Forecasting Research

Forecasting Research Division
Technical Report No. 109

Experiments with the preconditioning in the 2DVAR model

by

Andrew Lorenc and Anne Griffith

25 August 1994

Meteorological Office
London Road
Bracknell
Berkshire
RG12 2SZ
United Kingdom

Forecasting Research Division
Technical Report No. 109

Experiments with the preconditioning in the 2DVAR model.

Andrew Lorenc and Anne Griffith[1]

25 August 1994

Forecasting Research
Meteorological Office
London Road
Bracknell
RG12 2SZ

[1] Anne Griffith is a PhD student at the Reading University Mathematics Department, sponsored by the Met Office under the CASE Studentship scheme.

1 Introduction

The 2DVAR program (Shurlock & Lorenc 1994) was developed as a prototype to test critical aspects in the design of an operational variational scheme for the Met Office. This report describes experiments using the 2DVAR system to test the preconditioning using a horizontal filter.

The preconditioning currently used in the 2DVAR model ensures that in iterating to the minimum of the penalty function, very few penalty function evaluations are needed when a small number of observations are used in the assimilation. However, when the number of observations is increased significantly to thousands, the number of penalty function evaluations also increases. The aim of the experiments described here is to investigate whether fewer penalty function evaluations would be needed if the inner loop filters were omitted, i.e. if there was no preconditioning. We know that when there is no preconditioning, the number of penalty function evaluations required for the minimization of the penalty function actually decreases when more observations are used. These experiments, therefore, will investigate how many observations can be used before the preconditioning is no longer speeding up the convergence of the descent algorithm.

2 Theory

The "standard" formulation of variational analysis (Lorenc 1986) is - find the model state $\boldsymbol{x}$ which minimises a penalty (J) made up from a background term (J_b) and an observational term (J_o):

$$J(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}_b - \boldsymbol{x})^T \mathbf{B}^{-1}(\boldsymbol{x}_b - \boldsymbol{x}) + \frac{1}{2}(\boldsymbol{y}_o - \boldsymbol{y})^T (\mathbf{O}+\mathbf{F})^{-1}(\boldsymbol{y}_o - \boldsymbol{y}) \tag{1}$$

where $\boldsymbol{x}_b$ is a prior (background) estimate of $\boldsymbol{x}$, with error covariance $\mathbf{B}$, $\boldsymbol{y}_o$ is a vector of observed values, with error covariance $\mathbf{O}$, and $\boldsymbol{y}$ is a prediction of the observed values, given by:

$$\boldsymbol{y} = K(\boldsymbol{x})$$

$\mathbf{F}$ is the error covariance in the "generalised interpolation" K.

For the practical solution of this problem we make two transformations.

Firstly, we solve instead for a model perturbation $\boldsymbol{w}'$, which may be at lower resolution than

$\boldsymbol{x}$. That is, we find the perturbation model state $\boldsymbol{w}'$ which minimises:

$$J(\boldsymbol{w}') = \tfrac{1}{2}\left(\boldsymbol{w}'_b - \boldsymbol{w}'\right)^T \mathbf{B}_w^{-1}\left(\boldsymbol{w}'_b - \boldsymbol{w}'\right) + \tfrac{1}{2}\left(\boldsymbol{y}_o - \boldsymbol{y}\right)^T (\mathbf{O}+\mathbf{F})^{-1}\left(\boldsymbol{y}_o - \boldsymbol{y}\right) \qquad \textbf{(3)}$$

where we use interpolation G to transform the background $\boldsymbol{x}_b$, and the outer-loop estimate $\boldsymbol{x}_{ol}$, to the lower resolution of $\boldsymbol{w}'$:

$$\boldsymbol{w}'_b = G(\boldsymbol{x}_b) - G(\boldsymbol{x}_{ol}) \qquad \textbf{(4)}$$

$\boldsymbol{y}$, the prediction of the observed values, is now given by:

$$\boldsymbol{y} = K(\boldsymbol{x}_{ol}) + \tilde{K}(\boldsymbol{w}') \qquad \textbf{(5)}$$

This transformation is based on the belief that $\boldsymbol{x}_{ol} + \mathbf{H}\boldsymbol{w}'$ (where the interpolation $\mathbf{H}$ transforms from the low resolution of $\boldsymbol{w}'$ to that of $\boldsymbol{x}$), will be a good approximation to the $\boldsymbol{x}$ which minimises (1). It is possible to iterate this correction process for $\boldsymbol{x}$, outside of the minimisation iteration which finds $\boldsymbol{w}'$. We use the suffix $_{ol}$ to denote the current outer-loop estimate for $\boldsymbol{x}$.

