Algorithm (6): WRFDA Minimization Algorithms

Jonathan (JJ) Guerrette
NCAR/MMM

WRFDA Tutorial, July 2019

Revisiting the nonlinear variational cost functions:

3DVAR

$$J = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x_b})^{\mathrm{T}}\mathbf{B}^{-1}(\boldsymbol{x} - \boldsymbol{x_b}) + \frac{1}{2}(H(\boldsymbol{x}) - \boldsymbol{y})^{\mathrm{T}}\mathbf{R}^{-1}(H(\boldsymbol{x}) - \boldsymbol{y})$$

$$\boldsymbol{x} \in \mathbb{R}^N$$
$$\boldsymbol{y} \in \mathbb{R}^M$$

4DVAR

$$J = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x_b})^{\mathrm{T}}\mathbf{B}^{-1}(\boldsymbol{x} - \boldsymbol{x_b}) + \frac{1}{2}\sum_k \left[M_k\left(H_k(\boldsymbol{x})\right) - \boldsymbol{y}_k\right]^T \mathbf{R}^{-1}\left[M_k\left(H_k(\boldsymbol{x})\right) - \boldsymbol{y}_k\right]$$

Note: $M_k(H_k(x))$ is a chain of functional relationships, which can be represented as a single function, $H_k(x)$. This simplification and the 4DVAR summation collapsing to a single term will be used from this point forward for simplicity.

Iterative nonlinear minimization techniques

- Gradient or Steepest Descent
  - Nonlinear J
  - Step in opposite direction of the gradient: $\delta \boldsymbol{x} = -\gamma \cdot \boldsymbol{\nabla} J$
  - Perform line search to determine scalar γ
  - Slow convergence, but easy to formulate

- Truncated Gauss-Newton (TGN)
  - Minimize a sequence of *quadratic approximations* of J
  - Efficient for weakly nonlinear problems, but higher complexity
  - Used at most operational NWP centers and in WRFDA

- Quasi-Newton (e.g., BFGS or L-BFGS)
  - Nonlinear J
  - Use $\boldsymbol{\nabla} J$ to approximate the Hessian (second derivative of J) to speed up convergence
  - Effective for highly nonlinear problems OR when quadratic approximation of J is unavailable

What is a quadratic approximation of J?

Full Nonlinear J (same as slide 2)

$$J = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x_b})^{\mathrm{T}}\mathbf{B}^{-1}(\boldsymbol{x} - \boldsymbol{x_b}) + \frac{1}{2}(H(\boldsymbol{x}) - \boldsymbol{y})^{\mathrm{T}}\mathbf{R}^{-1}(H(\boldsymbol{x}) - \boldsymbol{y})$$

$$\boldsymbol{x} \in \mathbb{R}^N$$
$$\boldsymbol{y} \in \mathbb{R}^M$$

Preconditioned Quadratic Approximation

$$\tilde{J}_i = \frac{1}{2}\left(\sum_{j=1}^{i} \delta\boldsymbol{v}^{\mathrm{j}}\right)^{\mathrm{T}}\left(\sum_{j=1}^{i} \delta\boldsymbol{v}^{j}\right) + \frac{1}{2}\left(\mathbf{HL}\delta\boldsymbol{v}^{i} - \boldsymbol{d}_i\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{HL}\delta\boldsymbol{v}^{i} - \boldsymbol{d}_i\right)$$

"Incremental Variational DA":
Minimization is cast in terms of an increment
while the quantity of interest is held constant

$$\boldsymbol{d}_{\mathrm{i}} = \boldsymbol{y} - H(\boldsymbol{x}_{i-1})$$

$$\mathbf{B} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$$

$$\mathrm{i} \equiv [\text{outer loop iteration}]$$

Derived by approximating $H(\boldsymbol{x} + \delta\boldsymbol{x}) \cong H(\boldsymbol{x}) + \mathbf{H}\delta\boldsymbol{x}$
$\tilde{J}_i$ is quadratic in terms of the increment, $\delta\boldsymbol{v}^{\mathrm{i}}$
$\tilde{J}_i$ circumvents nonlinear functionals to enable alternative solution methods

Minimize $\tilde{J}_i$ OR find where $(\boldsymbol{\nabla}\tilde{J}_i = \mathbf{0})$

Zero Gradient

$$\boldsymbol{\nabla}_{\delta \boldsymbol{v}^i}\tilde{J}_i = \mathbf{0} = \sum_{j=1}^{i-1} \delta \boldsymbol{v}^j + \delta \boldsymbol{v}^{\mathrm{i}} + \mathbf{L}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{HL}\delta \boldsymbol{v}^i - \boldsymbol{d}_i\right)$$

Minimize $\tilde{J}_i$ means find where $\boldsymbol{\nabla \tilde{J}_i = 0}$

Zero Gradient

$$\boldsymbol{\nabla}_{\delta v^i} \tilde{J}_i = \mathbf{0} = \sum_{j=1}^{i-1} \delta \boldsymbol{v}^j + \delta \boldsymbol{v}^{\mathrm{i}} + \mathbf{L}^{\mathrm{T}} \mathbf{H}^{\mathrm{T}} \mathbf{R}^{-1} \left(\mathbf{HL} \delta \boldsymbol{v}^i - \boldsymbol{d}_i \right)$$

Solve for $\delta \boldsymbol{v}^{\mathrm{i}}$

$$\left(\mathbf{I} + \mathbf{L}^{\mathrm{T}} \mathbf{H}^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{HL} \right) \delta \boldsymbol{v}^i = - \sum_{j=1}^{i-1} \delta \boldsymbol{v}^j + \mathbf{L}^{\mathrm{T}} \mathbf{H}^{\mathrm{T}} \mathbf{R}^{-1} \boldsymbol{d}_i$$

$$\delta \boldsymbol{v}^{\mathrm{i}} = - \left(\mathbf{I} + \mathbf{L}^{\mathrm{T}} \mathbf{H}^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{HL} \right)^{-1} \left(\sum_{j=1}^{i-1} \delta \boldsymbol{v}^j + \mathbf{L}^{\mathrm{T}} \mathbf{H}^{\mathrm{T}} \mathbf{R}^{-1} \boldsymbol{d}_i \right)$$

where $\nabla^2 \tilde{J}_i = \left(\mathbf{I} + \mathbf{L}^{\mathrm{T}} \mathbf{H}^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{HL} \right)$ is the Hessian of $\tilde{J}_i$

Minimize $\tilde{J}_i$ means find where $\nabla \tilde{J}_i = \mathbf{0}$

Zero Gradient

$$\nabla_{\delta v^i} \tilde{J}_i = \mathbf{0} = \sum_{j=1}^{i-1} \delta v^j + \delta v^{\mathrm{i}} + \mathbf{L}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{HL}\delta v^i - d_i\right)$$

Solve for δv^{i}

$$\left(\mathbf{I} + \mathbf{L}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{HL}\right)\delta v^i = -\sum_{j=1}^{i-1} \delta v^j + \mathbf{L}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1} d_i$$

$$\delta v^{\mathrm{i}} = -\left(\mathbf{I} + \mathbf{L}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{HL}\right)^{-1}\left(\sum_{j=1}^{i-1} \delta v^j + \mathbf{L}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1} d_i\right)$$

