

U. S. DEPARTMENT OF COMMERCE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NATIONAL WEATHER SERVICE
NATIONAL METEOROLOGICAL CENTER

OFFICE NOTE 304

A KALMAN FILTER FOR A TWO-DIMENSIONAL SHALLOW-WATER MODEL:
FORMULATION AND PRELIMINARY EXPERIMENTS

DAVID F. PARRISH
SHORT-RANGE MODELING BRANCH
DEVELOPMENT DIVISION

and

STEPHEN E. COHN*
COURANT INSTITUTE OF MATHEMATICAL SCIENCES
NEW YORK UNIVERSITY
NEW YORK, N. Y. 10012

FEBRUARY 1985

THIS IS AN UNREVIEWED MANUSCRIPT, PRIMARILY INTENDED FOR
INFORMAL EXCHANGE OF INFORMATION AMONG NMC STAFF MEMBERS.

*Supported in part by NOAA Grant NA84AA-D-00018, by a Control Data Corp. PACER Fellowship, by NASA Grants NAG-5-341 and NSG-5130, by NSF Grant INT-8314934, and by Grant 1.01.10.021-83 of the Brazilian agency CNPq.

A KALMAN FILTER FOR A TWO-DIMENSIONAL SHALLOW-WATER MODEL:
FORMULATION AND PRELIMINARY EXPERIMENTS

Contents

	<u>Page</u>
Abstract	iii
I. Introduction	1
II. The Kalman Filter	4
III. Computational Considerations in the Forecast Step	8
IV. Computational Considerations in the Analysis Step	14
V. The Shallow-Water Model	21
VI. Numerical Experiments	25
VII. Concluding Remarks	33
Appendix A. Diagonalwise Matrix Triple Product	34
Appendix B. Serial Processing of Observations	40
Appendix C. Finite-Difference Scheme	43
References	48
Tables	50
Figures	53

ABSTRACT

The Kalman filter is a data assimilation scheme which, unlike currently operational methods such as optimal interpolation (OI), makes systematic use of forecast model dynamics in order to accurately determine the evolution of the forecast error covariance matrix. Previous studies with a simple one-dimensional model indicated that the Kalman filter, if applied operationally, would yield analyzed and subsequent forecast fields superior to those resulting from OI. These studies did not address the enormous computational burden that the Kalman filter would appear to pose if applied in an operational setting, to an actual numerical weather prediction (NWP) model.

In this report we introduce a number of techniques which, taken together, reduce dramatically the computational complexity of the Kalman filter. The new filter algorithm gains its efficiency, in part, by taking explicit advantage of the fact that forecast errors are significantly correlated only over rather small distances. It also utilizes fully the vector-processing capabilities of the CYBER 205 computer. Part of the overall method is an analysis algorithm which processes observations one at a time, i.e., it loops on observations rather than analysis grid points, thereby eliminating both the necessity of matrix inversions and the necessity of restarting the entire analysis to accommodate late-arriving observations. This analysis algorithm would be useful in OI schemes as well as in the Kalman filter.

We apply the Kalman filter to a two-dimensional shallow-water channel model. Numerical experiments demonstrate, first of all, that the filter is indeed computationally feasible in two dimensions. The results show also that actual forecast error correlations, which are computed exactly by the

Kalman filter differ markedly from the rather simple, homogeneous, correlations prescribed currently in the NMC OI analysis system. The experiments suggest a number of improvements to our computational approach, which should render the Kalman filter practical for operational data assimilation into fully three-dimensional NWP models.

I. Introduction

Data assimilation methods are designed to combine observations of the atmosphere at a given time with a forecast valid at that time, in such a way as to determine the best possible estimate of the atmospheric state. Fundamental considerations imply that, under rather mild assumptions, the statistically optimal estimate is obtained by linearly combining the set of observations and the forecast field, and weighting each according to its error covariance matrix. Thus, two key ingredients of optimal interpolation (OI) analysis schemes are the forecast and observation error covariances. Analysis accuracy depends strongly on the accuracy with which these covariances are specified. The forecast error covariance is usually the less accurately known of the two. It is specified in the OI system at NMC by assuming certain time- and space-independent forecast error correlation functions and by assuming a simple spatial dependence for the growth rate of forecast error variances.

