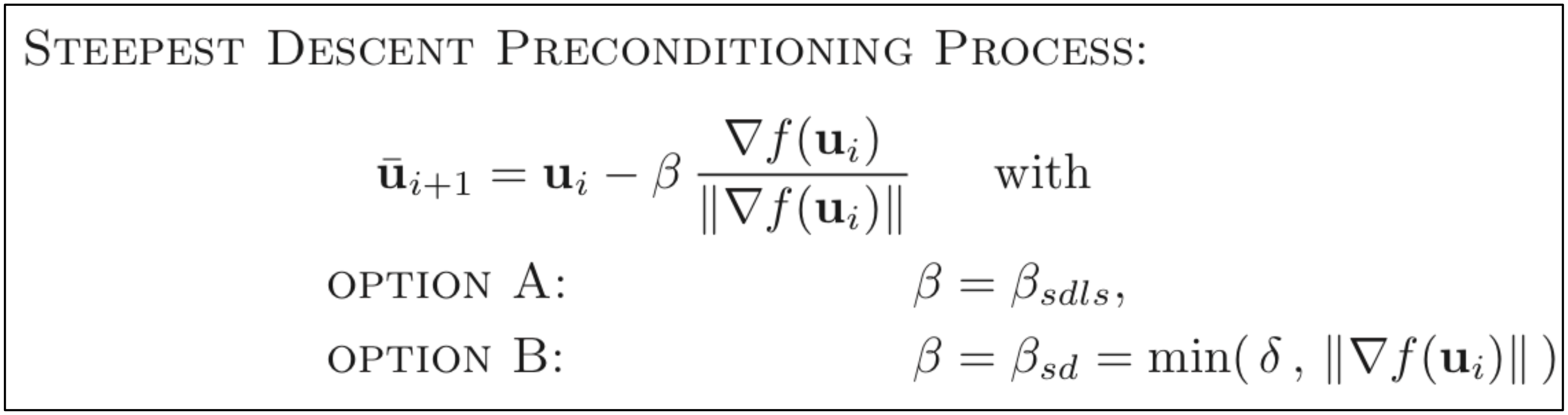
Step II: *(generate accelerated iterate by nonlinear GMRES step)*

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

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:

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

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 (note also: link with fixed-point equation)

UNIVERSITY OF WATERLOO

steepest-descent preconditioning

- claim: steepest descent is the 'natural' preconditioner for N-GMRES
- 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

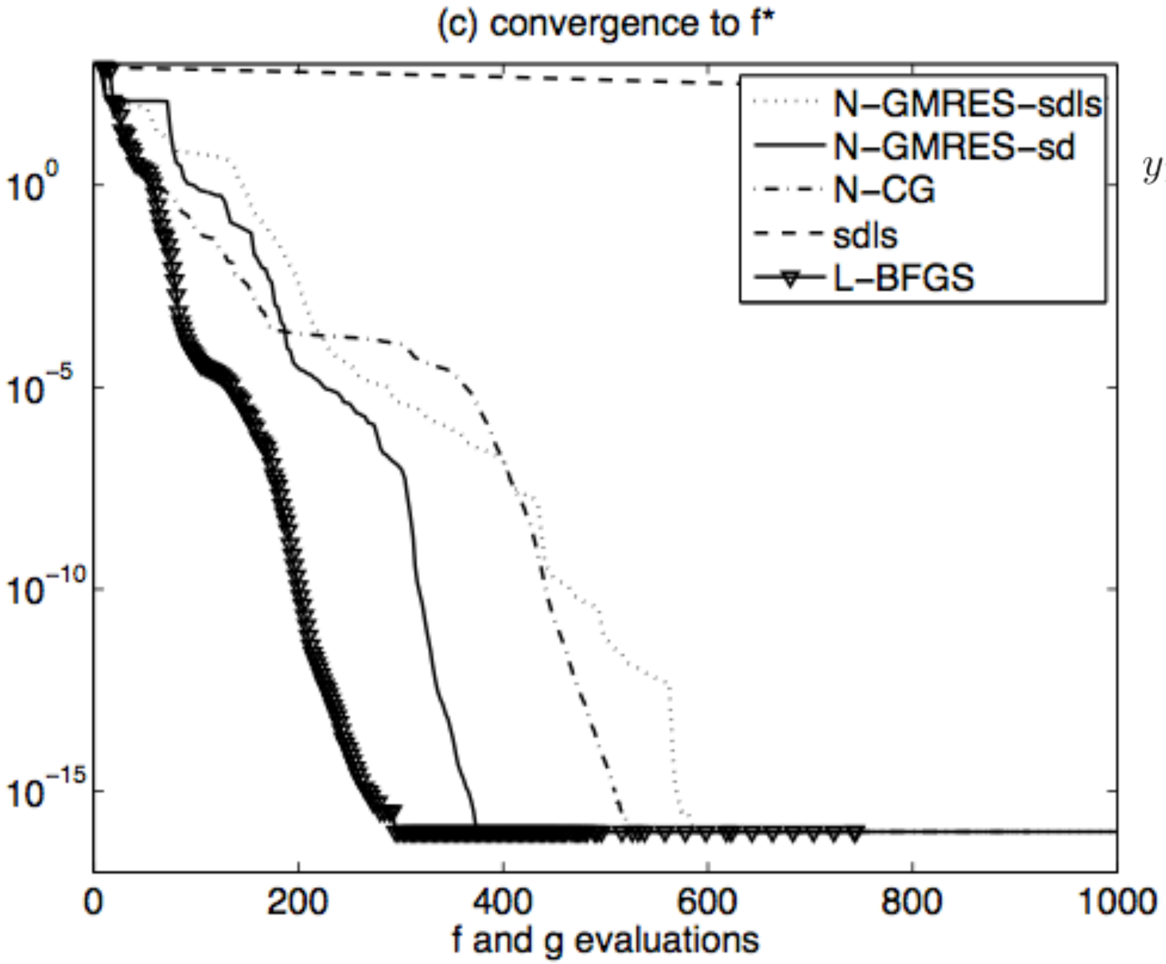
$$\bar{\mathbf{u}}_{i+1} = \mathbf{u}_i - \beta \frac{\nabla f(\mathbf{u}_i)}{\|\nabla f(\mathbf{u}_i)\|}$$ becomes $$\boxed{\bar{\mathbf{u}}_{i+1} = \mathbf{u}_i + \beta \frac{\mathbf{r}_i}{\|\mathbf{r}_i\|}}$$

