Efficient Computation of the Posterior Covariance Matrix in Large-Scale Variational Data Assimilation Problems

Alison Ramage and Kirsty Brown,
Mathematics and Statistics,
University of Strathclyde,
Glasgow, Scotland

Igor Gejadze,
National Research Institute of
Science and Technology for
Environment and Agriculture,
Montpelier, France

Data assimilation

- Numerical weather prediciton is an IVP: given initial conditions, forecast atmospheric evolution.
- Data assimilation is a technique for combining information such as observational and background data with numerical models to obtain the best estimate of state of a system (initial condition).
- Other application areas include hydrology, oceanography, environmental science etc.
- Variational assimilation is used to find the optimal analysis that minimises a specific cost function.

Motivation

Data assimilation problem

- Evolution process:

$$
\begin{aligned}
\frac{\partial \phi}{\partial t} &= F(\phi) + f, \qquad && t \in (0,T), \\
\phi\big|_{t=0} &= u, && \phi, u \in X,\ \phi \in Y
\end{aligned}
$$

true initial state	$\bar{u}$
true state evolution	$\bar{\phi}$
observation operator	$C_o : Y \to Y_o$
observations	$y = C_o \bar{\phi} + \xi_o$
background function	$u_b = \bar{u} + \xi_b$
background error	ξ_b
observation error	ξ_o

Discrete least-squares problem

- observations distributed within time interval (t_0, t_n)
- find $\mathbf{u}$ which minimises

$$
\begin{aligned}
J(\mathbf{u}) &= \frac{1}{2}(\mathbf{u}-\mathbf{u}_b)^T V_b^{-1}(\mathbf{u}-\mathbf{u}_b) \\
&+ \frac{1}{2}\sum_{i=0}^{N}(C_o(\mathbf{u}_i)-\mathbf{y}_i)^T V_o^{-1}(C_o(\mathbf{u}_i)-\mathbf{y}_i)
\end{aligned}
$$

subject to $\mathbf{u}_i$, $i = 1, \ldots, N$ satisfying

$$
\mathbf{u}_{i+1} = \mathcal{M}_{i,i+1}(\mathbf{u}_i), \qquad i = 0, \ldots, N-1.
$$

- discrete nonlinear evolution operator $\mathcal{M}_{i,i+1}$

Incremental 4D-Var

- Rewrite as an unconstrained minimisation problem using Lagrange's method.

- Incremental approach: linearise evolution operator and solve linearised problem iteratively.

- This involves a tangent linear model (TLM) and its adjoint.

- Each iteration requires one forward solution of the TLM equations and one backward solution of the adjoint equations.

Hessian matrix

- Linear system (Gauss-Newton method):

$$\mathcal{H}(\mathbf{u}_k)\delta\mathbf{u}_k = G(\mathbf{u}_k) \qquad (1)$$

Hessian of the cost function $\mathcal{H}$
gradient of the cost function $G(\mathbf{u}_k)$

- Solve (1) using PCG.
- Convergence depends on conditioning of the Hessian

$$\mathcal{H} = V_b^{-1} + R^T C_o^T V_o^{-1} C_o R.$$

- $\mathcal{H}$ is often too large to be stored in memory: all we need for PCG is $\mathcal{H}\mathbf{v}$.
- Evaluating $\mathcal{H}\mathbf{v}$ is very expensive, so we need a good preconditoner.

Approximating the inverse Hessian

- $\mathcal{H}^{-1}$ represents an approximation of the Posterior Covariance Matrix (PCM).
- The PCM can be used to find confidence intervals and carry out *a posteriori* error analysis.
- $\mathcal{H}^{-1/2}$ can be used in ensemble forecasting.
- $\mathcal{H}^{-1/2}$ can be used for preconditioning of the update equation (1).

- Our aim here is to construct a limited-memory approximation to $\mathcal{H}^{-1}$ using only matrix-vector multiplication.

First level preconditioning

- Use the background covariance matrix V_b.

- Projected Hessian:

$$H = (V_b^{1/2})^T \mathcal{H} V_b^{1/2} = I + (V_b^{1/2})^T R^T C_o^T V_o^{-1} C_o R V_b^{1/2}.$$

- Easy to recover $\mathcal{H}$ in the original space.

- Eigenvalues of H are usually clustered in a narrow band above one, with few eigenvalues distinct enough to contribute noticeably to the Hessian value.

- This makes $\mathcal{H}$ amenable to limited-memory approximation.

Limited-memory approximation

- Find n_e leading eigenvalues and orthonormal eigenvectors using the Lanczos method.

