Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2013) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Science, ICCS 2013

Dynamic sensor location using model singular vectors

Alexandru Cioaca[a], Adrian Sandu[a], Vishwas Rao[a]

[a]*Virginia Tech, Department of Computer Science, Blacksburg,VA-24060, USA*

Abstract

Click here and insert your abstract text.

Keywords: Type your keywords here, separated by semicolons ;

1. Introduction

The singular vector method searches for the error structures in the analysis field at current time which, propagated by the forecast model, achieve maximum growth at the verification time t_v. Singular vectors (SVs) are the directions of fastest error growth over a finite time interval [**?** **?** **?** **?**]. Buizza and Montani [**?**] showed that SVs can identify the most sensitive regions of the atmosphere for targeted observations. They are useful as long as the linearity assumption of error propagation holds [**?**]. Majumdar et al.[**?**] compare the SV approach for observation targeting to the ensemble transform Kalman filter. Palmer et al. [**?**] argue that for predictability studies an appropriate metric is the perturbation energy. Daescu and Navon [**?**] discuss the adaptive observation problem in the context of 4D-Var data assimilation. Estimation of the optimal placement of adaptive observations is also discussed in [**?** **?**]. Leutbecher [**?**] derives optimal locations of observations by minimizing the variance of the assimilated field; a computationally tractable problem is obtained by projecting the covariance on the subspace of the dominant singular vectors.

Adaptive observations placed in well-chosen locations can reduce the initial condition uncertainties and decrease forecast errors. A number of methods were proposed to "target observations", i.e. to select areas where additional observations are expected to improve considerably the skill of a given forecast. Singular vectors identify sensitive regions of the atmospheric flow and can be used to optimally configure the observational network.

2. Model singular vectors

Singular vectors (SVs) determine the most rapidly growing perturbations in the atmosphere. The magnitude of the perturbation at the initial time t_0 is measured in the L^2 ("energy") norm defined by a symmetric positive definite matrix $\mathbf{E}$

$$\| \delta x(t_0) \|^2_{\mathbf{E}} = \langle \, \delta x(t_0) \, , \; \mathbf{E} \; \delta \mathbf{x}(t_0) \, \rangle \, . \tag{1}$$

*Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: author@institute.xxx.

Similarly, the perturbation magnitude at the final time t_f is measured in a norm defined by a positive definite matrix $\mathbf{F}$

$$\left\| \delta\mathbf{x}(t_f) \right\|^2_{\mathbf{F}} = \langle\, \delta\mathbf{x}(t_f)\,,\ \mathbf{F}\,\delta\mathbf{x}(t_f)\,\rangle\,. \tag{2}$$

We call the norms (1) and (2) squared the "perturbation energies".

Our main interest is to minimize the forecast uncertainty over a well defined area (the "verification domain" $\Omega^{\mathrm{v}} \subset \Omega$) at a well defined time (the "verification time" t_f).We define a spatial projection operator $\mathbf{\Pi}$ from the whole model domain to the verification domain:

$$\mathbf{\Pi} : \Omega \subset \Re^n \longrightarrow \Omega^{\mathrm{v}} \subset \Re^{n_{\mathrm{v}}}\,, \quad n_{\mathrm{v}} \ll n\,. \tag{3}$$

After a permutation of columns we can write $\mathbf{\Pi} = [\mathbf{I}, \mathbf{0}]$.

The ratio between perturbation energies at t_f (over the verification domain) and at t_0 (over the entire domain) offers a measure of error growth:

$$\begin{aligned}
\sigma^2 &= \frac{\|\mathbf{\Pi}\,\delta\mathbf{x}(t_f)\|^2_{\mathbf{F}}}{\|\delta\mathbf{x}(t_0)\|^2_{\mathbf{E}}} = \frac{\mathbf{\Pi}\,\delta\mathbf{x}(t_f), \mathbf{F}\mathbf{\Pi}\,\delta\mathbf{x}(t_f)}{\delta\mathbf{x}(t_0), \mathbf{E}\delta\mathbf{x}(t_0)} \\
&= \frac{\mathbf{\Pi}\,\mathbf{M}\,\delta\mathbf{x}(t_0)\ ,\ \mathbf{\Pi}\,\mathbf{F}\,\mathbf{M}\,\delta\mathbf{x}(t_0)}{\delta\mathbf{x}(t_0)\ ,\ \mathbf{E}\,\delta\mathbf{x}(t_0)} \\
&= \frac{\delta\mathbf{x}(t_0), \mathbf{M}^*\mathbf{\Pi}^*\,\mathbf{F}\,\mathbf{\Pi}\,\mathbf{M}\delta\mathbf{x}(t_0)}{\delta\mathbf{x}(t_0), \mathbf{E}\,\delta\mathbf{x}(t_0)}
\end{aligned} \tag{4}$$

In (4) we use the fact that perturbations evolve in time according to the dynamics of the tangent linear model (**??**).