Recent studies, summarized in Ghil et al. (1982) have suggested the Kalman filter as a potential alternative to the conventional OI methodology. The Kalman filter differs from OI primarily in that it determines the forecast error covariance accurately, by using the forecast model itself to evolve the covariance in time. This difference is crucial. The Kalman filter properly accounts for propagation of information between regions of different data density and quality, thus leading to superior analyses and subsequent forecasts. On the other hand, the correct evolution of the forecast error covariance given by the Kalman filter, would appear to present an enormous computational burden for actual numerical weather prediction (NWP) models. Previous studies have been carried out using simple one-dimensional models.

In this report we introduce a number of techniques which, taken together, reduce dramatically the computational complexity of the Kalman filter. We

apply the new filter algorithm to assimilate data into a two-dimensional shallow-water channel model. The experiments demonstrate, first of all, that the computation is indeed feasible for a two-dimensional model. Second, they both confirm and extend the earlier, one-dimensional results. In particular, they show clearly that actual forecast error correlations differ markedly from the rather simple, homogeneous, correlations prescribed currently in the OI system at NMC. Third, by simulating different data distributions, the experiments show in a precise and quantitative way which distributions are the most effective at reducing overall forecast error. The Kalman filter is ideal for carrying out such observing system simulation experiments (OSSE) because it determines forecast error variances accurately.

The key idea we have used in reducing the computational complexity of the Kalman filter is to calculate only those elements of the covariance matrix which differ significantly from zero, and we have organized our algorithm around this approach. Since covariances tend to zero with increasing distance, this means that we calculate and store only a number of diagonals of the forecast error covariance matrix, rather than the entire matrix. Since the forecast model dynamics must be applied repetitively to these diagonals, we also increase computational efficiency by explicitly formulating the forecast model as a matrix, whose elements need be calculated only once. This matrix also consists of only a small number of nonzero diagonals, and its action on the forecast error covariance matrix is calculated by an algorithm for diagonalwise matrix multiplication which was designed for efficient computation on the CYBER 205 vector processor at NMC.

Our Kalman filter scheme also implements an algorithm which processes observations serially, i.e., one at a time: the algorithm loops over the observations, rather than analysis grid points. This algorithm eliminates

both the necessity of matrix inversion and the necessity of restarting the entire analysis to accomodate late-arriving observations. This analysis algorithm would be useful in OI schemes as well as in the Kalman filter.

After reviewing the essentials of Kalman filtering in Section II, we describe our method for evolving the forecast error covariance matrix in Section III. The serial observation-processing algorithm is detailed in Section IV. The shallow-water model is formulated in Section V and the numerical results are presented in Section VI. Concluding remarks appear in Section VII. This paper is meant, in part, as a technical report, and a number of the technical details are explained in three appendices.

II. The Kalman Filter

In this section we describe the essentials of the Kalman filter. More detailed information can be found in standard texts such as Gelb (1974) and Jazwinski (1970), and in Ghil et al. (1981) and Cohn (1982).

Suppose we are given a numerical forecast model

$$w_k^f = \Psi_{k-1} w_{k-1}^a \quad (2.1)$$

Here w_k^f is the forecast vector at time step k and w_{k-1}^a is the analysis vector at the previous time step; we simply define $w_{k-1}^a \equiv w_{k-1}^f$ if no observations were analyzed then. Both forecast and analysis vectors have length n , the number of degrees of freedom of the model. In a finite-difference model, say, this n is the number of grid points times the number of dependent variables. The $n \times n$ dynamics matrix Ψ may depend on time, $\Psi = \Psi_{k-1}$, and reflecting model nonlinearity, depends on w_{k-1}^a as well, $\Psi = \Psi_{k-1}(w_{k-1}^a)$.

Equation (2.1) simply sets up a notation in which we will view one time step of a forecast as being a matrix-vector multiplication. The trivial observation that one time step can be viewed as such will actually be one key to our computational simplification of the Kalman filter.