Secondly, we transform to a variable $\boldsymbol{v}$ designed to improve the conditioning of the Hessian matrix in the minimisation process. The Hessian is a matrix of second order partial derivatives with respect to the control variables. e.g. for (1) the Hessian is defined as:

$$\left(\frac{\partial^2 J}{\partial \boldsymbol{x}^2}\right) = \begin{pmatrix} \dfrac{\partial^2 J}{\partial x_1 \partial x_1} & \dfrac{\partial^2 J}{\partial x_1 \partial x_2} & \cdots & \dfrac{\partial^2 J}{\partial x_1 \partial x_{n_x}} \\ \dfrac{\partial^2 J}{\partial x_2 \partial x_1} & \dfrac{\partial^2 J}{\partial x_2 \partial x_2} & \cdots & \dfrac{\partial^2 J}{\partial x_2 \partial x_{n_x}} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 J}{\partial x_{n_x} \partial x_1} & \dfrac{\partial^2 J}{\partial x_{n_x} \partial x_2} & \cdots & \dfrac{\partial^2 J}{\partial x_{n_x} \partial x_{n_x}} \end{pmatrix} \qquad \textbf{(6)}$$

For (3), if $\tilde{K}$ is linear, the Hessian is given by

$$\left(\frac{\partial^2 J}{\partial \boldsymbol{w}'^2}\right) = \mathbf{B}_w^{-1} + \tilde{K}^T (\mathbf{O}+\mathbf{F})^{-1} \tilde{K} \qquad \textbf{(7)}$$

The generalised interpolation $\tilde{K}$ in the second term in (7) depends on the positions of the observations being used. It is hard to analyse its conditioning in a general way, so we concentrate on the first term, which depends on the background error covariance. It has been observed that the errors in $\boldsymbol{x}_b$ are usually balanced, and smooth. This means that balanced and smooth modes will correspond to small eigenvalues of $\mathbf{B}_w^{-1}$, while imbalanced, or rough modes will correspond to large eigenvalues. This large range of eigenvalues means that $\mathbf{B}_w^{-1}$ is ill-conditioned.

To alleviate this ill-conditioning, we use a filter $\mathbf{U}$ designed to reduce the power in unbalanced or rough modes, and its inverse $\mathbf{T}$. We design these such that, approximately:

$$\begin{aligned} \mathbf{B}_w^{-1} &\approx \mathbf{T}^T\mathbf{T} \\ \mathbf{B}_w &\approx \mathbf{U}\mathbf{U}^T \end{aligned} \tag{8}$$

Then, defining a new control variable $\boldsymbol{v}$ such that

$$\boldsymbol{w}' = \mathbf{U}\boldsymbol{v} \tag{9}$$

our transformed variational problem is to find the $\boldsymbol{v}$ which minimises

$$J(\boldsymbol{v}) = \tfrac{1}{2}(\boldsymbol{v}_b-\boldsymbol{v})^T\mathbf{B}_v^{-1}(\boldsymbol{v}_b-\boldsymbol{v}) + \tfrac{1}{2}(\boldsymbol{y}_o-\boldsymbol{y})^T(\mathbf{O}+\mathbf{F})^{-1}(\boldsymbol{y}_o-\boldsymbol{y}) \tag{10}$$

where

$$\boldsymbol{v}_b = \mathbf{T}\,\boldsymbol{w}'_b \tag{11}$$

and the estimates of the observations are now given by:

$$\boldsymbol{y} = K(\boldsymbol{x}_{ol}) + \tilde{K}(\mathbf{U}\boldsymbol{v}) \tag{12}$$

The Hessian of (10) is given by:

$$\left(\frac{\partial^2 J}{\partial v^2}\right) = \mathbf{B}_v^{-1} + \mathbf{U}^T\tilde{K}^T(\mathbf{O}+\mathbf{F})^{-1}\tilde{K}\mathbf{U} \tag{13}$$

Because of (8)

$$\mathbf{B}_v^{-1} \approx \mathbf{I} \tag{14}$$

so the first term in (13) is much better conditioned than in (7).