SVs are defined as the directions of maximal error growth, i.e. the vectors $s_k(t_0)$ that maximize the ratio σ^2 in equation (4). These directions are the solutions of the generalized eigenvalue problem

$$\mathbf{M}^*\,\mathbf{\Pi}^*\,\mathbf{F}\,\mathbf{\Pi}\,\mathbf{M}\,s_k(t_0) = \sigma_k^2\,\mathbf{E}\,s_k(t_0)\,. \tag{5}$$

The left side of (5) involves one integration with the tangent linear model followed by one integration with the adjoint model. The eigenvalue problem (5) can be solved efficiently using the software package ARPACK [**?**], or its parallel version PARPACK [**?**].

Using the square root of the the symmetric positive definite matrix $\mathbf{E}$ the generalized eigenvalue problem (5) can be reduced to a simple eigenvalue problem

$$\mathbf{E}^{-\frac{1}{2}}\,\mathbf{M}^*\,\mathbf{\Pi}^*\,\mathbf{F}\mathbf{\Pi}\ \mathbf{M}\,\mathbf{E}^{-\frac{1}{2}}\,v_k = \sigma_k^2\,v_k(t_0)\,, \quad v_k = \mathbf{E}^{\frac{1}{2}}\,s_k(t_0)\,. \tag{6}$$

Furthermore, $v_k(t_0)$ are the left singular vectors in the singular value decomposition

$$\mathbf{F}^{\frac{1}{2}}\,\mathbf{\Pi}\,\mathbf{M}\,\mathbf{E}^{-\frac{1}{2}} = U\cdot\Sigma\cdot V^T \quad \text{where} \quad \Sigma = \mathrm{diag}_k\{\sigma_k\}\,, \quad \sigma_k\,u_k = \mathbf{F}^{\frac{1}{2}}\,\mathbf{\Pi}\,s_k(t_f)\,. \tag{7}$$

The SVs s_k are $\mathbf{E}$-orthogonal at t_0 and $\mathbf{F}$-orthogonal at t_f

$$\langle\, s_k(t_0),\, \mathbf{E}s_j(t_0)\,\rangle = 0 \qquad \text{and} \qquad \langle\, \mathbf{\Pi}s_k(t_f),\, \mathbf{F}\mathbf{\Pi}s_j(t_f)\,\rangle = 0 \qquad \text{for} \quad j \neq k\,. \tag{8}$$

The equations (7) and (8) justify the name of "total energy singular vectors". The singular value decomposition of the linear operator $M_{t_0\to t_f}$, with the $\mathbf{E}$ scalar product at t_0 and the $\mathbf{F}$ scalar product at t_f, has the left singular vectors $s_k(t_0)$ and the right singular vectors $s_k(t_f)$. The singular values σ_k are the error amplification factors along each direction s_k.

A special set of energy norms is provided by the choice $\mathbf{F} = \mathbf{\Pi} = \mathbf{I}$ and $\mathbf{E} = \mathbf{P}^a(t_0)^{-1}$. In this case the resulting singular vectors $s_k(t_0)$ evolve into the leading eigenvectors $s_k(t_f)$ of the forecast error covariance matrix $\mathbf{P}^f(t_f)$,

$$\begin{aligned}
\mathbf{M}^T\,\mathbf{M}\,s_k(t_0) &= \sigma_k^2\,\mathbf{P}^a(t_0)^{-1}\,s_k(t_0) \\
\mathbf{P}^a(t_0)\,\mathbf{M}^T\,\mathbf{M}\,s_k(t_0) &= \sigma_k^2\,s_k(t_0) \\
\mathbf{M}\,\mathbf{P}^a(t_0)\,\mathbf{M}^T\,\mathbf{M}\,s_k(t_0) &= \sigma_k^2\,\mathbf{M}\,s_k(t_0) \\
\mathbf{P}^f(t_f)\,s_k(t_f) &= \sigma_k^2\,s_k(t_f)\,.
\end{aligned} \tag{9}$$