Side note: gradient descent increment is $\delta v^{\mathrm{i}} = -\gamma \cdot \left(\sum_{j=1}^{i-1} \delta v^j + \mathbf{L}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1} d_i\right)$

Minimize $\tilde{J}_i$ OR find where $(\nabla \tilde{J}_i = \mathbf{0})$

Gradient

$$\nabla_{\delta v^i} \tilde{J}_i = \mathbf{0} = \sum_{j=1}^{i-1} \delta v^j + \delta v^i + \mathbf{L}^T \mathbf{H}^T \mathbf{R}^{-1} \left(\mathbf{H} \mathbf{L} \delta v^i - \boldsymbol{d}_i \right)$$

Solve for δv^i

$$\underbrace{\left(\mathbf{I} + \mathbf{L}^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{L} \right)}_{\mathbf{A}} \delta v^i = \underbrace{-\sum_{j=1}^{i-1} \delta v^j + \mathbf{L}^T \mathbf{H}^T \mathbf{R}^{-1} \boldsymbol{d}_i}_{\boldsymbol{b}}$$

Finally, we see that the quadratic minimization is a <u>linear algebra problem</u>

$$\mathbf{A} \hat{\boldsymbol{x}} = \boldsymbol{b} \quad \text{where} \quad \mathbf{A} \equiv \left(\mathbf{I} + \mathbf{L}^T \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \mathbf{L} \right) = \nabla^2 \tilde{J}_i$$

$$\hat{\boldsymbol{x}} \equiv \delta v^i$$

$$\boldsymbol{b} \equiv -\sum_{j=1}^{i-1} \delta v^j + \mathbf{L}^T \mathbf{H}^T \mathbf{R}^{-1} \boldsymbol{d}_i = -\nabla \tilde{J}_i \Big|_{\hat{\boldsymbol{x}}=\mathbf{0}}$$

Linear Algebra Solvers ($\mathbf{A}\hat{\boldsymbol{x}} = \boldsymbol{b}$)

Commonly used methods for *explicit* systems:

- Gaussian Elimination
- Cramer's Rule
- Gauss-Seidel
- LU Decomposition
- Singular Value Decomposition (SVD)

All of these require an *explicit* representation of $\mathbf{A}$, and are computationally prohibitive in high dimensions

The Hessian for NWP problems is *implicit* (formed through numerical operations), symmetric, and can have dimensionality of $N \sim 10^6$ to 10^8

Krylov Subspaces (used in WRFDA)

- Examples: Conjugate Gradient (CG) or Lanczos Recurrance (Lanczos is mathematically equivalent to CG in infinite precision)
  e.g., Golub and Van Loan, Matrix Computations, 3rd ed. (1996) or 4th ed. (2013)

- Iterative; hence the phrase “inner-loop”, in addition to the “outer-loop” comprising the TGN minimization

- Designed specifically to work with *implicit* and symmetric $\mathbf{A}$

- Each inner iteration: derive an update to $\delta \boldsymbol{v}^i$ by multiplying $\mathbf{A}$ (Hessian of $\tilde{J}_i$) by a vector related to $\boldsymbol{b}$ (gradient of $\tilde{J}_i$)

Krylov Subspaces; how do they work?

The $\text{rank} - l$ Krylov subspace is formed by $l - 1$ multiplications of $\mathbf{A}$ by $\boldsymbol{b}$

$$\mathcal{K}_l(\mathbf{A}, \boldsymbol{b}) \cong \text{span}\{\boldsymbol{b}, \mathbf{A}\boldsymbol{b}, \mathbf{A}^2\boldsymbol{b}, \dots, \mathbf{A}^{l-1}\boldsymbol{b}\}$$
$$\cong \text{span}\{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_l\}$$

where $\boldsymbol{p}_i \mathbf{A} \boldsymbol{p}_j = \mathbf{0}$ for $i \neq j$

Each polynomial term is linearly independent or conjugate to all others with respect to inner-product with $\mathbf{A}$

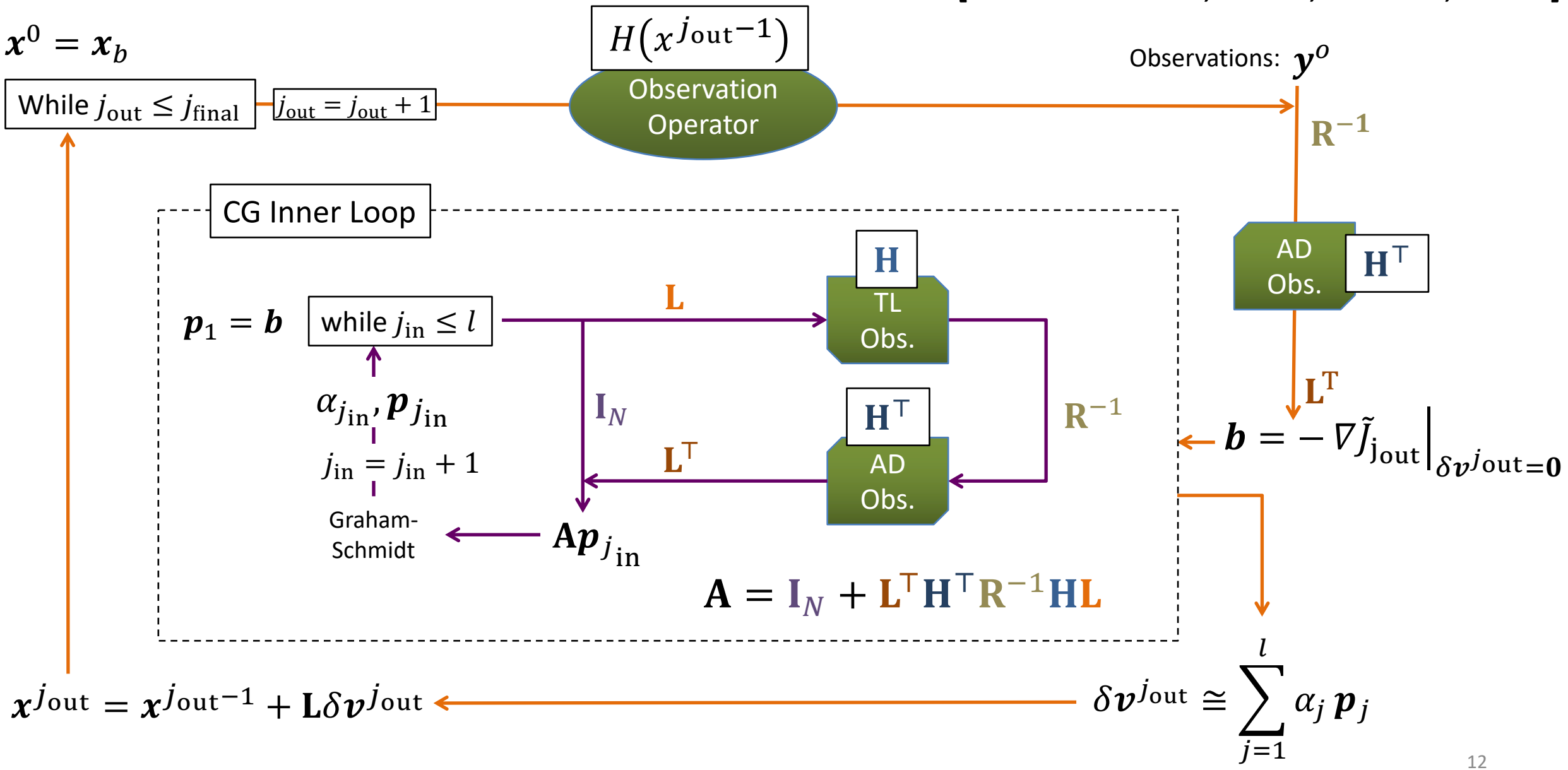
Thus $\mathbf{Q}_l$ is a low-rank basis for $\mathbf{A}$ and a solution to $\mathbf{A}\hat{\boldsymbol{x}} = \boldsymbol{b}$ can be expressed by a linear combination of $\boldsymbol{p}_j$'s:

$$\hat{\boldsymbol{x}}_* \cong \sum_{j=1}^{l} \alpha_j \boldsymbol{p}_j$$

- For CG, α_j is found by minimizing the residual error norm of the cost function gradient,
$$\|\boldsymbol{r}_j\| = \|\boldsymbol{b} - \mathbf{A}\hat{\boldsymbol{x}}_j\|$$
e.g., Matrix Computations by Golub and Van Loan.
- $\boldsymbol{r}_j$ and α_j depend on all previous iterations

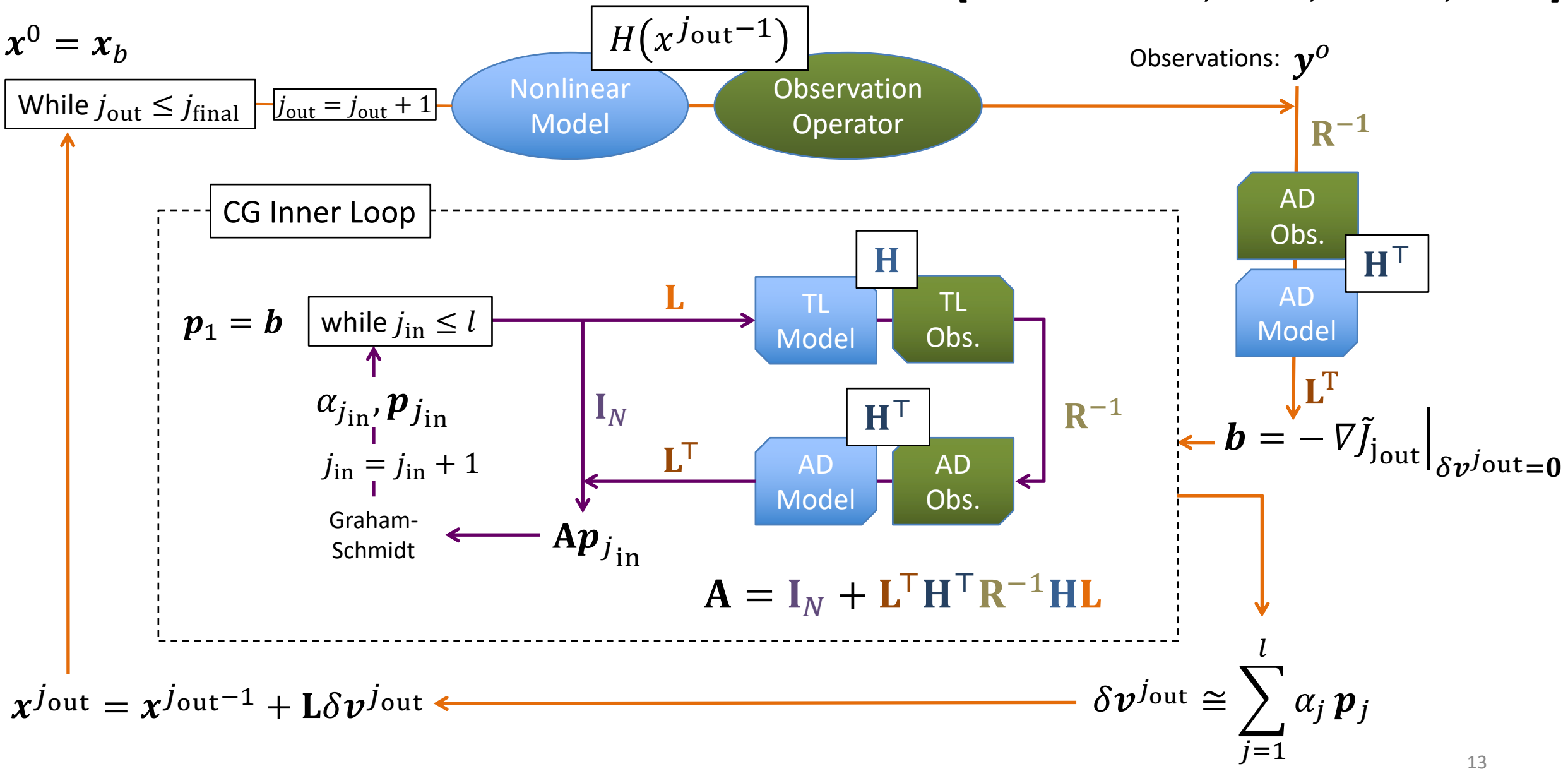
3D-Var TGN w/ CG in NWP

[Courtier et al., 1994; Lorenc, 1998]



4D-Var TGN w/ CG in NWP

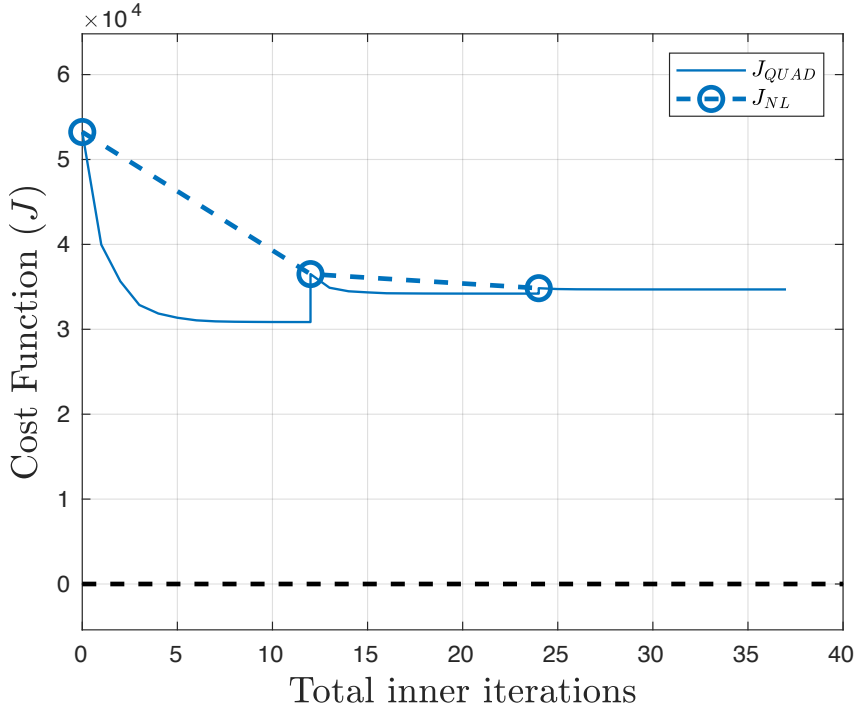
[Courtier et al., 1994; Lorenc, 1998]