3 Implementation in 2DVAR

2DVAR is designed to test the above for a single two-dimensional field on a sphere, either globally, or for a rectangular limited area. For a single field there is no concept of "balance"; the only prior knowledge about background errors is that they are likely to be smooth. Thus the transformation **U** is implemented (in subroutine VtoW) using a horizontal filter. In the default version we use one pass in each direction of a recursive filter. It is assumed that (8) and (14) are exactly true. The $\boldsymbol{v}$ which minimises (10) is found using the NAG conjugate gradient descent algorithm E04DGF.

The program by default (mode_v=0) finds the minimum of (10). This is referred to below as the method with preconditioning. For limited area grids there is an option (modev=1) to instead find the minimum of (3). This is referred to below as the method without preconditioning.

The recursive filter is described in Appendix A of Lorenc 1992. A one-dimensional wave of length $2\pi/k$ is damped by each pass by a factor given by:

$$S(k) = \frac{1}{1 + \frac{\alpha}{(1-\alpha)^2}\left[2\sin\left(\frac{k\,\delta x}{2}\right)\right]^2} \qquad \textbf{(15)}$$

where the filter coefficient α is given by:

$$\begin{aligned} \alpha &= 1 + E - \sqrt{E(E+2)} \\ E &= 2\,N_{pass}\,\delta x^2/4s^2 \end{aligned} \qquad \textbf{(16)}$$

where s is the horizontal correlation scale, δx is the grid-length, and $N_{pass}=1$ in the default version. Note that the factor 2 multiplying N_{pass} arises because **U** appears twice in (8).

Using (15) we can calculate the eigenvalues of $\mathbf{B}_w^{-1}$. Figure 1 shows a plot of the relative[2] eigenvalues for the configuration used in most of the tests described below, for a 31by31 grid with a gridlength of 100km, and a horizontal correlation scale of 400km. From the ratio of

[2] The full inverse covariance also contains a scaling dependent on the background error variance, and the horizontal scale.