The eigenvectors $s_k(t_f)$ are called "Hessian singular vectors" (HSVs) associated with the cost functional Ψ (Barkmeijer, 1998). Since the leading eigenvectors of $\mathbf{P}^f(t_f)$ are the directions of maximum variance of forecast error, HSVs define the directions along which we must do a good job of analysis in order to minimize the forecast error at t_f. We assume that the model error is negligible over the period $[t_0, t_f]$. From equation (5) it follows that Hessian singular vectors are the solutions of the following generalized eigenvalue problem

$$\mathbf{M}^* \mathbf{M}\, s_k(t_0) = \sigma_k^2\, \mathbf{P}^a(t_0)^{-1} s_k(t_0) = \sigma_k^2 \left(\nabla^2_{\mathbf{x}_0,\mathbf{x}_0} \Psi\right)\, s_k(t_0)\,. \tag{10}$$

The second relation comes from the fact that the inverse of the analysis covariance matrix is equal to the Hessian matrix of the analysis cost function Ψ in the variational analysis system. This motivates the name *Hessian singular vectors* for the solutions $s_k(t_0)$ of the eigenproblem (**??**).

The eigenvalue problem (10) is solved in the time interval $t_f - t_0$ for each target using an iterative method that require only matrix–vector products (e.g. the ARPACK software, Lehoucq et al. 1998; P_ARPACK, Maschhoff and Sorensen 1996). In (Liao et al., 2006) we have computed the total energy singular vectors for CTMs. To the best of our knowledge this is the first work that attempts the computation of Hessian singular vectors for CTMs.

The singular vectors associated with the largest m singular values,

$$\mathcal{V}_m(t_0) = \left[s_1(t_0), \cdots, s_m(t_0)\right]\,, \tag{11}$$

span a subspace of the state space which will have the maximum influence on the verification area at the verification time.

3. Numerical Model

We apply this computational methodology to the shallow-water equations (SWE), which approximate fluid flow inside a shallow basin. Consider the following two-dimensional PDE system:

$$\begin{aligned}
\frac{\partial}{\partial t}h + \frac{\partial}{\partial x}(uh) + \frac{\partial}{\partial y}(vh) &= 0\\
\frac{\partial}{\partial t}(uh) + \frac{\partial}{\partial x}(u^2h + \frac{1}{2}gh^2) + \frac{\partial}{\partial y}(uvh) &= 0\\
\frac{\partial}{\partial t}(vh) + \frac{\partial}{\partial x}(uvh) + \frac{\partial}{\partial y}(v^2h + \frac{1}{2}gh^2) &= 0\,.
\end{aligned}$$

A numerical model was built to compute the solution of these equations. The space discretization was performed using a finite volume-type scheme and the time discretization using fourth-order Runge-Kutta. This method was introduced by Liska and Wendroff in [**?**].

The spatial domain is square shaped ($\Omega = [-3, 3]^2$), and the integration window is $t^0 = 0 \le t \le t^F = 0.1$. Here $h(t, x, y)$ denotes the fluid layer thickness, and $u(t, x, y)$, $v(t, x, y)$ are the components of the velocity field. For a square grid of size n^2, the model has $3 \cdot n^2$ variables. g is the standard value of the gravitational acceleration. The boundary conditions are periodic in both directions. For ease of presentation, we arrange the n discretized state variables in a column vector

$$x = \begin{bmatrix} \hat{h} \\ \hat{uh} \\ \hat{vh} \end{bmatrix} \in R^n\,.$$

For more advanced applications, such as nonlinear optimization and sensitivity analysis, we need to compute derivatives of the model states and parameters. This can be accomplished using the methodology of adjoint models [**? ?**]. A distinction is made between continuous adjoints, obtained by linearizing the differential equations, and discrete adjoints, obtained by linearizing the numerical method. The resulting adjoint equations are then solved numerically through time integration.

We build the adjoint models associated with our SWE model through automatic differentiation [**?**] using the TAMC tool [**? ?**]. The tangent-linear model (TLM) propagates perturbations forward in time. The first-order

adjoint model (FOA) propagates perturbations backwards in time and can be used to compute the gradients of a scalar cost function defined on the model trajectory. Second-order adjoint models (SOA) compute the product between the Hessian of f and a vector. Second-order adjoint models are considered to be the best approach to compute Hessian-vector products, but have yet to become popular in practice because of their computational demands. When one does not have access to the second-order adjoint, Hessian-vector products can be computed through various approximations, such as finite difference with gradients or the Gauss-Newton approximation.