Nonlinear versus quadratic cost function reduction

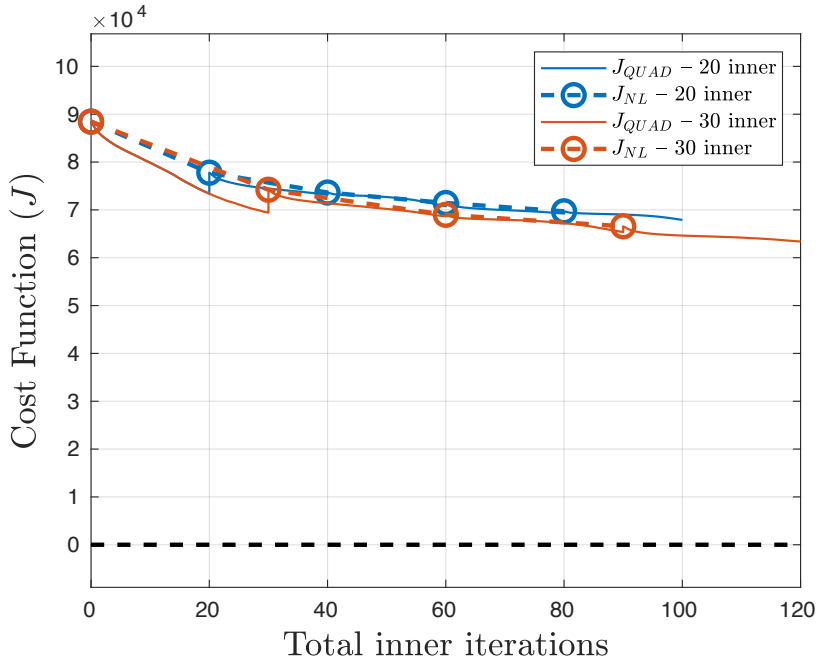
The quadratic approximation error shows up when H is nonlinear

Nearly linear case: 3D-Var 15km NWP w/ GTS observations over North America (N = 27x10^6, M = 36x10^3)



- Differences between J_{NL} and J_{QUAD} are larger when increment is of larger magnitude (e.g., 1st outer iteration)

Weakly nonlinear case: 3D-Var 3km NWP w/ Community Radiative Transfer Model (CRTM) for infrared radiances over Eastern CONUS (N = 497x10^6, M = 114x10^3)

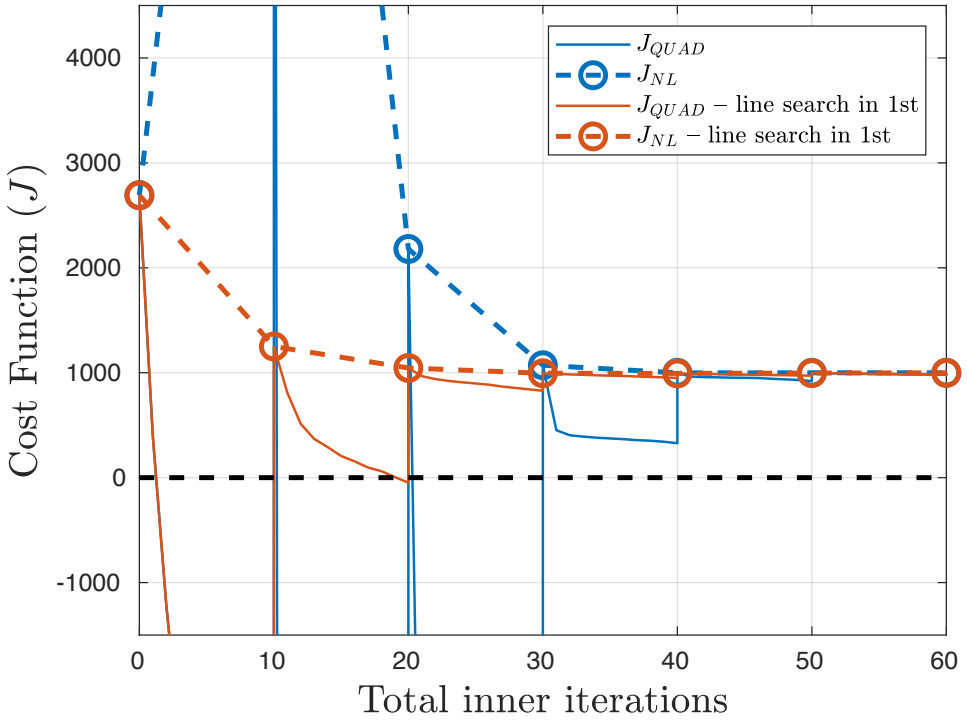


- J_{NL} convergence rate depends on number of inner iterations
- Cost function has not converged despite seemingly growing linearity

Nonlinear versus quadratic cost function reduction

The quadratic approximation error shows up when H is nonlinear

Very nonlinear case: 4D-Var Black Carbon emission inversion with assumed lognormal background error distribution and in-situ obs. (N = 250x10^3, M = 12x10^3)



WRFDA-Chem line search described in Guerrette and Henze (2017)

Standard TGN:

- J_{NL} increases in 1st iteration
- J_{QUAD} is *negative* in early outer iterations

TGN + line search after 1st outer iteration:

- Evaluate $J(\boldsymbol{x}^0 + \gamma \cdot \delta \boldsymbol{x}^1)$ ~9 times to determine optimal step length (γ)
- Improves J_{NL} convergence rate
- Reduces nonlinearity in subsequent iterations, measured by J_{QUAD} error
- Additional expense, but prevents divergence in extremely nonlinear problems

Q: Why is it called “truncated” Gauss-Newton?
A:

- The rank-l approximation to the Hessian inverse, $\mathbf{A}^{-1} \cong \widetilde{\mathbf{A}}_l^{-1}$, is truncated by the number of inner loop iterations (l) chosen, producing an approximate solution, $\widehat{\boldsymbol{x}} \cong \widehat{\boldsymbol{x}}_l = \widetilde{\mathbf{A}}_l^{-1}\boldsymbol{b}$. The non-truncated Gauss-Newton solution requires a rank-N Hessian inverse (and N basis vectors/iterations)
- Due to the minimum-residual condition for deriving $\alpha_j, \boldsymbol{q}_j$ in most iterative Krylov methods, they converge much faster than a rank-l eigen-decomposition of $\mathbf{A}$ for solving $\mathbf{A}\widehat{\boldsymbol{x}} = \boldsymbol{b}$
- Several tradeoffs when choosing l:
  - Each iteration increases memory and wall-time requirements
  - Higher rank approximation yields a more accurate solution to the quadratic problem, but not necessarily to the nonlinear problem
  - Later iterations suffer from round-off error that cause loss of orthogonality/conjugacy between basis vectors ($\boldsymbol{q}_j$’s); can be mitigated with re-orthogonalization (e.g., Modified Gramm-Schmidt algorithm) or by using fewer iterations
- Note: a convergence criteria is useful to cut-off inner loop when # observations is small (available in WRFDA)