- 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 N-GMRES preconditioner corresponds to identity preconditioner for linear GMRES

(and: small step is sufficient)

UNIVERSITY OF WATERLOO

numerical results: steepest-descent preconditioning



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

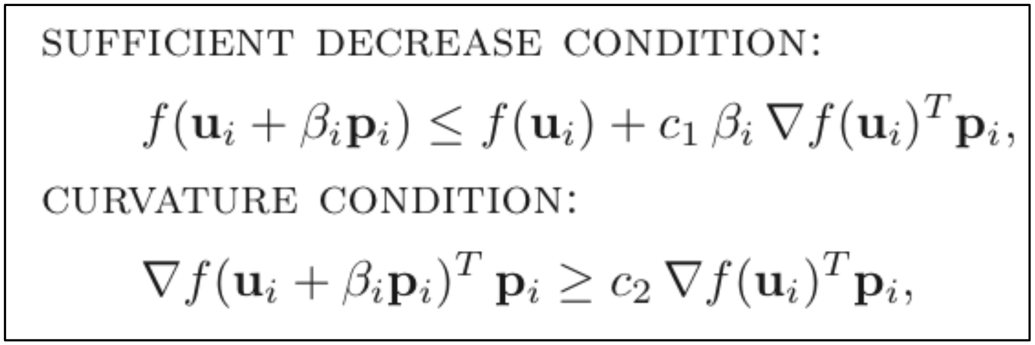
with $D = \mathrm{diag}(1, 2, \ldots, n)$ and $\mathbf{y}(\mathbf{x})$ given by

$$y_1(\mathbf{x}) = x_1 \text{ and } y_i(\mathbf{x}) = x_i - 10\,x_1^2 \ (i = 2, \ldots, n).$$

- 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)

convergence of steepest-descent preconditioned N-GMRES optimization

- assume line searches give solutions that satisfy Wolfe conditions:

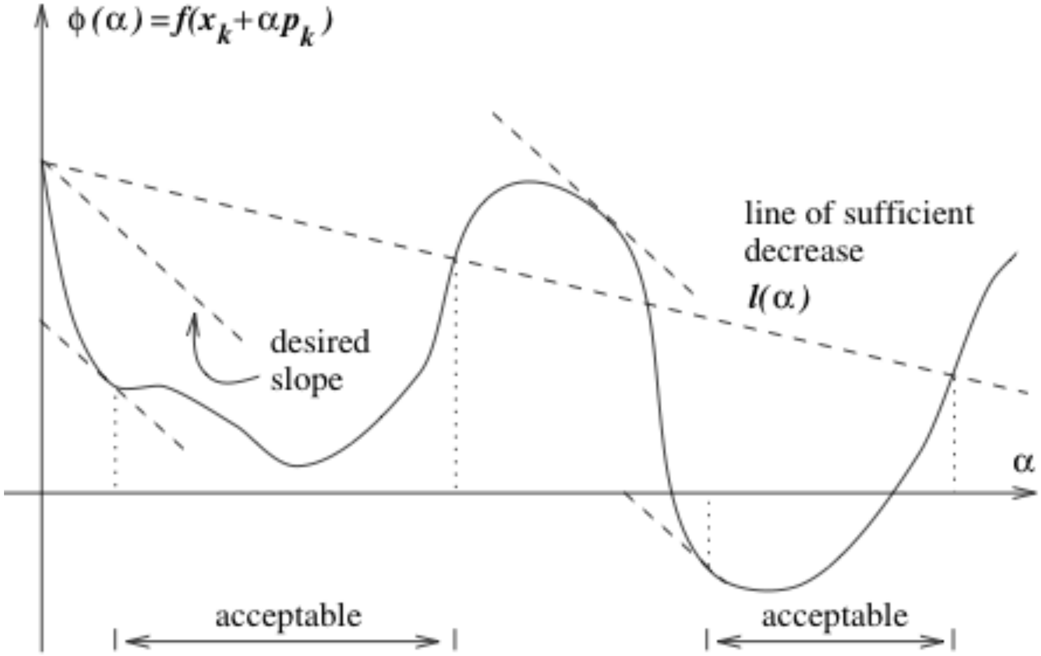


SUFFICIENT DECREASE CONDITION:

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

CURVATURE CONDITION:

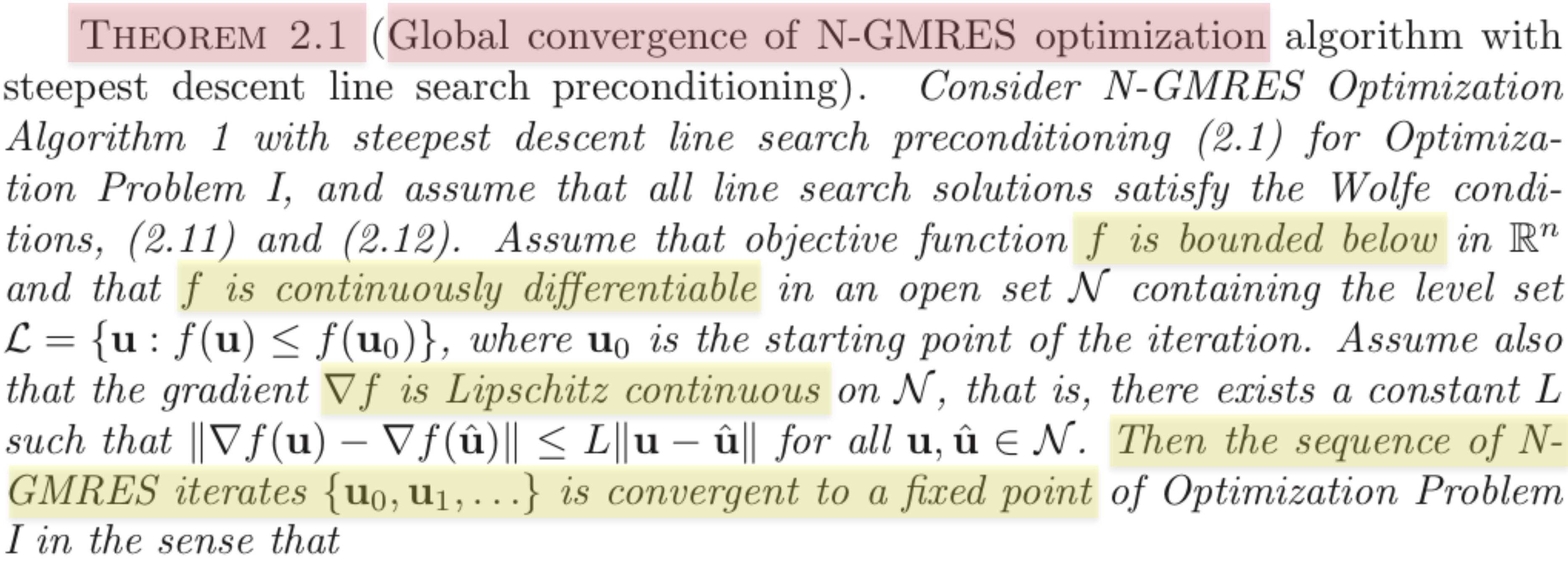
$$\nabla f(\mathbf{u}_i + \beta_i \mathbf{p}_i)^T\, \mathbf{p}_i \geq c_2\, \nabla f(\mathbf{u}_i)^T \mathbf{p}_i,$$



$\phi(\alpha) = f(x_k + \alpha p_k)$
line of sufficient decrease
$l(\alpha)$
desired slope
α
acceptable
acceptable

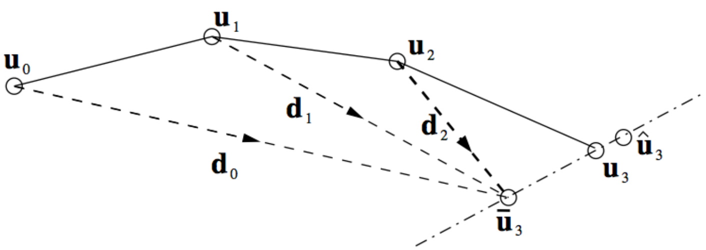
(Nocedal and Wright, 2006)

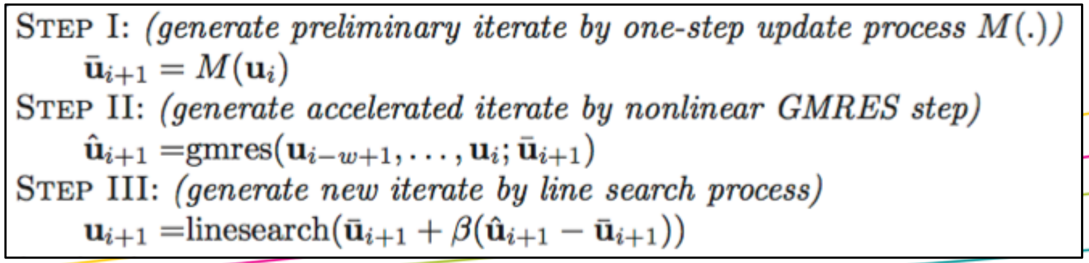
convergence of steepest-descent preconditioned N-GMRES optimization