The overhead of running adjoint models is a crucial aspect of the computational strategy. For our particular implementation, one full SOA integration is about 3.5 times more expensive than a single first-order adjoint run, while the FOA takes 3.7 times longer than the forward run. However, this ratio depends on the numerical methods used to solve the differential equations or the automatic differentiation tool employed. For certain numerical methods used to solve the forward model, it is possible to develop smart strategies of reusing computation which lead to adjoint models that take less time to run than the forward model. An example can be found in [**?**] where the adjoint equations of this very SWE system are derived by hand and then implemented as numerical models.

4. Data assimilation

We apply our computational framework to two data assimilation scenarios with SWE. These two scenarios have very similar setting, the only difference being the amount of observations we assimilate. The 4D-Var parameters are as follows:

- The resolution of the 2D computational grid on which the models operate is 40 grid points on both directions (1600 in total). Thus, the model has 4800 variables.
- The size of the timestep is set at $1e-3$ and the models are configured to run for $1-e1$ seconds ($N = 100$ discrete steps of time).
- The reference solution is synthesized for h as a Gaussian pulse of amplitude $A = 30$ centered on the grid. The u/v fields are made consistent with h with the help of the forecast model.
- The background solution x^b is generated by applying a correlated perturbation on the reference solution for h, u and v.
- The background error covariance B was generated for a standard deviation of 5% with a nondiagonal structure and correlation distance of 5 grid points. This will help the 4D-Var method to spread the information for each grid point to its neighbors.
- The model is ran starting from the reference solution in order to generate the synthetic observations. The observation frequency is set to once every 20 time steps. In order to simulate the effect of noise over observations, we apply a normal random perturbation to the perfect synthetic observations.
- The observation error covariance R is a diagonal matrix, based on the assumption that the observational errors are uncorrelated. The standard deviation of these errors was set to 1% of the largest absolute value of the observations for each variable.
- The observation operator H is configured to select observations for each variable at each point of the grid (observations are dense in space). In a realistic setting, the operator H is enforced by the structure of the observational network.