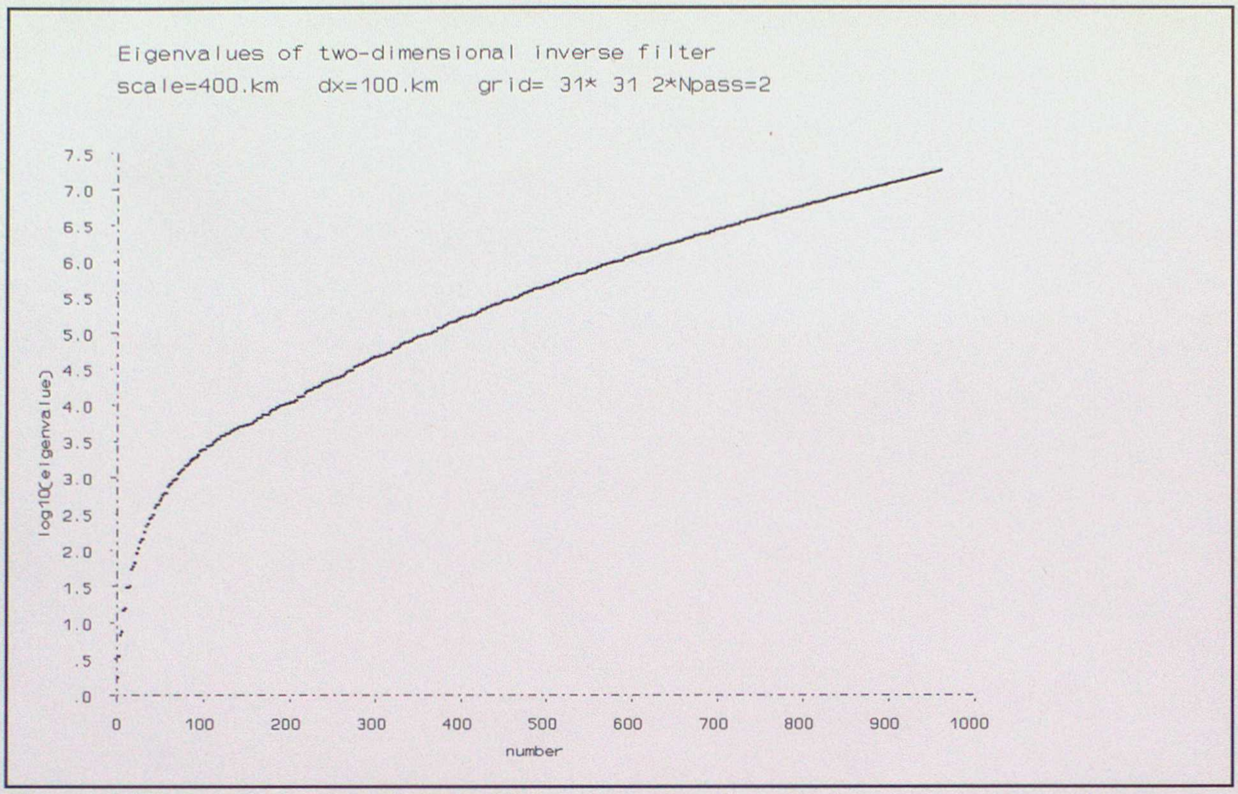

Figure 1 Eigenvalues of an inverse filter proportional to the inverse error covariance.

the largest and smallest eigenvalues we can see the contribution of $\mathbf{B}_w^{-1}$ to the condition number of the Hessian. For our example this is larger than 10^7, and it increases rapidly as the grid-length is reduced below 100km. So if J_b dominates, the problem is badly ill-conditioned, and we can expect convergence of minimisation routines without preconditioning to be slow. This is demonstrated in a simple experiment with a single observation, with value $y_o=1.0$, at the centre of the square grid area. The background field, and first-guess, is zero everywhere. For this case the preconditioning is perfect, so the descent algorithm converges in one iteration (which requires two penalty evaluations). The method without preconditioning however needs several thousand penalty evaluations before nearing convergence. (Convergence is judged by the approach of the norm of the gradient of J to zero), as shown in figure 2.

Despite the many iterations, the method without preconditioning has still not converged to the exact solution, as can be seen in figure 3. We verified that the method with preconditioning was indeed giving the exact solution by using it as first-guess for another descent iteration. Neither method could improve on it.