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*

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





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)*

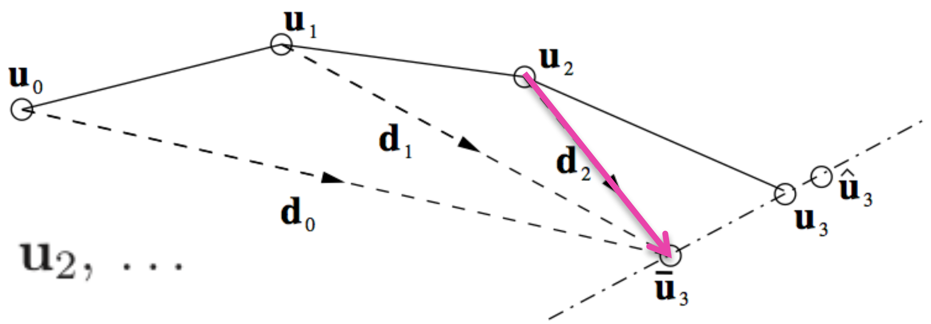
$\hat{\mathbf{u}}_{i+1} = \text{gmres}(\mathbf{u}_{i-w+1}, \ldots, \mathbf{u}_i; \bar{\mathbf{u}}_{i+1})$

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

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. preconditioned N-GMRES as a general optimization method

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

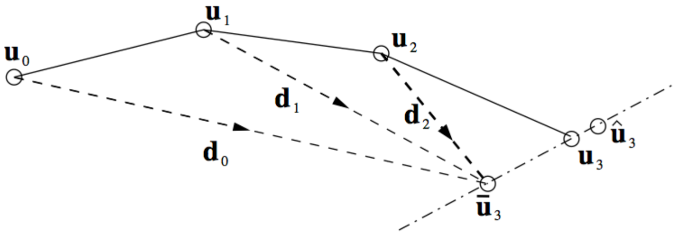
- acceleration step in N-GMRES is similar to Anderson acceleration and DIIS, but not exactly the same
- “mathematical equivalence” with GMRES in the linear case is discussed by Washio and Oosterlee (1997), Walker and Ni (2011), Rohwedder and Schneider (2011), and others
- equivalence of Anderson/DIIS 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

UNIVERSITY OF WATERLOO

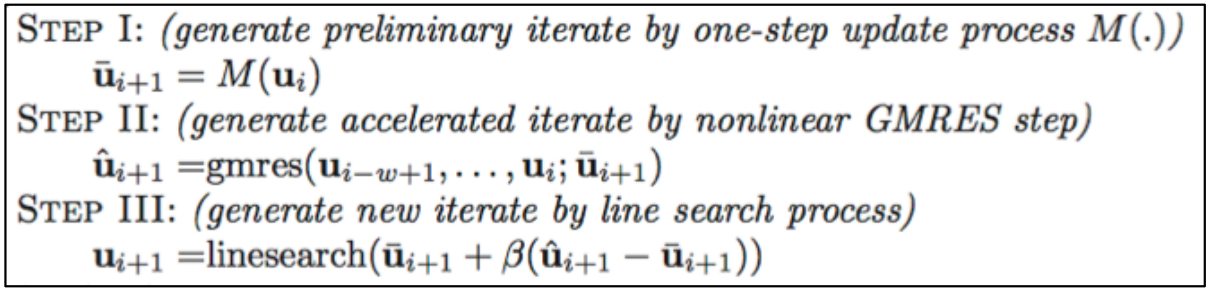
a few more notes on related methods for $\mathbf{g}(\mathbf{u}^*) = 0$

- as mentioned before, there are quite a few other papers that present related ideas:
  - 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.

6. conclusions



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



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

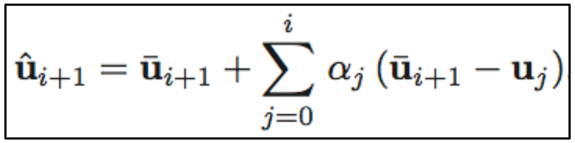
$$\bar{\mathbf{u}}_{i+1} = M(\mathbf{u}_i)$$

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

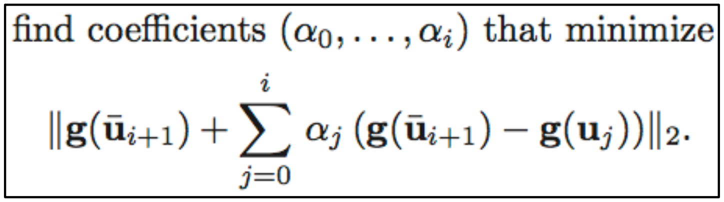
$$\hat{\mathbf{u}}_{i+1} = \text{gmres}(\mathbf{u}_{i-w+1}, \ldots, \mathbf{u}_i; \bar{\mathbf{u}}_{i+1})$$