Since the objective of an analysis system is to estimate the true atmospheric state, whose evolution is modeled imperfectly by (2.1), we assume that the true state, denoted by w_k^t , evolves instead according to

$$w_k^t = \Psi_{k-1}^t w_{k-1}^t + r_{k-1}^t \quad (2.2a)$$

where $\Psi_{k-1}^t \equiv \Psi_{k-1}(w_{k-1}^t)$ and where the model error, r_k^t , is a random n -vector which is white in time and has mean zero and covariance matrix Q_k :

$$E r_k^t = 0, \quad E (r_k^t)(r_l^t)^T = Q_k \delta_{kl} \quad (2.2b.c)$$

Here the symbol E denotes the expectation, or ensemble-averaging operator, the superscript T denotes the transpose, and δ_{kl} is the Kronecker delta, $\delta_{kl} = 0$ for $k \neq l$ and $\delta_{kk} = 1$. Equations (2.2) actually define what we mean by the true atmospheric state: it is the n -vector w_k^* .

We assume further that meteorological observations w_k^o are in fact linear combinations of elements of the true state vector w_k^* , contaminated by white noise:

$$w_k^o = H_k w_k^* + l_k^o \quad (2.3a)$$

The length of the observation vector w_k^o is the number of observations p available at time k , $p = p_k$. The observation matrix H_k accounts for interpolation between model grid points and observation locations, and for any necessary conversion between observed variables and state variables, e.g., between satellite radiances and temperatures. The observation error, l_k^o , is a random p -vector which is assumed to have mean zero and covariance matrix R_k , and is assumed to be uncorrelated with the model error:

$$E l_k^o = 0, \quad E (l_k^o) (l_k^o)^T = R_k \delta_{kl}, \quad (2.3b,c)$$

$$E (l_k^o) (l_l^*)^T = 0. \quad (2.3d)$$

Under the stated assumptions, the extended Kalman filter (EKF) is the data assimilation system

$$w_k^f = \Psi_{k-1} w_{k-1}^a, \quad (2.4a)$$

$$P_k^f = \Psi_{k-1} P_{k-1}^a \Psi_{k-1}^T + Q_{k-1}, \quad (2.4b)$$

$$K_k = P_k^f H_k^T (H_k P_k^f H_k^T + R_k)^{-1} \quad (2.4c)$$

$$P_k^a = (I - K_k H_k) P_k^f, \quad (2.4d)$$

$$w_k^a = w_k^f + K_k (w_k^o - H_k w_k^f), \quad (2.4e)$$

for $k=1,2,3,\dots$. Equations (2.4a,b) are the state forecast and estimation error covariance forecast, respectively. Equation (2.4c) gives the Kalman gain matrix K_k which is the weight with which observations and forecast are combined in (2.4e) to yield the analyzed field w_k^a . Equation (2.4d) gives the matrix P_k^a which is needed at the next time step in (2.4b).

In case the dynamics are linear, i.e., if Ψ does not depend on w , then the matrices P_k^f and P_k^a in (2.4b-d) are precisely the forecast and analysis error covariance matrices, i.e.,

$$P_k^{f,a} = E (w_k^{f,a} - w_k^t) (w_k^{f,a} - w_k^t)^T. \quad (2.5a,b)$$

In this linear case, equations (2.4a-e) are known simply as the Kalman, or Kalman-Bucy filter, and they represent a data assimilation scheme which is optimal in the sense of minimizing the analysis error and subsequent forecast error at every grid point, given all current and previous observational information. If in addition the model error b_k^t and observational error b_k^o are Gaussian, the Kalman filter is optimal in an even broader mathematical sense (e.g., Jazwinski, 1970).

In the nonlinear case, the EKF (2.4a-e) is only a first-order approximation to the optimal filter. From a computational standpoint, however, it is clear that the EKF and the Kalman filter are nearly identical, and most nearly-optimal alternative nonlinear filters are also similar to the Kalman filter. Since our purpose in this paper is to study the computational issues in optimal filtering, equations (2.4a-e) are the primary consideration.

We also point out that most of the assumptions we have made have analogues in the formulation of present-day optimal interpolation (OI) analysis schemes (e.g., Bergman, 1979; Lorenc, 1981). Many of these assumptions can be relaxed. For example, biases can be allowed in the model and observational errors (2.2b, 2.3b), correlations between these errors (2.3d) can be accounted for, and the errors need not even be white (2.2c, 2.3c). Each of these generalizations is accounted for by additional terms in the corresponding optimal filter (Jazwinski, 1970, pp. 158-159 and 209-218). Again, since the main computational issues arise already in (2.4a-e), we shall not pursue these or other generalizations here.