- Construct approximation

$$H \approx I + \sum_{i=1}^{n_e} (\lambda_i - 1)\mathbf{u}_i \mathbf{u}_i^T$$

- Easy to evaluate matrix powers:

$$H^p \approx I + \sum_{i=1}^{n_e} (\lambda_i^p - 1)\mathbf{u}_i \mathbf{u}_i^T$$

Second level preconditioning

- Construct a multilevel approximation to H^{-1} based on coarser grids (where it is cheaper to use Lanczos).
- Discretise evolution equation on a grid with $m+1$ nodes (level 0).
- Grid level k contains $m_k = m/2^k + 1$ nodes.
- Grid transfer matrices using piecewise cubic splines:
  - Restriction matrix R_c^0 from $k=0$ to $k=c$.
  - Prolongation matrix P_0^c from $k=c$ to $k=0$.
- Identity matrix I_k on grid level k.
- Hessian H_0 available on finest grid level.

Grid transfers with "correction"

- Need grid transfer operators which minimise the loss of information between grid levels.
- Introduce specific operators which transfer a matrix between a course grid level c and a fine grid level f.

  - From coarse to fine:

$$M_{c \to f} = P_f^c (M_c - I_c) R_c^f + I_f$$

  - From fine to coarse:

$$M_{f \to c} = R_c^f (M_f - I_f) P_f^c + I_c$$

Outline of multilevel algorithm

- Represent H_0 at a given level (k, say):

$$H_{0\to k} = R_k^0 (H_0 - I_0) P_0^k + I_k$$

- Precondition to improve eigenvalue spectrum:

$$\tilde{H}_{0\to k} = (B_k^{k+1})^T H_{0\to k} B_k^{k+1}$$

- Find n_k eigenvalues/eigenvectors of $\tilde{H}_{0\to k}$ using the Lanczos method.

- Approximate $\tilde{H}^{-1}_{0\to k}$:

$$\tilde{H}^{-1/2}_{0\to k} \approx I_k + \sum_{i=1}^{n_k} \left(\frac{1}{\sqrt{\lambda_i}} - 1 \right) \mathbf{u}_i \mathbf{u}_i^T .$$

Preconditioners

- On coarsest grid, level $k+1$ does not exist so set $B_k^{k+1} = I_k$.

- For other levels, construct preconditioners recursively:

$$B_k^{k+1} = \left[B_{k+1}^{k+2} \tilde{H}_{0\to k+1}^{-1/2} \right]_{\to k}, \quad B_k^{k+1^T} = \left[\tilde{H}_{0\to k+1}^{-1/2} B_{k+1}^{k+2^T} \right]_{\to k}$$

- Square brackets represent projection to the correct grid level using "corrected" grid transfers, e.g.

$$[M_{k+1}]_{\to k} = R_k^{k+1}(M_{k+1} - I_{k+1})P_{k+1}^k + I_k$$

Finest level

- We already have H_0, so precondition to obtain

$$\tilde{H}_0 = {B_0^1}^T H_0 B_0^1$$

- Find n_0 eigenvalues/eigenvectors of $\tilde{H}_0$ using the Lanczos method.

- Approximate $\tilde{H}_0^{-1}$:

$$\tilde{H}_0^{-1} \approx I_k + \sum_{i=1}^{n_0} \left(\frac{1}{\lambda_i} - 1 \right) \mathbf{u}_i \mathbf{u}_i^T$$