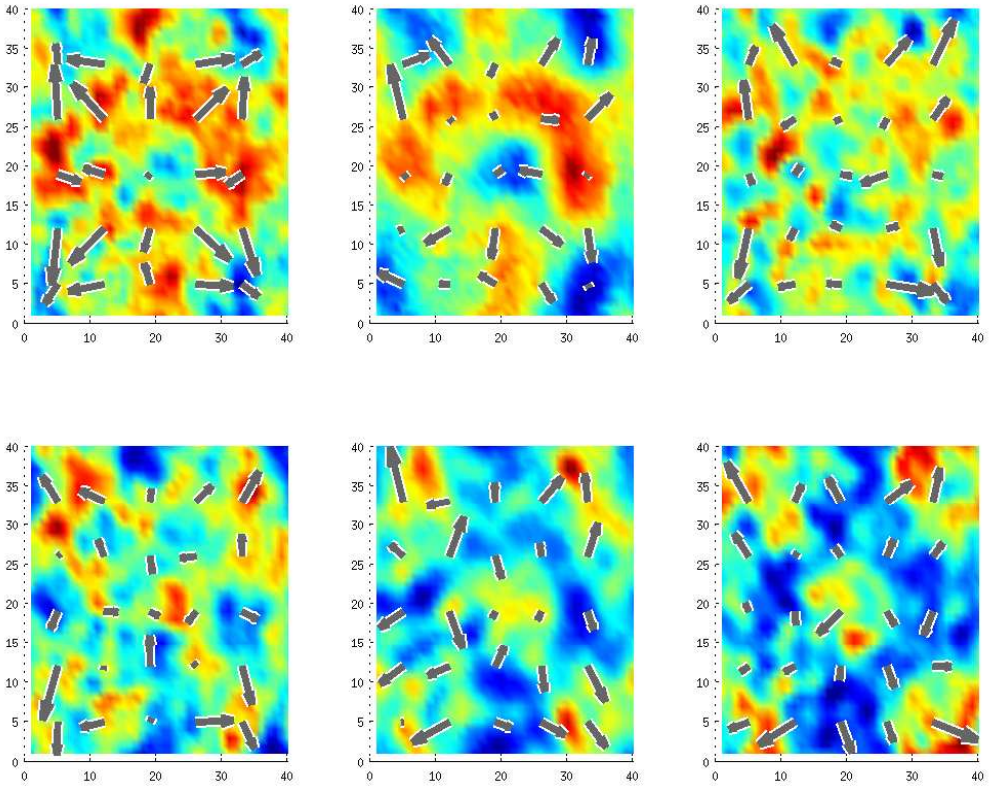
Difference between the two experiments

1. REG300. Observations are available for h, u and v at every other fourth grid point on horizontal and vertical. This leads to 100 observations for each variable and 300 in total.
2. SEL10. Observations are available for h, u and v at 10 selected locations.

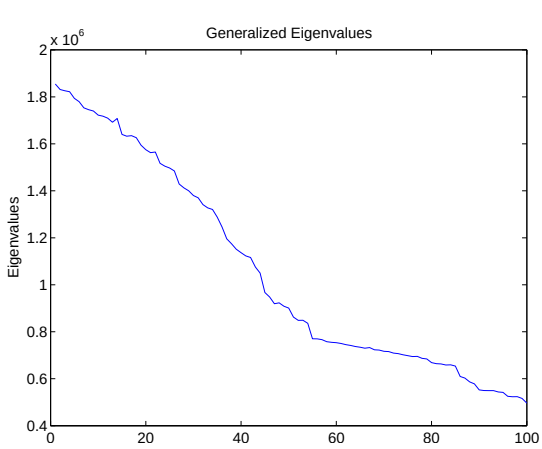
To minimize the 4D-Var cost function, we use the L-BFGS-B solver [**?**]. Since the SWE model does not represent a significant computational burden, we allowed the solver to run for 500 iterations, or until the gradient of the 4D-Var cost function was reduced from a magnitude of $1e+2$ to $1e-4$.

5. Results

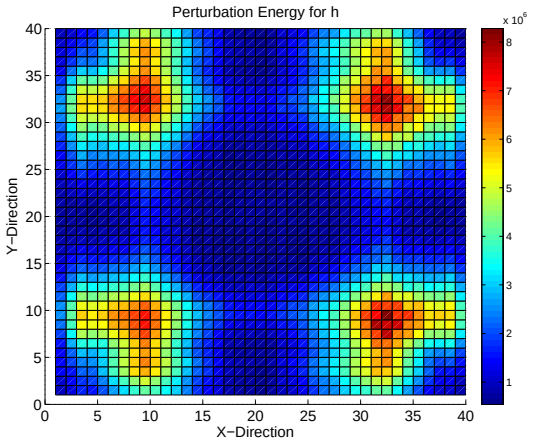
The minimizer of the 4D-Var cost function as provided by L-BFGS-B represents an improved estimate of the initial state of the model assimilating the available observations. The next step is to compute the singular



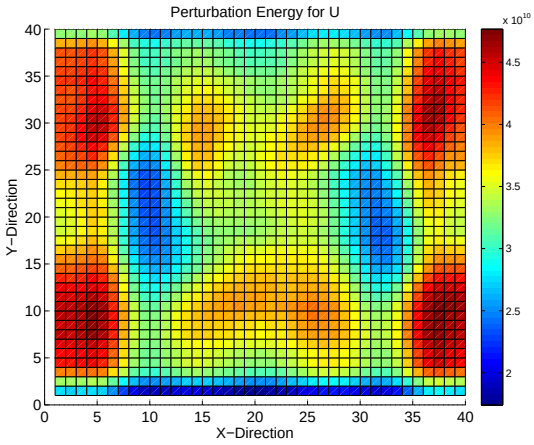
(a) The evolution of H



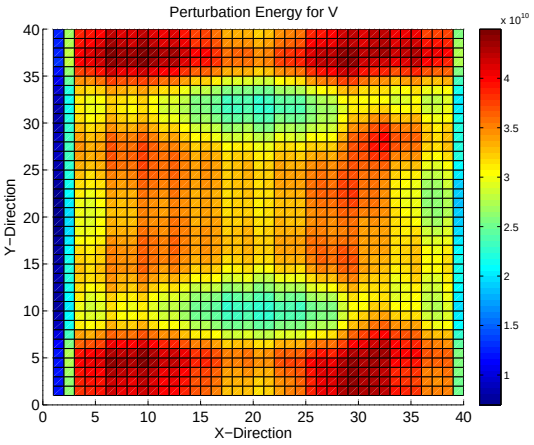
(b) Generalized Eigenvalues



(c) Height



(d) Velocity vector (U)