In the present study we will take the model and observational error covariance matrices, Q_h and R_h , to be known. In an operational setting, one would want to actually estimate Q_h and R_h . Phillips (1982) has pointed out, for example, that accurate knowledge of these matrices can be crucial to the successful performance of the Kalman filter. Dee *et al.* (1984, 1985) show that it is possible to estimate Q_h and R_h during the assimilation process itself, at the cost of calculating a number of additional Kalman filters. Once again, even when Q_h and R_h must be estimated, equations (2.4a-e) encompass the main computational burden.

In addition to assuming knowledge of Q_h and R_h , we will take the dynamics to be linear, so that the filter (2.4a-e) is indeed optimal and all its ingredients are known. Thus we have constructed an environment in which many questions regarding data assimilation can be answered in a precise way.

III. Computational Considerations in the Forecast Step

The fundamental difference between our formulation of the data assimilation (filtering) problem in Section II and the way in which OI schemes are formulated is that we have stated in a precise and convenient way what is meant by the true atmospheric state: eqs. (2.2). This statement, combined with the forecast model (2.1) and definitions (2.5a,b) leads, in the linear case, immediately to equation (2.4b) which gives the evolution of the forecast error covariance matrix P_h^f :

$$\begin{aligned}
 P_h^f &= E(w_h^f - w_h^t)(w_h^f - w_h^t)^T \\
 &= E[\Psi_{h-1}(w_{h-1}^a - w_{h-1}^t) - l_{h-1}^t][\Psi_{h-1}(w_{h-1}^a - w_{h-1}^t) - l_{h-1}^t]^T \\
 &= \Psi_{h-1} [E(w_{h-1}^a - w_{h-1}^t)(w_{h-1}^a - w_{h-1}^t)^T] \Psi_{h-1}^T + E(l_{h-1}^t)(l_{h-1}^t)^T \\
 &= \Psi_{h-1} P_{h-1} \Psi_{h-1}^T + Q_{h-1}
 \end{aligned}$$

In fact, it is the appearance of eq. (2.4b) which distinguishes the Kalman filter (2.4a-e) from OI schemes: in the Kalman filter, the forecast error covariance matrix is known precisely, whereas in OI schemes it is approximated by means of several ad hoc assumptions (e.g., Cohn and Morone, 1984; Morone and Cohn, 1984, 1985). This difference is crucial. The evolution equation (2.4b) determines, for example, how errors propagate between data-sparse and data-dense regions, and is the basis for the optimality properties possessed by the Kalman filter.

The "analysis" part of the Kalman filter, (2.4c-e), is identical to OI: given the forecast error covariance matrix, the Kalman filter and OI are essentially the same. Equations (2.4c-e) are merely a rather shorthand way of writing the usual OI analysis equations (e.g., Cohn et al., 1981). In

operational implementation of OI, though, only the diagonal elements of P_k^a , that is, the analysis error variances, are computed. Also, data selection procedures which reduce the computational effort in calculating the gain matrix (2.4c) are invoked. An efficient algorithm for implementing the analysis equations (2.4c-e) will be described in Section IV.

The appearance of equation (2.4b) for the evolution of the forecast error covariance matrix not only accounts for the optimality properties of the Kalman filter, but also represents the main computational burden of the filter. In the current section we discuss means of reducing this burden. Our remarks here are of a rather general nature: they apply to Kalman filtering in multiple space dimensions. The particular model to which we have applied these techniques is not discussed until Section V.

The forecast error covariance equation (2.4b) is the most expensive calculation in the Kalman filter because it involves multiplication of $n \times n$ matrices; the state forecast (2.4a), by contrast, involves only matrix-vector multiplication. In a brute-force approach, the estimation error covariance forecast would therefore require $2n$ times as much work as the state forecast. For $n \approx 10^5$ or 10^6 , typical of current numerical weather prediction (NWP) models, this is clearly prohibitive. We have developed a number of strategies to reduce this burden to manageable levels.

First, we observe that significant savings can be realized by explicitly computing and storing the elements of the dynamics matrix Ψ_k . This is not, of course, what is done in the usual NWP code: Ψ_k is available only as an operator, or procedure, for calculating w_k^f given w_{k-1}^a . In the covariance equation (2.4b), however, the same dynamics are needed repeatedly, $2n$ times in fact, so it is best to calculate them once and for all. Doing so reduces the computational expense of (2.4b) by about an order of magnitude:

most of the expense in conventional NWP codes lies in what can be thought of as computing the elements of the dynamics matrix Ψ_h , rather than in performing the matrix-vector multiplication in (2.4a).