4 Experiments to test preconditioning with more observations

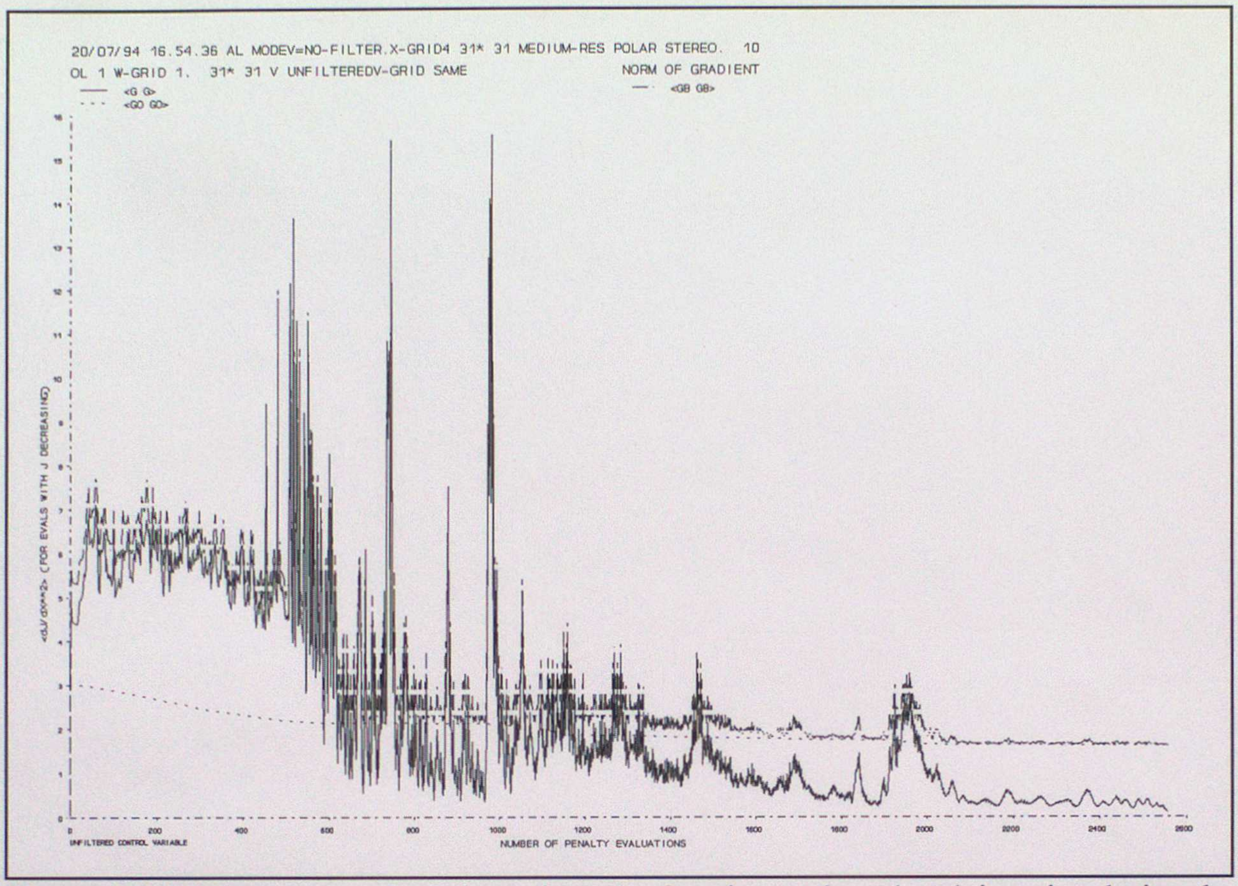

Figure 2 Solid line - norm of gradient of penalty function evaluated each iteration during the iterative minimisation (at convergence it should be zero). Dashed line & dotted line - norm of contribution to gradient from Jb and Jo terms.

4.1 Method

For the investigations described here, several runs of the program "almodev.pre" were carried out using different numbers of observations and different grid resolutions. This program is a version of the 2DVAR model which runs the same experiment using both methods; once with the preconditioning and once without. The background field errors were taken to be random, and the initial guess of the background was set to zero. A few runs were also done where the background was initially set to the analytical solution, or "truth field", but this made little difference to the results. The observations were randomly generated.

In each case the root mean square differences (RMS values) between the analysis and the truth field were compared for the two methods, as a means of assessing their relative accuracy. The maximum and minimum values of each analysis were also recorded, as a check that the two methods were giving a similar analysis. In addition, the values of the penalty functions J, J_b and J_o and their gradients at the end of each penalty function evaluation were recorded for both the methods.