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

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

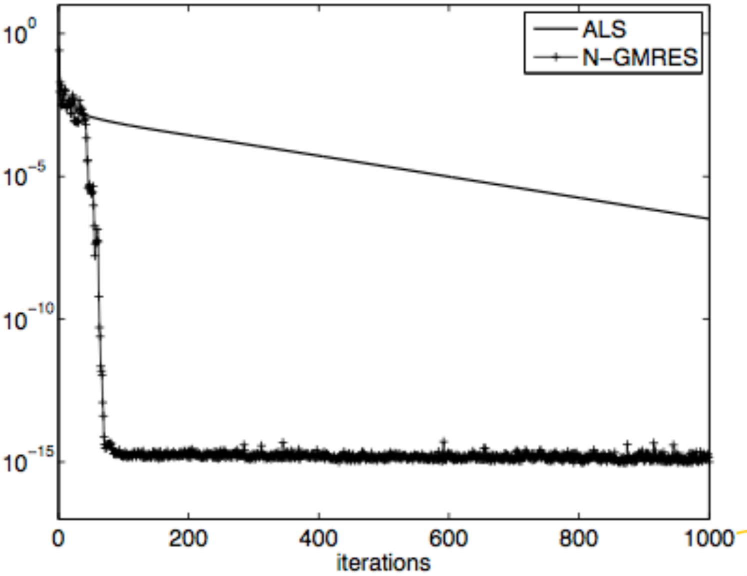


$$\hat{\mathbf{u}}_{i+1} = \bar{\mathbf{u}}_{i+1} + \sum_{j=0}^{i} \alpha_j \left(\bar{\mathbf{u}}_{i+1} - \mathbf{u}_j\right)$$



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

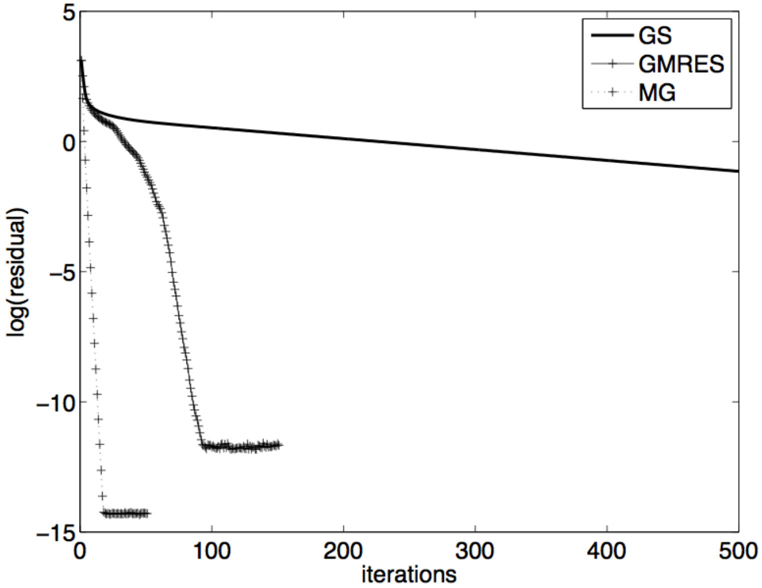
$$\|\mathbf{g}(\bar{\mathbf{u}}_{i+1}) + \sum_{j=0}^{i} \alpha_j \left(\mathbf{g}(\bar{\mathbf{u}}_{i+1}) - \mathbf{g}(\mathbf{u}_j)\right)\|_2.$$

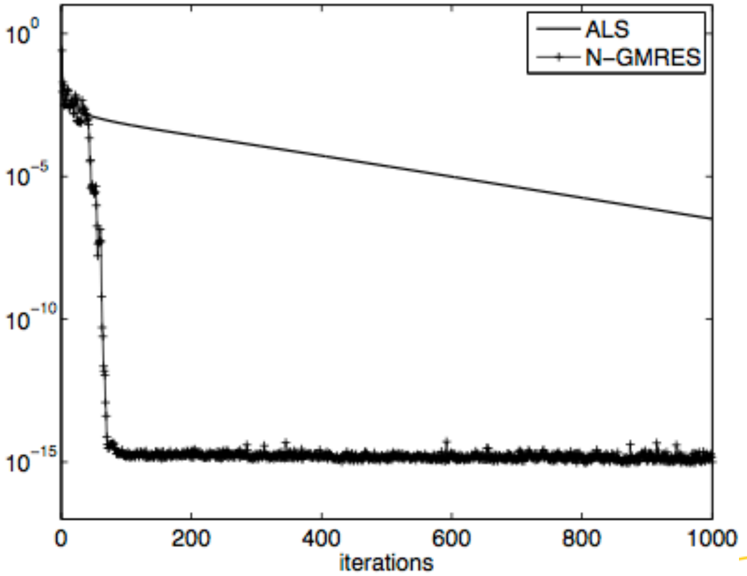


conclusions

- N-GMRES optimization method:

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





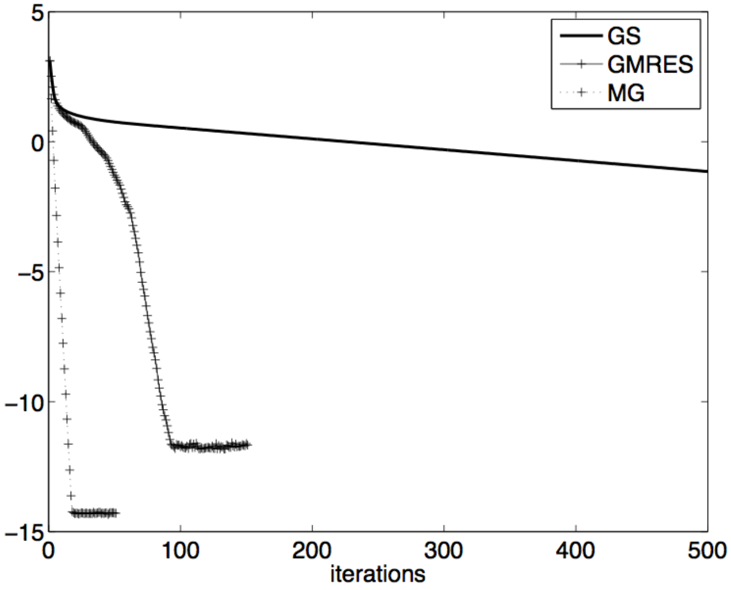
UNIVERSITY OF WATERLOO

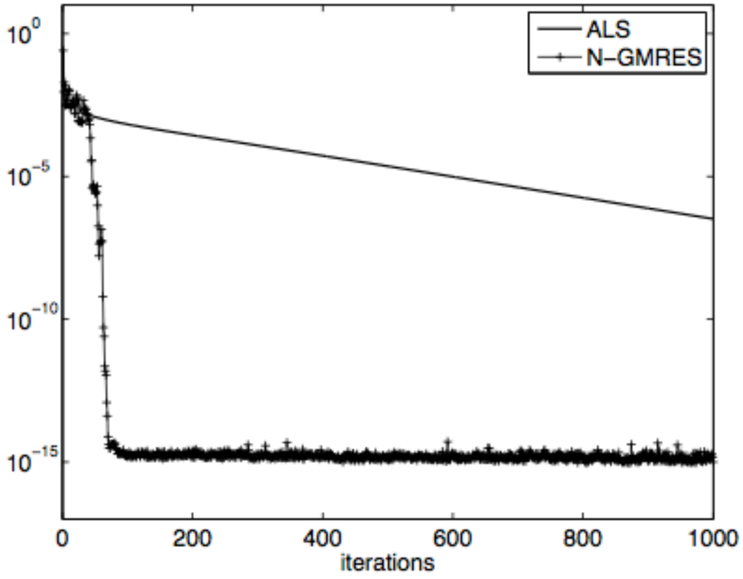
conclusions

- N-GMRES (with ALS preconditioning) gives strongly improved solver for canonical tensor decomposition
- note: we can also do nonlinear adaptive multigrid acceleration for the tensor nonlinear optimization problem!

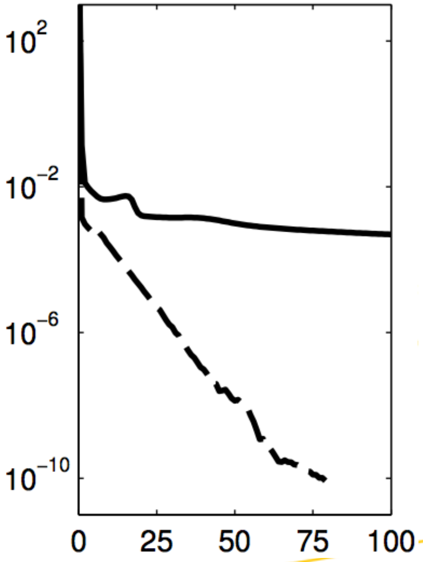
see MS 37. **Optimization methods for tensor decomposition,** Wednesday

12:15–12:40 “*An algebraic multigrid optimization method for low-rank canonical tensor decomposition*”, Killian Miller (ALS smoother)