The cost, of course, is the additional storage. This may pose a problem in an operational setting, although we point out that (2.4b) could be carried out at lower resolution than (2.4a), reducing both storage and time requirements. One should also keep in mind that computers with about 100 times the storage and speed of the CYBER 205 are expected to be commercially available in two years.

Second, we point out that the dynamics matrix Ψ , as well as the covariance matrices $P^{f,a}$, Q and R should be stored by diagonals rather than by the conventional column storage, and that all matrix-matrix and matrix-vector operations should be carried out by diagonals. This point is crucial for reducing both storage and execution-time requirements. If the dynamics are based on a finite-difference scheme, then Ψ has a banded structure: a small number of diagonals of Ψ contain nonzero elements, and all the remaining diagonals are zero. It is natural, then, to store only the nonzero diagonals, each as a vector, and to formulate the matrix-vector operation (2.4a) and matrix-matrix operations (2.4b) in terms of operations with these vectors. Our choice of dynamics matrix Ψ will be described in detail in Section V.

Madsen et al. (1976) have shown how to multiply a matrix by either a vector or another matrix, using operations only on the diagonals of the matrices. We show in Appendix A how to evaluate the matrix triple product in (2.4a) by diagonalwise operations, in such a way that, essentially, P_h^f is overwritten on P_{h-1}^a . Here we only note that diagonalwise matrix operations are very efficient on the CYBER 205 vector processor: the diagonals (vectors) are very

long, they are few in number, and they contain few, if any, zeros.

This brings us to the third, and most important point concerning implementation of (2.4b). It is well known that forecast error correlations (and hence covariances) tend rapidly to zero with distance. That is, the matrices $P^{f,a}$ have a banded structure, similar to Ψ , in which only a small number of diagonals differ significantly from zero. The remaining diagonals, in fact most of $P^{f,a}$, can be considered to be exactly zero. Thus we introduce what we refer to as the banded approximation in our calculation of (2.4b). Data selection procedures used in OI schemes are similar in spirit to this banded approximation.

The banded approximation works in the following way. For a given experiment, we select a bandwidth \underline{b} , which is the number of grid points away from each base point over which covariances will be calculated. Only those diagonals of P_{k-1}^a within this distance are stored. The operation $\Psi_{k-1} P_{k-1}^a \Psi_{k-1}^T$ acts only on the stored diagonals of P_{k-1}^a : the rest are assumed to be zero and do not enter the computation (Appendix A). This operation also introduces additional diagonals in principle, but they are not calculated: they are assumed to be negligibly small. The diagonals of Q_{k-1} within bandwidth \underline{b} are then added to those of $\Psi_{k-1} P_{k-1}^a \Psi_{k-1}^T$, yielding P_k^f with bandwidth \underline{b} . The experiments reported in Section VI demonstrate the accuracy of the banded approximation for different choices of \underline{b} .

The banded approximation reduces dramatically both the execution time and storage requirements of the Kalman filter. We will calculate here the storage required as a function of bandwidth, and defer discussion of execution time to Section V. Our experience to date has been that storage poses more of a problem than execution time. All our computations are performed in central memory, in order to circumvent paging problems.

For bandwidth b , the number of grid points in a given coordinate direction which are correlated with each base point is $2b+1$, i.e., b points to each side of the base point plus the base point itself. In a d -dimensional problem, then, the number of correlated grid points is $(2b+1)^d$, assuming we calculate in squares ($d=2$) or cubes ($d=3$), rather than circles or spheres, say. In a problem with v dependent variables ($v=3$ for the shallow-water equations, and $d=2$), the number of nonzero elements in each row of $P^{f,a}$ is therefore $v(2b+1)^d$. The number of rows of $P^{f,a}$ is simply $n=Nv$, where N is the total number of grid points in the model (see remarks following eq. (2.1)). The total storage S required for $P^{f,a}$ under the banded approximation is therefore

$$S = Nv^2(2b+1)^d.$$

By contrast,

$$S = (Nv)^2$$

if the full $P^{f,a}$ were stored, without the banded approximation. In either case, in principle one could reduce the storage by a factor of almost 2 by taking advantage of the fact that P^f and P^a are symmetric matrices. In fact, the diagonalwise matrix multiplication algorithm (Appendix A) does calculate only the lower triangular part of P^f , and references only the lower triangular part of P^a . However, the algorithm does not allow the lower triangular part of P^f to be overwritten on that of P^a : storage is required, essentially, for (the nonzero diagonals of) one covariance matrix.