On the loss of orthogonality

(only showing 1st outer iteration)

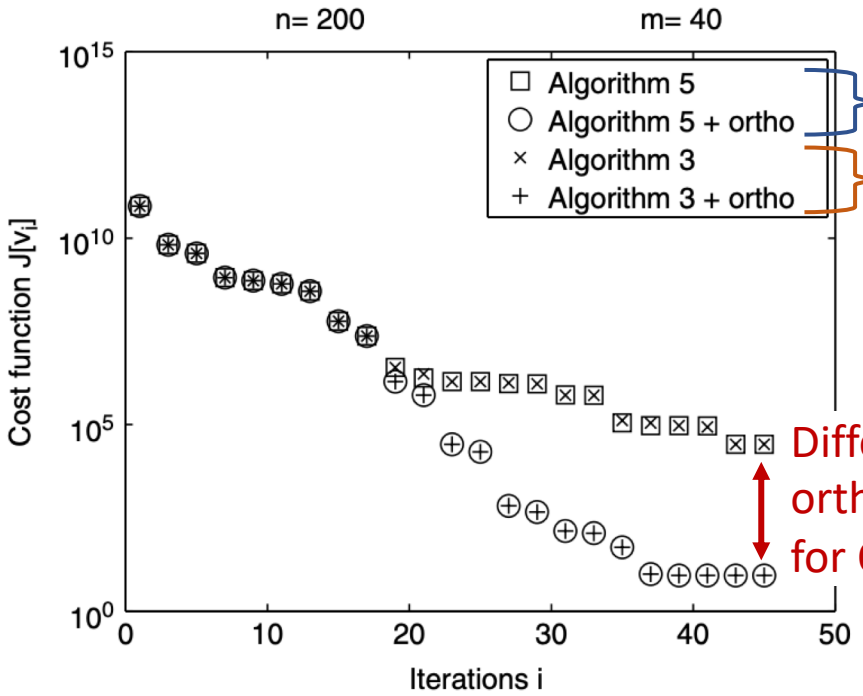


Figure 1. The value of the quadratic function $J[\mathbf{v}_i]$ during the solver iterations i. Here, $m = 40$ and $n = 200$. Results are displayed for algorithm 3 with (○) and without (□) orthogonalization and for algorithm 5 with (+) and without (×) re-orthogonalization. Note that the values of the cost function at the last iteration are 9 and 3×10^4, respectively.

Gratton and Tshimanga (2009), QJRMS

RPCG: CG performed in observation space to reduce computational costs

CG performed in model space as presented in previous slides

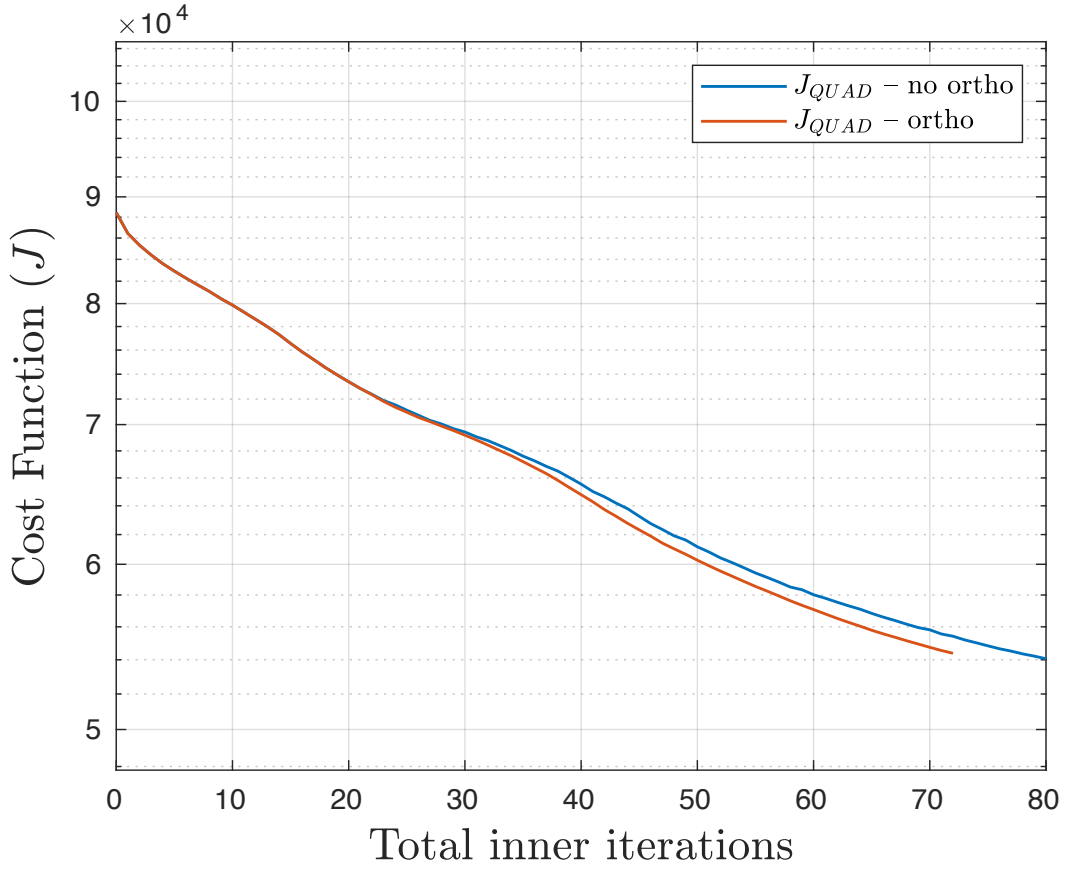
Small linear system
model parameters, N = 200
observations, M = 40

As # iterations approaches M, the loss of orthogonality causes J_{QUAD} to diverge from best numerical approximation

This may be less of an issue when M is very large in real data applications…

On the loss of orthogonality

(only showing 1st outer iteration)



LARGE, weakly nonlinear system
model parameters, N = 497x10^6
observations, M = 114x10^3
(same problem as slide 14)

iterations << M; rate of divergence for J_{QUAD} without re-orthogonalization is small for reasonable l