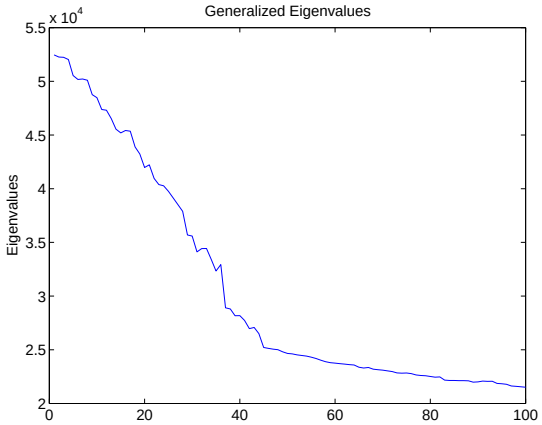
(e) Velocity vector (V)

vectors as described in Section **??**. This requires the solution of a generalized eigenvalue problem. The main computational constraint consists in circumventing the fact that we do not have access to the full Jacobian (and its transpose) of the model states or to the 4D-Var Hessian. Instead, we can compute their action on a vector by using adjoint models. Jacobian-vector products can be computed by running the TLM initialized with said vector, and the transpose-Jacobian-vector products by running the FOA backwards in time, initialized with the seed vector. 4D-Var Hessian vector-products can be obtained using the second-order adjoint model.

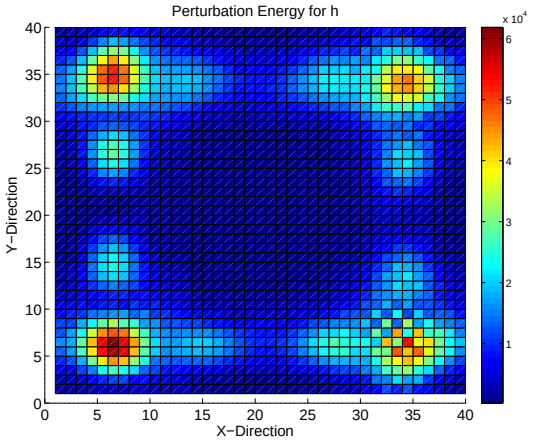
Generalized eigenvalue problems can be solved iteratively when we only have access to matrix-vector products. Classic algorithms such as Arnoldi (reference) or Lanczos (reference) compute the eigenvectors associated with the m largest or smallest eigenvalues, where m corresponds to the number of matrix-vector products to be evaluated and is specified by the user. For our scenario, we compute 100 leading eigenvectors and then compute the perturbation energy associated with them following equation **??**.

We use an interative algorithm to compute the eigenvectors associated with the largest 100 eigenvalues for our two test scenarios, whose values are plotted in Figures **??** and **??**. We notice there is a significant cutoff after the first 50 eigenvalues so we keep only these eigenvectors for computing the energy norm E.

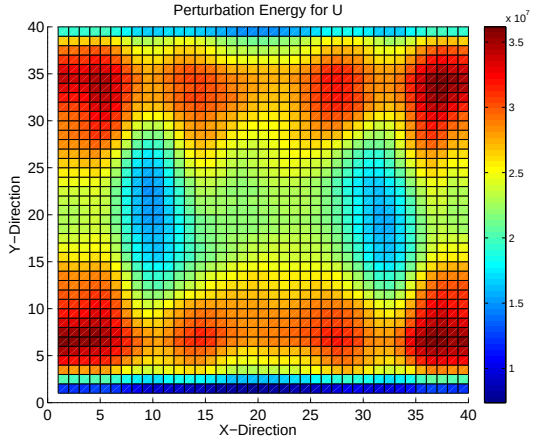
Figures **??** and **??** show the plot of the perturbation energy of h, u and v on the computational grid, for REG300 and SEL10, respectively. We notice similar features across both cases. The profile of the perturbation energy for h, the height of the flow, reveals areas of high uncertainty in the four corners of the computational grid. Meanwhile, the perturbation energy of the wind vector components u and v is considerably larger on the opposite edges of the grid which correspond to the direction represented by the vector component: East-West for u and North-South for