In Table I, we list the value of $S/N = v^2(2b+1)^d$, i.e., the storage required per grid point, as a function of b for the case $v=3$ and $d=2$. We also give $N = S[v^2(2b+1)^d]^{-1}$ for the same case, with $S = 3 \times 10^6$, as well as the maximum dimensions of a uniform square grid for this S , namely $\sqrt{N} \times \sqrt{N}$.

Currently there are 8×10^6 halfwords of memory on the CYBER 205 at NMC, of which 4×10^6 are accessible to the user. The last row of Table I gives the figures corresponding to the preceding rows, but for the case in which the full $P^{f,a}$ is stored.

The last row shows that a grid of only about 580 points, e.g., a 24 x 24 grid, is permitted if the banded approximation is not used. In fact, we were able to run full- $P^{f,a}$ experiments on a 24 x 24 grid, enough storage still being available for Ψ , H, Q and R, but not on a 25 x 25 grid. On the other hand, the table shows that a 64 x 64 grid is permissible for $b=4$, and 82 x 82 for $b=3$. Experiments reported in Section VI indicate that a bandwidth of 3 to 4 gives good accuracy, independently of resolution. The accuracy of the banded approximation could be improved even further, or the number of required diagonals decreased, by extrapolating to actually calculate nearby diagonals rather than setting them to zero.

As a final remark concerning storage of $P^{f,a}$, we note that a further reduction of storage by a factor of about 4 is possible if, instead of storing $P^{f,a}$, we store separately the forecast and analysis error variances $D^{f,a}$,

$$D^{f,a} = \text{diagonal part of } P^{f,a},$$

and correlation matrices $C^{f,a}$,

$$C^{f,a} = (D^{f,a})^{-1/2} P^{f,a} (D^{f,a})^{-1/2}.$$

Since the numerical values of correlations lie between -1.0 and +1.0 by definition, eight bits of storage per correlation, say, would probably allow adequate precision: entire halfwords (or fullwords) are not necessary. The cost of this approach, which we have not yet tried, would be somewhat increased execution time.

IV. Computational Considerations in the Analysis Step

The "analysis step" (2.4c-e) is generally less expensive in meteorological applications than the "forecast step" (2.4a,b) because at least one dimension of most of the matrices and vectors involved in the analysis step is p , the number of observations, which is usually much smaller than the number of state variables n : H_k is $p \times n$, K_k is $n \times p$ and R_k is $p \times p$. For global-scale analysis, the number of observations per time step is far smaller than n , and even if observations are grouped in six-hour windows, their number is much less than n . Even for the next generation of mesoscale models, which will assimilate wind-profiler data, one expects model resolution to exceed data quantity.

Still, the analysis step does represent a large calculation, and one must pay attention to the computational cost. The most expensive part of the analysis step would appear to be (2.4c), in which one must either invert the $p \times p$ matrix $H_k P_k^f H_k^T + R_k$ or, more simply, solve n linear systems with this matrix as the coefficient matrix. A remarkable fact about the analysis step is that it can be formulated as a computationally efficient algorithm which does not require inverting, or solving systems with, $p \times p$ matrices.

The intuitive idea is to view the observations at time step k as occurring in fact over a sequence of "time steps" over which no dynamics (2.4a,b) take place. One should therefore be able to apply the analysis step (2.4c-e) to each observation in turn, ending up with the same result as if the observations had been processed simultaneously. That is, performing p analyses according to (2.4c-e), one observation per analysis, should be, and in fact is, equivalent to performing a single analysis using all p observations. Such an algorithm is very efficient because the linear systems to be solved are now all scalar: $p=1$ for each analysis. This idea of processing observations

serially, i.e., one at a time, is a familiar one in the estimation literature and has been implemented in a number of ways (e.g., Bierman, 1977). We describe a very simple and rather standard implementation here.