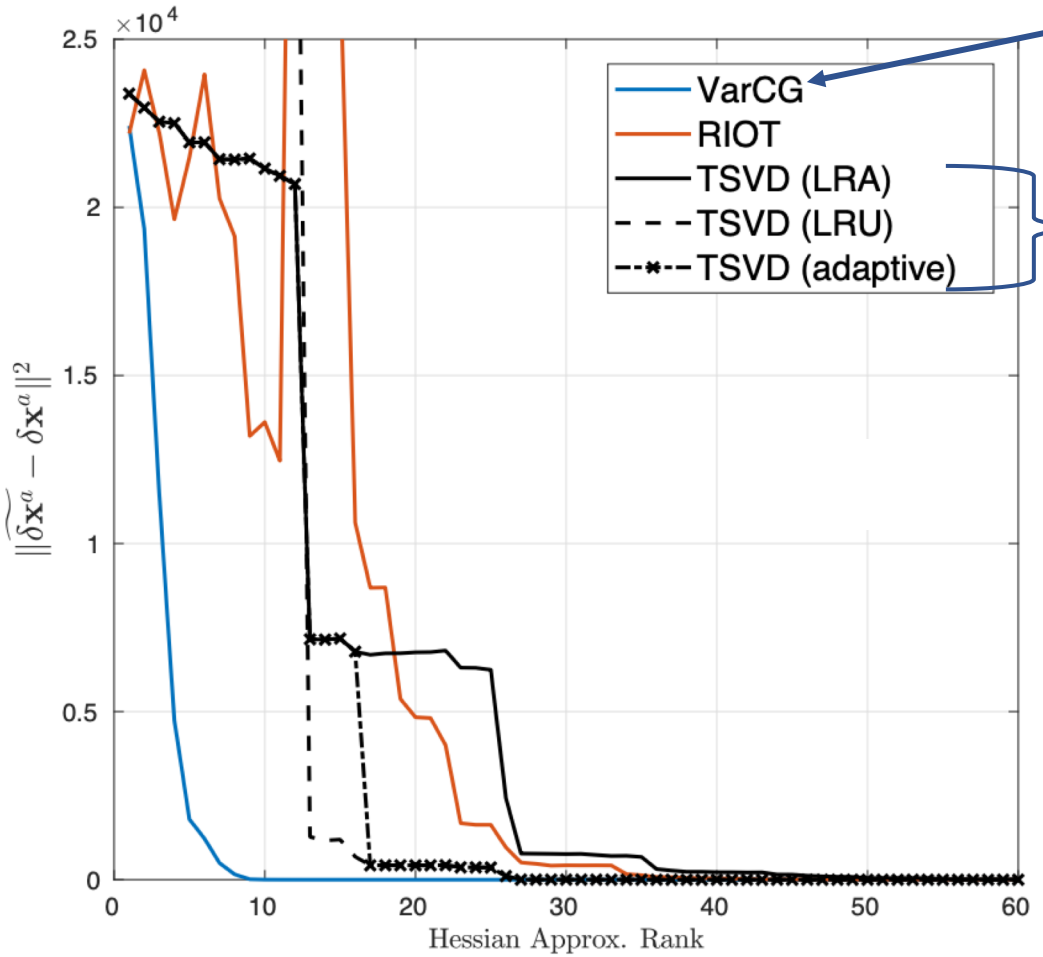
Re-orthogonalization has high memory cost, which must be weighed against application-dependent needs and benefit (MUCH more expensive in EnVar). It is likely used in operations due to demands of accuracy.

CG/Lanczos converges faster than TEVD

Black Carbon Emission flux inversion with WRFDA-Chem

Bousserez, Guerrette, and Henze (submitted manuscript)

Norm of increment residual error, relative to fully converged $\delta \boldsymbol{x}$



Lanczos Recurrence

Same as truncated eigenvalue decomposition (TEVD)

Lanczos Recurrence and Eigen-pairs

- Hessian eigenvalues are a useful diagnostic for comparing observation DOFs
- Hessian eigen-pairs can be used in preconditioning subsequent outer iterations

Lanczos produces a special basis set, $\mathbf{Q}_l$, that satisfies

$$\mathbf{Q}_l^{\mathrm{T}}\mathbf{A}\mathbf{Q}_l = \mathbf{K}_l \in \mathbb{R}^{l\times l}, \text{ which is an } l\times l \text{ tridiagonal matrix}$$

An eigen-decomposition for $\mathbf{K}_l$ is found easily ($l \lesssim 100$) and can be used to produce a rank-l decomposition of $\mathbf{A}$:

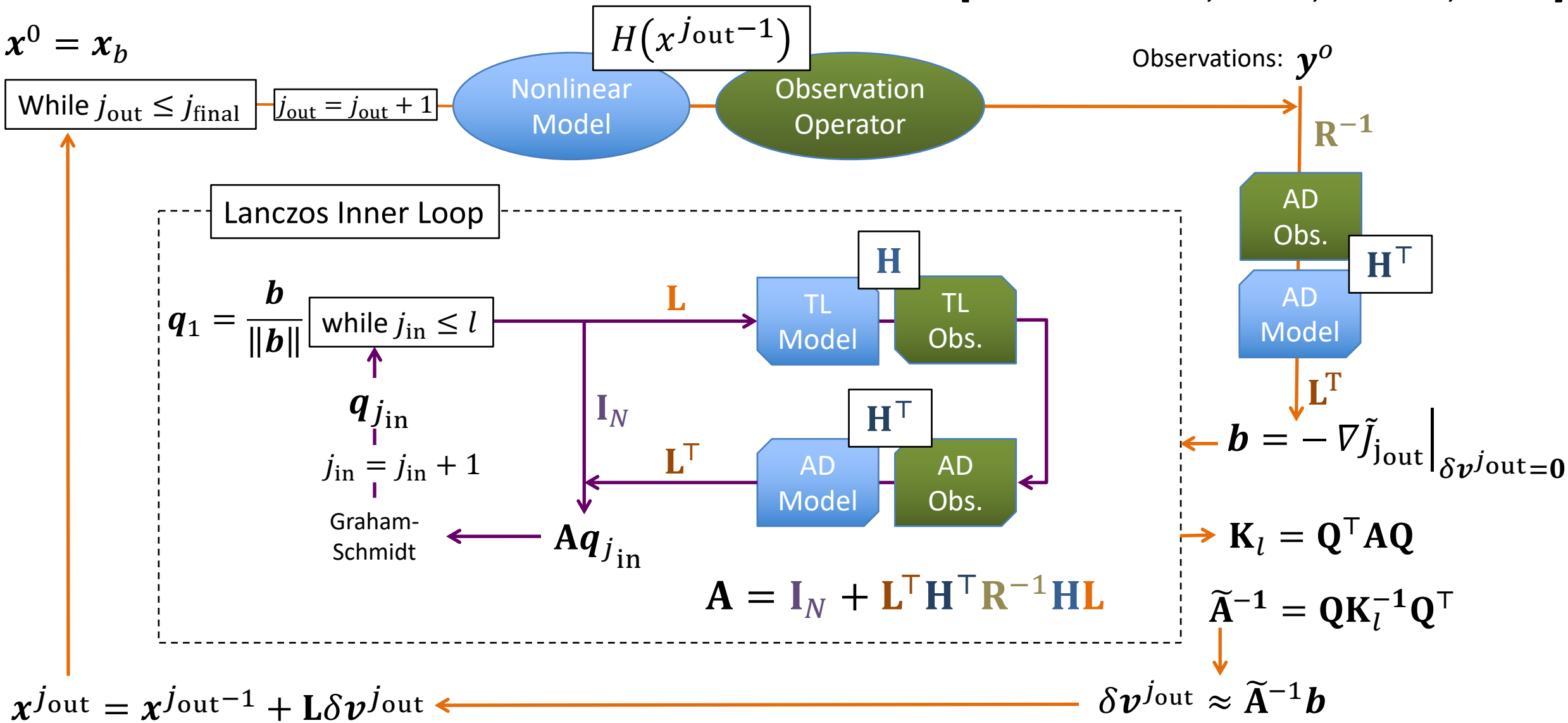
$$\mathbf{A} \cong \mathbf{Q}_l\mathbf{K}_l\mathbf{Q}_l^{\mathrm{T}} = \mathbf{Q}_l\mathbf{Z}_l\mathbf{\Lambda}_l\mathbf{Z}_l^{\mathrm{T}}\mathbf{Q}_l^{\mathrm{T}} = \mathbf{E}_l\mathbf{\Lambda}_l\mathbf{E}_l^{\mathrm{T}}$$

diagonal Ritz value matrix (good approximations to leading eigenvalues of $\mathbf{A}$)