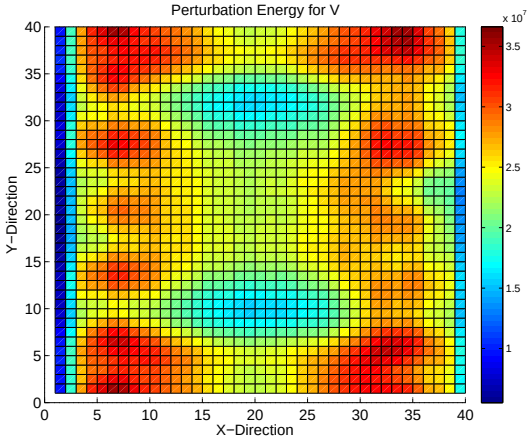
(f) Generalized Eigenvalues



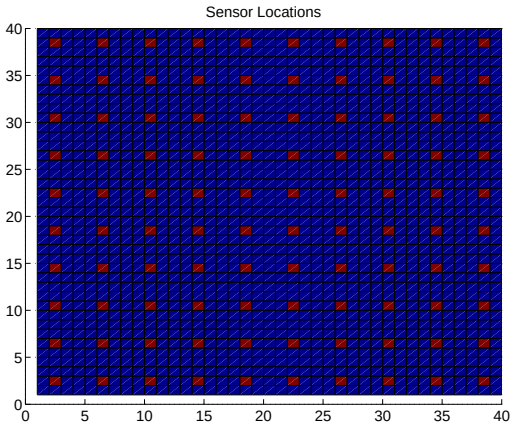
(g) Height



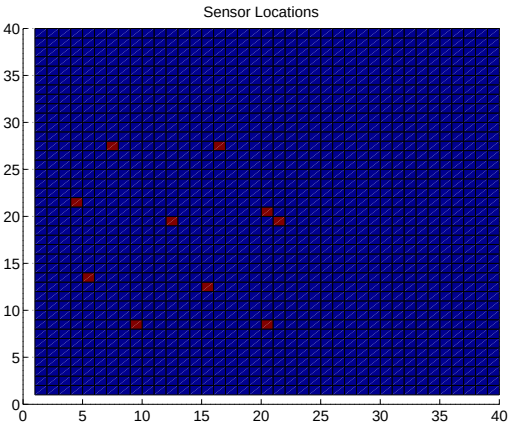
(h) Velocity vector (U)



(i) Velocity vector (V)



(j) Sensor Locations 1



(k) Sensor Locations 2

v. For the u and v variables there can also be noticed visible features in the middle of the grid that are similar between the two cases. This is to be expected, since both cases operate on the same forecast scenario.

The magnitude of the perturbation energies reflects the fact that one of the scenario (REG300) assimilates more information than the other (SEL10). For the former case, the perturbation energies have values of order 10^4 which can be associated with a reanalysis of smaller uncertainty than for other case, where magnitudes are of order 10^6. Also, the areas of high uncertainty in h for the REG100 case are smaller than their equivalents for the SEL10 case. This is a clear indication that the forecast started from the reanalysis obtained from assimilating more information (REG100) is less likely to be affected by perturbations situated along the principal directions of error growth. These directions of error growth constitute the left-hand side of the generalized eigenvalue problem and are the same for any data assimilation scenario. The difference between various scenarios is found on the right-hand side in the matrix norm, the 4D-Var Hessian, which gives us a meaningful measure for the perturbation energy that reflects the particular data assimilation setting used. The 4D-Var Hessian approximates the inverse of the covariance matrix of the errors in the reanalysis, so it can be interpreted as a measure of trust in the reanalysis.

In order to improve the quality of the reanalysis forecast, we need to target for observation (and subsequently, assimilation) those areas where the perturbation energy is large. This means we have to install additional sensors in the corners of the grid and on the borders.

Acknowledgements

This work was supported by National Science Foundation through the awards NSF DMS-0915047, NSF CCF-0635194, NSF CCF-0916493 and NSF OCI-0904397; and by AFOSR through the award FA9550–12–1–0293–DEF.

References

Appendix A. An example appendix

Authors including an appendix section should do so after References section. Multiple appendices should all have headings in the style used above. They will automatically be ordered A, B, C etc.

Appendix A.1. Example of a sub-heading within an appendix

There is also the option to include a subheading within the Appendix if you wish.