- Recover projected inverse Hessian using

$$H_0^{-1} = B_0^1 \tilde{H}_0^{-1} {B_0^1}^T$$

Summary

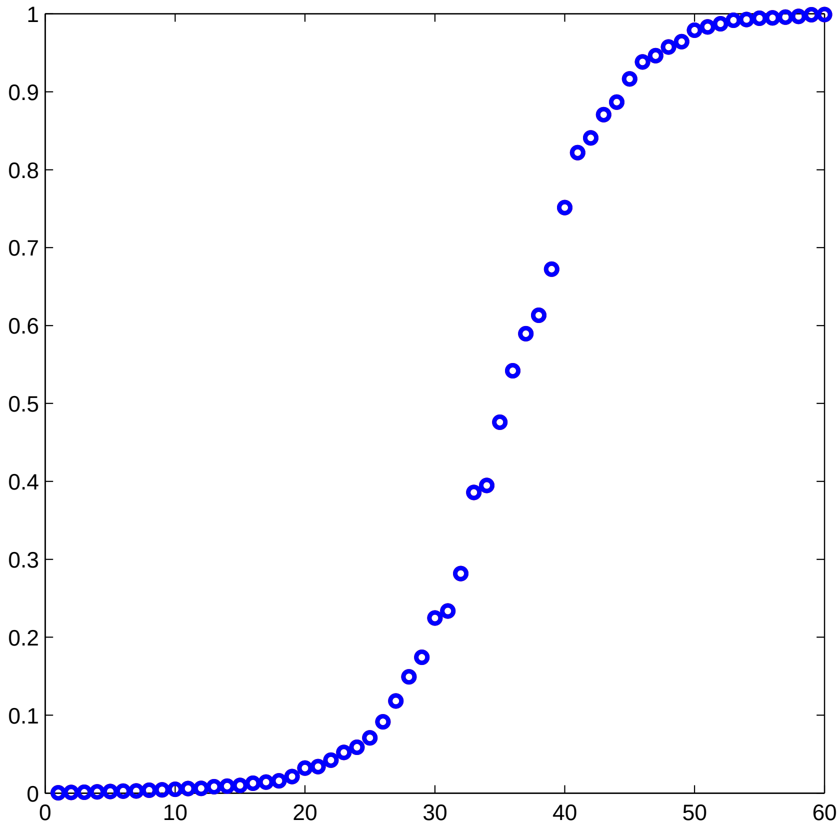
- At each level:
  - represent H_0 on this level;
  - find its eigenvalues/vectors using the Lanczos method;
  - use these to approximate H_0^{-1} in this level;
  - use preconditioners based on these local approximations to accumulate the key eigenvalue structure from every grid level.

Example

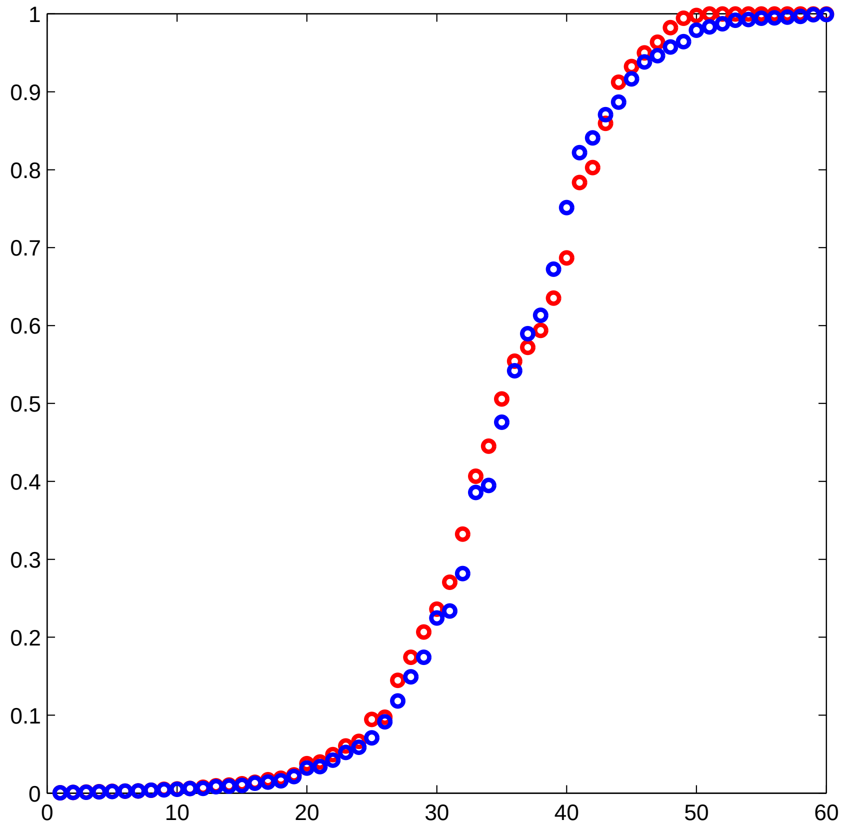
- Test using 1D Burgers' equation.
- 1D uniform grid with 5 sensors.
- Multilevel preconditioning with four grid levels: $k = 0, 1, 2, 3$ with 401, 201, 101 and 51 grid points, respectively.
- Vary number of eigenvalues chosen on each grid level (n_0, n_1, n_2, n_3).
- Compare eigenvalues of exact and computed projected inverse Hessian on the finest grid level $k = 0$.