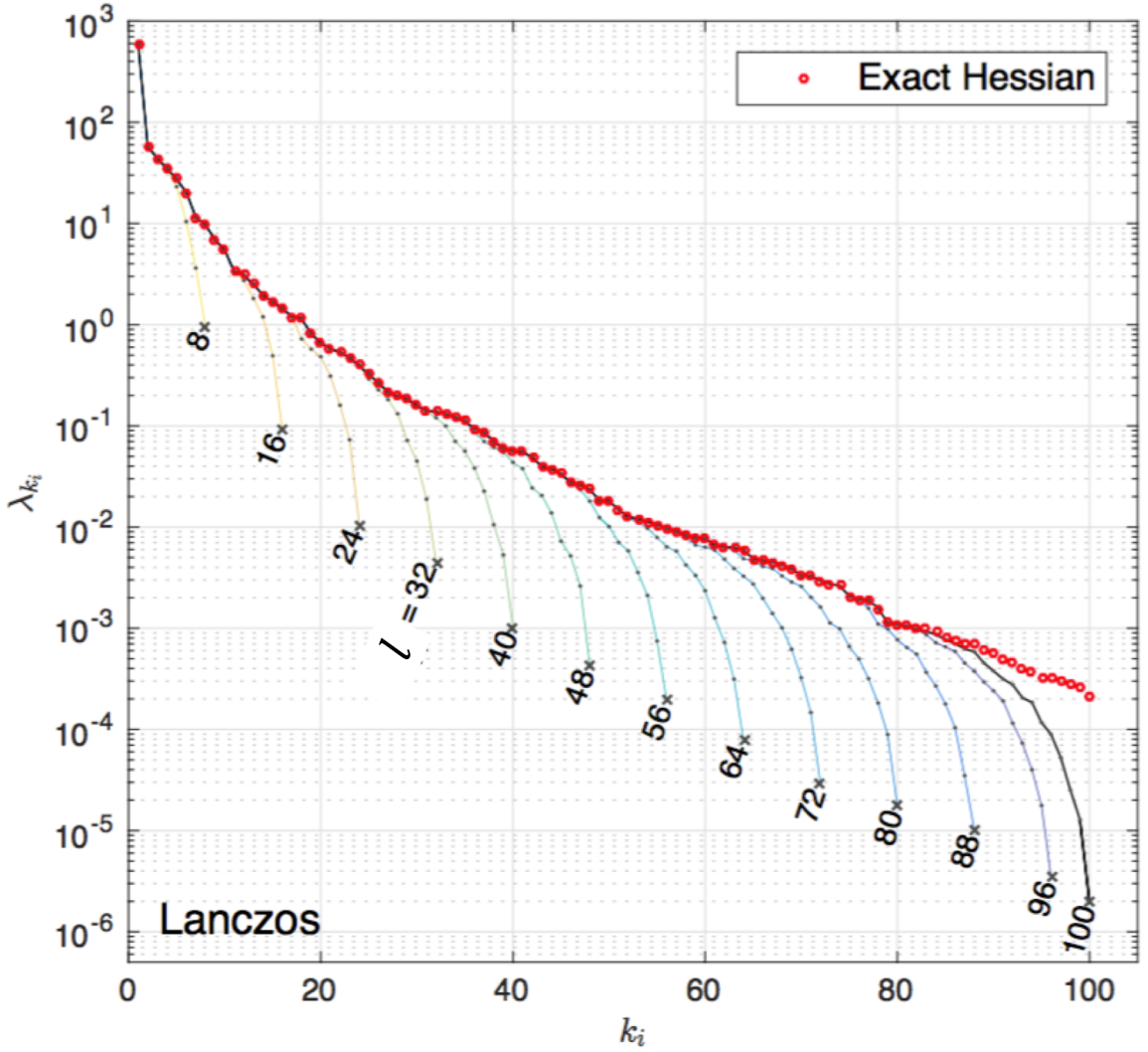
Ritz vector matrix (good approximations to leading eigenvectors of $\mathbf{A}$)

4D-Var TGN w/ Lanczos in NWP

[Courtier et al., 1994; Lorenc, 1998]



Example: approximated Eigenvalues



Eigenvalues

Eigenvalue Index

- Each colored curve shows the Ritz values (approximate eigenvalues) of the Hessian (minus 1) from the Lanczos recurrence for a different number of iterations (l)
- Leading eigenvalues are approximated well (matching exact Hessian eigenvalues)
- Trailing/intermediate eigenvalues are severely under-estimated, because each non-converged Ritz-mode provides a mixture of eigen-modes

Relevant WRFDA namelist settings

&wrfvar6

1. max_ext_its [int]: number of outer loop iterations
2. ntmax [int]: maximum number of inner loop iterations (unless converged)
   (specify as many values as max_ext_its)
3. eps [float]: relative reduction in $(\nabla \tilde{J}_i)^{\mathrm{T}}(\nabla \tilde{J}_i)$ for convergence test
   (specify as many values as max_ext_its)
4. orthonorm_gradient [bool]: use modified Gramm-Schmidt for re-orthogonalization
5. use_lanczos [bool]: use Lanczos recurrance instead of CG; note that WRFDA's Lanczos option always includes re-orthogonalization

Concluding

- WRFDA provides a testing ground for TGN and Krylov Subspace methods on regional NWP problems
- Can be used to learn about the properties of those algorithms and the eigen-spectra of realistically sized applications
- Note: 3DEnVar in WRFDA uses the same minimization algorithms as 3D-Var with addition of ALPHA control variable for ensemble perturbations
- Multi-resolution 4D-Var (coming soon to WRFDA) will be closer to the capability used at NWP centers, utilizing a lower resolution for the quadratic minimization
- New algorithms are (or will be) used in next generation DA systems based around OOPS (including JEDI)
  - Full $\mathbf{B}$ preconditioning instead of sqrt($\mathbf{B}$)
  - Observation (dual) space minimization instead of model (primal) space
  - Block algorithms in the inner loop (extra slides)

Block inner-loop algorithms for TGN (outside WRFDA)

Recall for serial Krylov $\mathcal{K}_l(\mathbf{A}, \boldsymbol{b}) \cong \text{span}\{\boldsymbol{b}, \mathbf{A}\boldsymbol{b}, \mathbf{A}^2\boldsymbol{b}, \ldots, \mathbf{A}^{l-1}\boldsymbol{b}\}$

1. Randomized Singular Value Decomposition (RSVD)

   Simultaneous parallel multiplications of Hessian by Gaussian random noise
   Reduces number of inner loop iterations, which are often time-consuming