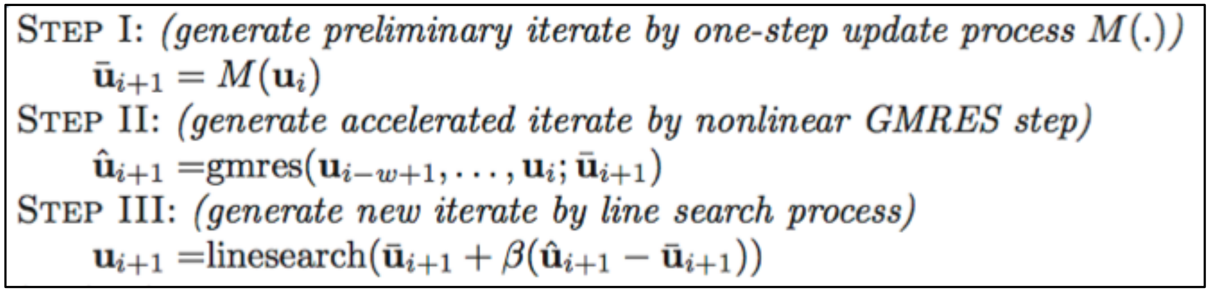
multigrid



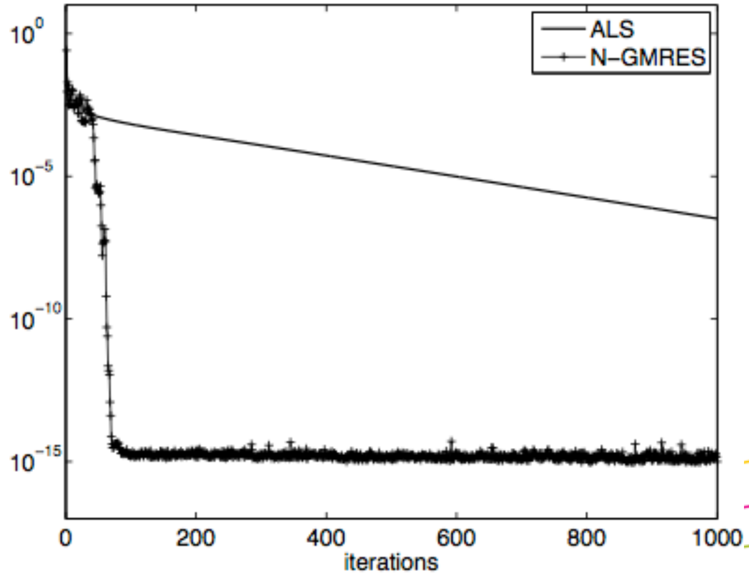
UNIVERSITY OF WATERLOO

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



open questions: convergence speed of N-GMRES

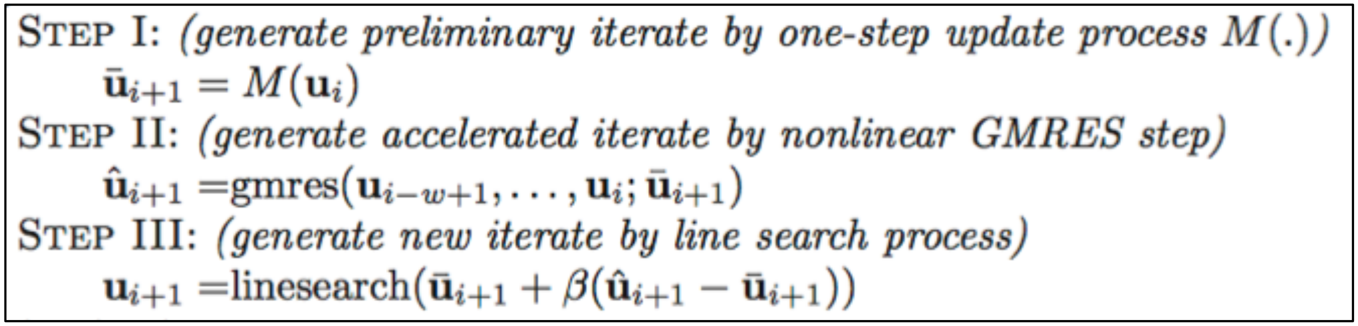
- GMRES (linear case): convergence rate can be analyzed by considering optimal polynomials etc.
- convergence speed of N-GMRES for optimization: many open questions

  (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

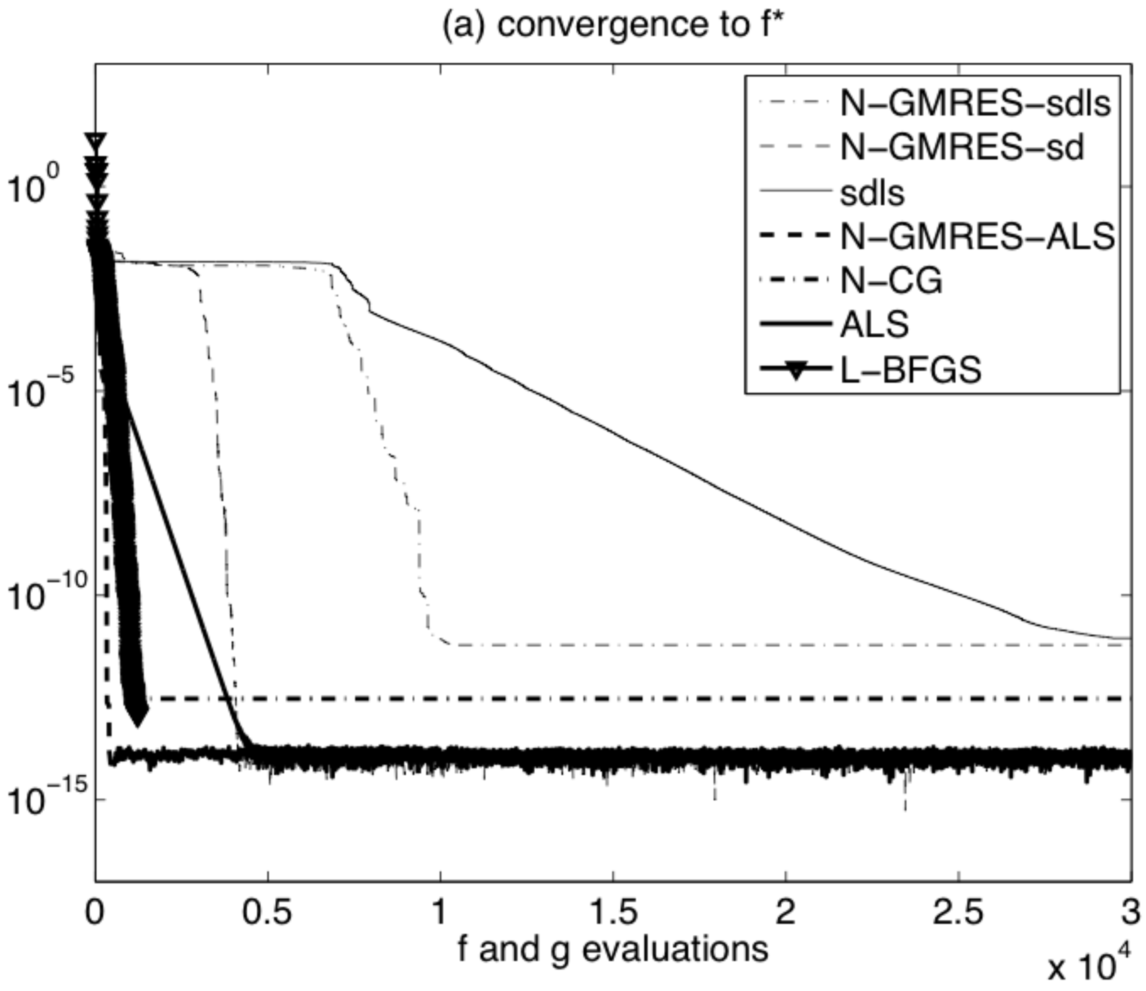
- real power of N-GMRES: N-GMRES optimization framework can employ sophisticated nonlinear preconditioners (use ALS in tensor case)
- power lies in good preconditioners (like case of GMRES for linear systems)