Eigenvalues of the inverse Hessian

- Exact (blue circles), approximated (red circles)



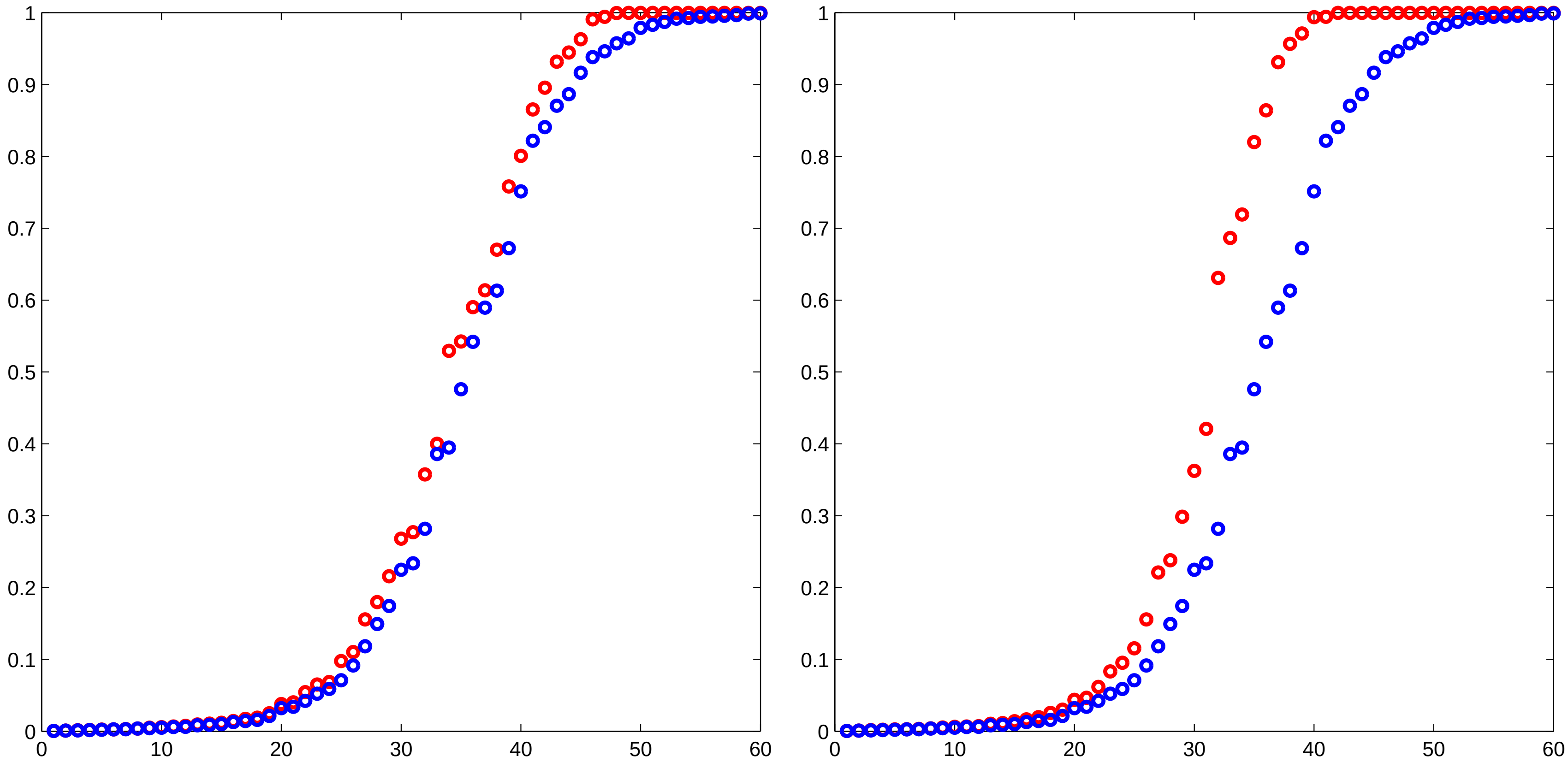
(400,0,0,0)



(4,8,16,32)

Eigenvalues of the inverse Hessian

- Exact (blue circles), approximated (red circles)



(0,8,16,32) (0,0,16,32)

PCG iteration for Newton step

- measurement units:
  - memory: length of vector on finest grid L
  - cost: cost of MVM on finest grid M

Preconditioner	# CG iterations	storage	cost
none	57	0L	57M
MG(400,0,0,0)	1	400L	402M
MG(4,8,16,32)	4	16L	34M
MG(0,8,16,32)	5	12L	14M
MG(0,0,16,32)	8	8L	10M

Conclusions and next steps

- Although this is only one test problem, multilevel preconditioning looks promising for constructing a good limited-memory approximation to H^{-1}.
- The balance between restrictions on memory/cost limitations may vary between particular applications.
- Identifying globally appropriate values for $(n_0, n_1, n_2, .n_3)$ is tricky.

- Future investigations will include

- constructing local Hessians based on sensor domains of influence;
- applying Lanczos locally at sensor level.

It is sometimes nice in Scotland. . .