   $\mathbf{Y}_{l,m}(\mathbf{A}) \cong \text{span}\{\boldsymbol{A}^m \boldsymbol{\Omega}_m\}$ where $\boldsymbol{\Omega}_m = [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_m] \sim \mathcal{N}(0,1)$

   Halko et al. (2011); Bousserez and Henze (2018); Bousserez, Guerrette, and Henze (submitted)

2. Block Krylov More efficient use of gradient information, but very expensive when using full re-orthogonalization

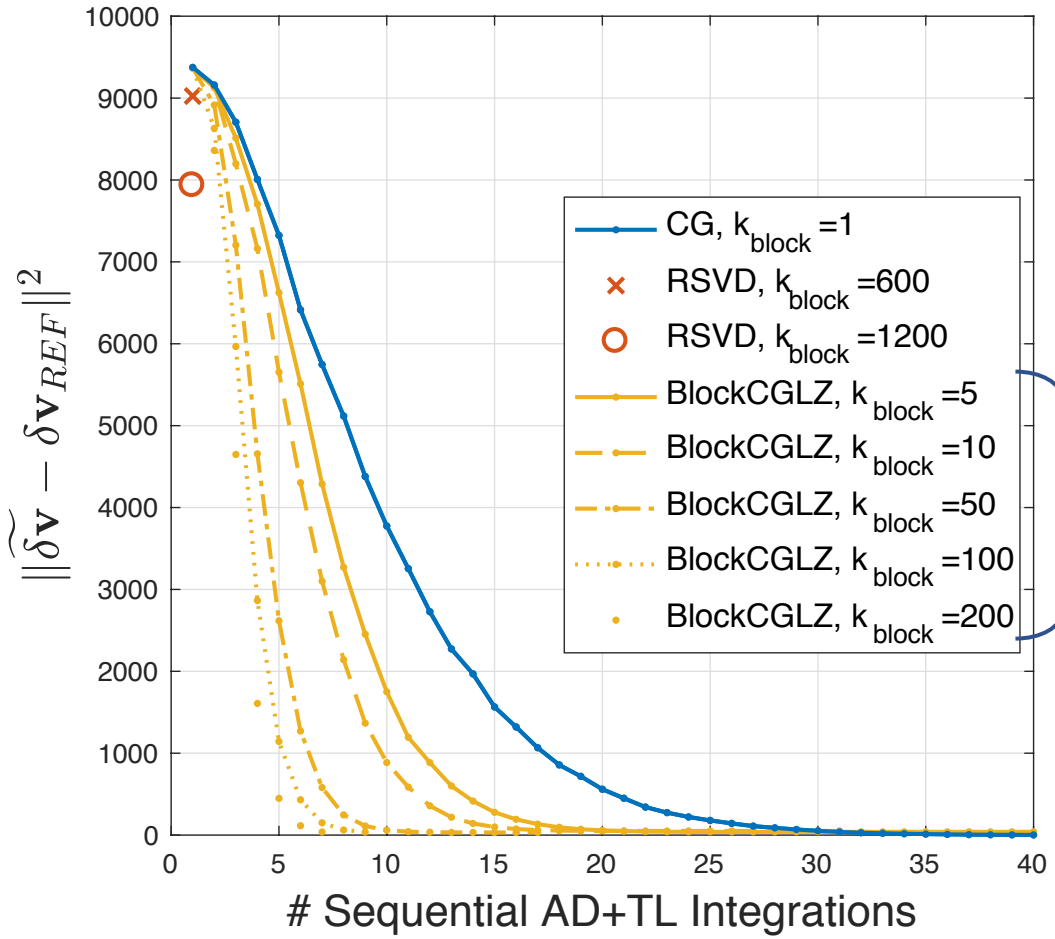
   $\mathcal{K}_{l,m}(\mathbf{A}, \boldsymbol{\beta}_m) \cong \text{span}\{\boldsymbol{\beta}_m, \mathbf{A}\boldsymbol{\beta}_m, \mathbf{A}^2\boldsymbol{\beta}_m, \ldots, \mathbf{A}^{l-1}\boldsymbol{\beta}_m\}$ where $\boldsymbol{\beta}_m$ contains realizations of $\boldsymbol{b}$

   Each gradient realization comes from a different forecast ensemble member

   Golub and Underwood (1977); Golub and Van Loan (1996,2013); Musco and Musco (2015); Mercier et al. (2018,2019); Bousserez, Guerrette, and Henze (submitted)

Inner-loop convergence of “block” algorithms

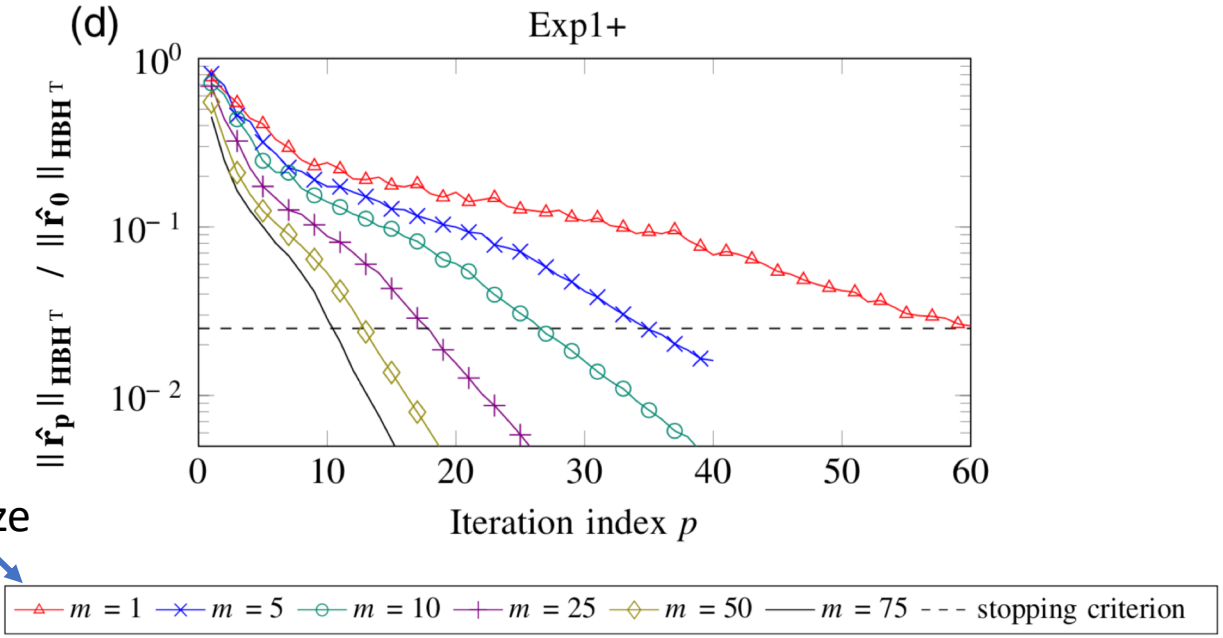
Experimental WRFDA 4D-Var code, 60km CONUS domain and 65K conventional meteorological obs.



Block Size

Guerrette et al. (unpublished work)

Regional AROME 3D-Var EDA, 7km Europe domain, 2K radar obs. and 500K IR and MW radiance obs.



Mercier et al. (2019), QJRMS