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

UNIVERSITY OF WATERLOO

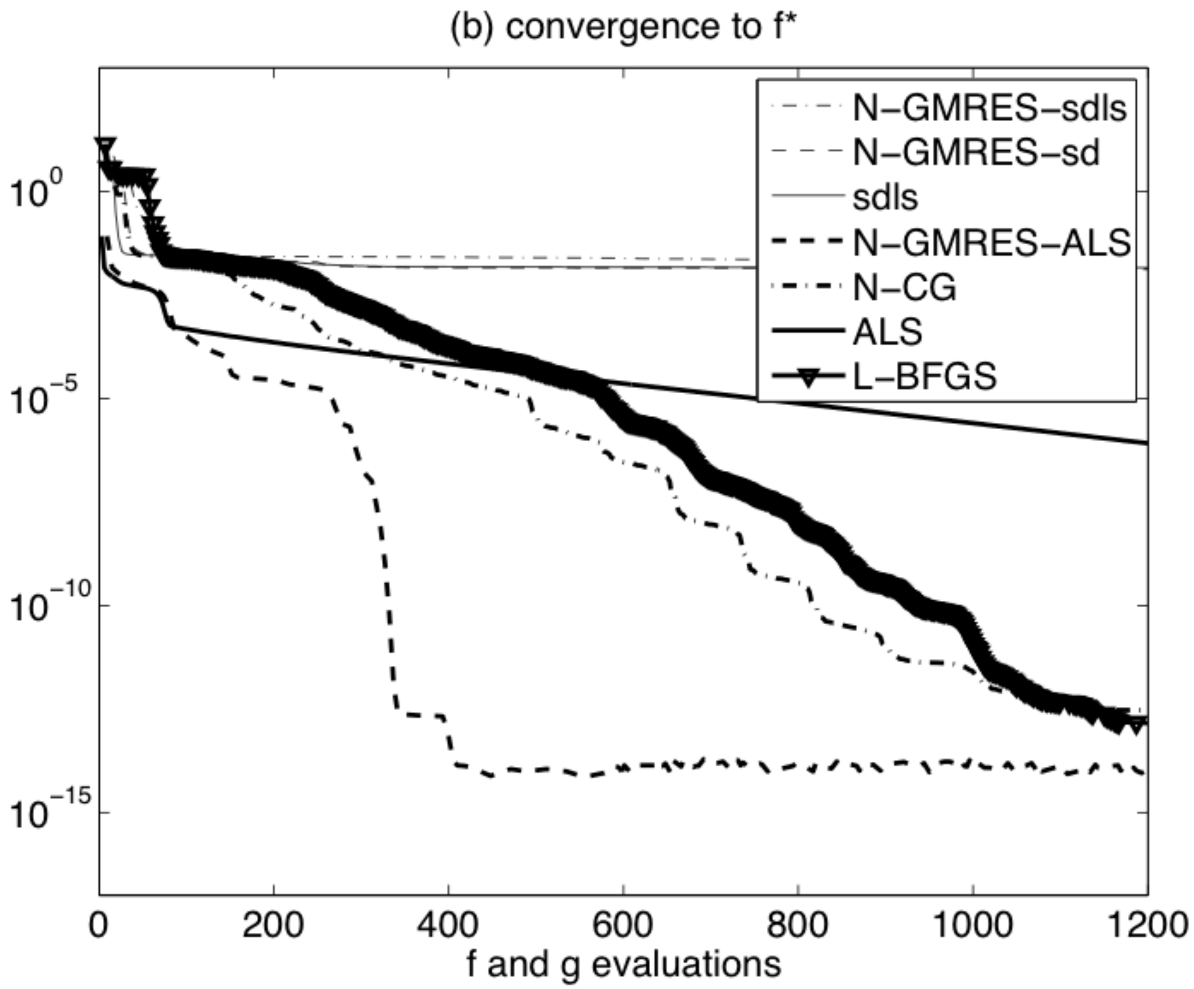
the power of N-GMRES optimization (tensor problem)



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!

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'*, submitted to SIAM J. Sci. Comp., 2011