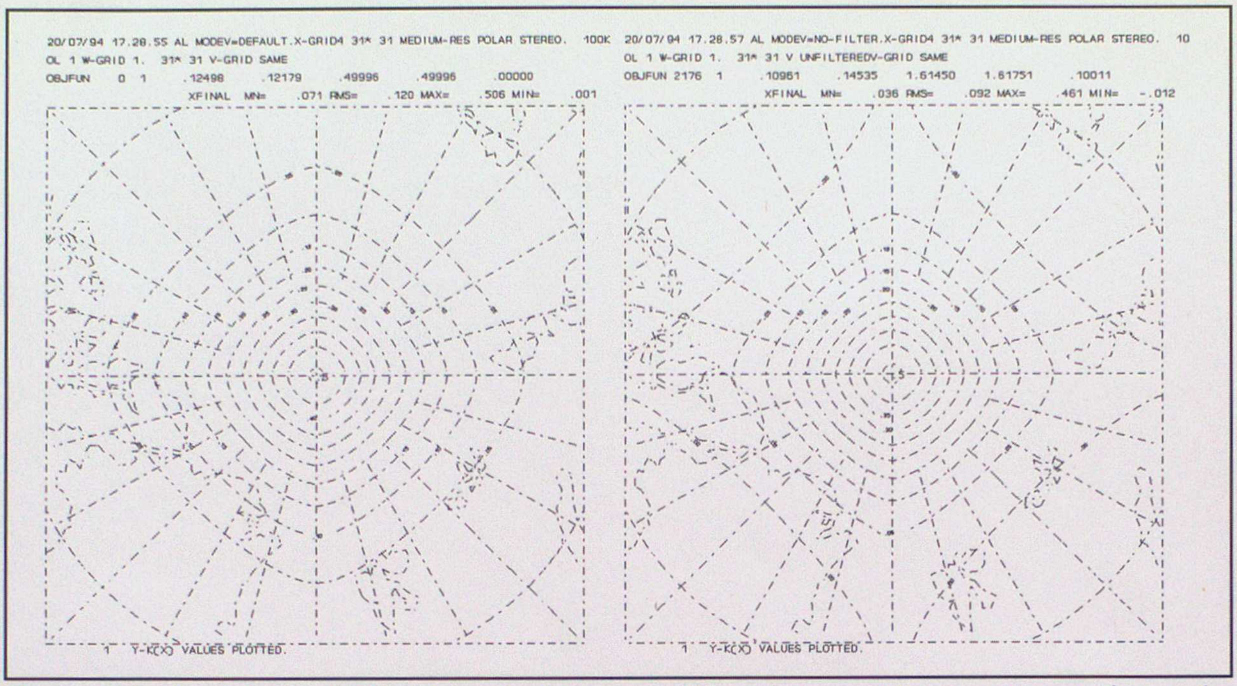

Figure 3 Left - solution field obtained with preconditioned method for one observation at the pole. Right - solution field obtained with the method without preconditioning.

The main test of the effectiveness of the preconditioning is of course the number of penalty function evaluations needed with and without the preconditioning. These values for both methods were recorded in all the runs carried out with a different number of observations, and the runs at different grid resolutions.

4.2 Results

4.2.1 Comparing the analyses for both methods

The differences in the RMS values were very small, generally less than 0.01, although slightly higher than this in the runs at high resolution with many observations. As expected, the RMS values decreased as the number of observations used was increased, with a slower decrease at higher resolutions.

Comparing the maximum and minimum values of the analysis for the two methods indicated agreement between the two analyses. For each resolution the maximum values and the minimum values were generally consistent for all the runs, with a few exceptions when smaller numbers of observations were used, particularly at high resolution. In these examples it is likely that the minimisation routine has stopped without reaching true convergence.

These results check that the same analysis is derived with or without the preconditioning, provided convergence is reached.

4.2.2 Comparing the penalty functions and their gradients

The final value of the penalty function J each time it is evaluated is very similar for both methods. J_b and J_o also agree for both methods, which checks that both methods give the observations the same weight relative to the background.

The gradients for the two methods cannot be directly compared, since they are with respect to different variables. The gradient of J is much higher when no preconditioning is used. There was also some difference between the two methods in the gradients of J_b and J_o, when the preconditioning is used grad J_o is always greater than or equal to grad J_b, and when the preconditioning is omitted, it is the other way round.

4.2.3 Comparing the number of penalty function evaluations needed

The best way to assess the merits of using the preconditioning is to compare the number of penalty function evaluations needed with and without it. These values depend on the number of observations, and on the grid resolution. The results from these experiments make it possible to assess which method is better for a given number of observations.

resolution & method		number of observations											
		100	500	1,000	5,000	10,000	15,000	20,000	50,000	55,000	60,000	75,000	100 000
med	1	15		31	51	57	65	47					
	2	1113		339	107	71	54	**47**	**89**				
med high	1	13	19	25	37	45			47	47	67	65	
	2	1957	1063	747	245	147			61	59	**59**	**51**	
high	1	11	15	19	29	39							69
	2	3227	1755	1243	415	255							83

Table showing number of penalty evaluations required to reach the minimisation routine's convergence criterion, for method 1 - with preconditioning, and method 2 - without preconditioning.

In the medium resolution runs, fewer evaluations are needed for the method with preconditioning when up to 15,000 observations are used, but when 20,000 observations or more are used, the method without preconditioning is better. In the medium-high resolution, the point at which the preconditioning becomes ineffective in reducing the number of iterations is for somewhere between 55 and 60,000 observations. However, in the high resolution runs, the method with preconditioning was still more efficient even when 100,000 observations were used.

5 Conclusions

The results of these experiments show that preconditioning effectively reduces the number of iterations needed to minimize the penalty function, unless a large number of observations are used. For higher grid resolutions the preconditioning is still effective when an even greater number of observations is used. The results indicate, for the particular cases investigated, how many observations can be included before it becomes more effective to omit the preconditioning.

In the light of these results, given the varying densities of observations which occur in practice, the method with preconditioning is recommended for practical implementation.

6 References

Lorenc, A.C. 1986
"Analysis methods for numerical weather prediction." *Quart. J. Roy. Met. Soc.,* **112**, 1177-1194.

Lorenc, Andrew 1992
"Iterative analysis using covariance functions and filters" *Quart. J. Roy. Met. Soc.*, **118**, 569-591

Shurlock, Jon and Andrew Lorenc, 1994
"Two-dimensional grid-point variational analysis." FR Tech Report 108