Before describing the algorithm, we stress again that the Kalman analysis equations (2.4c-e) are identical to the OI analysis equations, given a forecast error covariance matrix P_k^+ . The serial observation-processing algorithm to be described would therefore be useful in OI schemes as well as in the Kalman filter. The algorithm is merely a convenient and efficient way of implementing the analysis equations; it is algebraically equivalent to (2.4c-e). The algorithm represents a significant departure from standard OI implementations, such as the OI system at NMC which loops over analysis grid points rather than observations. Serial processing of observations would be especially convenient in an operational setting, since observations are processed as they become available: there would be no need to start the analysis from scratch at later data cutoff times.

In formulating an analysis algorithm, one can and should take advantage of the particular structure of observation error covariances. For example, many pairs of meteorological observations have uncorrelated errors, such as observations from different instruments. The algorithm we describe takes explicit advantage of this fact, and gains efficiency the more uncorrelated observations there are. We begin by describing how the algorithm takes advantage of uncorrelated observations. Since all our discussion in this section concerns the analysis at a single time step k , we will omit k as a subscript: the observation model (2.3a,c) is written as

$$w^o = H w^x + l^o, \quad (4.1a)$$

$$E(l^o)(l^o)^T = R, \quad (4.1b)$$

and the analysis equations (2.4c-e) are written as

$$K = P^f H^T (H P^f H^T + R)^{-1}, \quad (4.2a)$$

$$P^a = (I - KH) P^f, \quad (4.2b)$$

$$w^a = w^f + K(w^o - H w^f). \quad (4.2c)$$

That many pairs of observations are uncorrelated means that R has a block-diagonal structure,

$$R = \begin{bmatrix} R_1 & & & \\ & R_2 & & \\ & & \ddots & \\ & & & R_J \end{bmatrix}, \quad (4.3a)$$

where each block R_j , $j=1,2,\dots,J$, is a $p_j \times p_j$ matrix and $p_1 + p_2 + \dots + p_J = p$, the number of observations at time step k . Thus there are J batches of observations, $w_1^o, w_2^o, \dots, w_J^o$, each w_j^o being a p_j -vector, such that different observations in the same batch may be correlated, but observations in different batches are not correlated. If we partition the full observation vector w^o and the observation matrix H as

$$w^o = \begin{bmatrix} w_1^o \\ w_2^o \\ \vdots \\ w_J^o \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_J \end{bmatrix}, \quad (4.3b,c)$$

where each H_j is a $p_j \times n$ matrix, and similarly partition the observational error vector b^o , then the observation model (4.1a,b) becomes simply

$$w_j^o = H_j w^t + b_j^o, \quad (4.4a)$$

$$E(b_j^o (b_j^o)^T) = R_j, \quad (4.4b)$$

for $j=1,2,\dots,J$.

Now, suppose we define

$$P_0 = P^f, \quad w_0 = w^f. \quad (4.5a,b)$$

The precise statement that observations may be processed serially is that if we calculate

$$K_j = P_{j-1} H_j^T (H_j P_{j-1} H_j^T + R_j)^{-1}, \quad (4.6a)$$

$$P_j = (I - K_j H_j) P_{j-1}, \quad (4.6b)$$

$$w_j = w_{j-1} + K_j (w_j^o - H_j w_{j-1}), \quad (4.6c)$$

for $j=1,2,\dots,J$, then in fact

$$P_J = P^a \quad \text{and} \quad w_J = w^a \quad (4.7a,b)$$

where w^a and P^a are the analysis vector and analysis error covariance matrix that would be obtained from (4.2a-c). Equations (4.6) represent far less computational effort than equations (4.2): the relevant dimension of the matrices in (4.6) is p_j , whereas in (4.2) it is p itself. In practice, the maximum number of intercorrelated observations, such as those from a single instrument, is far smaller than the total number of observations available. We prove the equivalence of (4.2) and (4.6) in Appendix B.

Equations (4.6) describe only one part of the serial processing algorithm: we have shown so far only how to process uncorrelated batches of data independently. The data within a batch are correlated and we must still show how to process them serially, i.e., how to carry out a single step j of (4.6). We will show, essentially, that one need only consider the case $p_j = 1$: one can "decorrelate" those observations which are correlated. Before showing how to do this, then, we first describe in detail how to implement (4.6)

when $p_j = 1$.

In case $p_j = 1$, i.e., batch j consists of a single observation uncorrelated with all the other observations, then R_j is a scalar, H_j is a row vector of length n and K_j is a column vector of length n . To conform with the convention that matrices, column vectors and scalars are generally denoted by upper-case Roman, lower-case Roman, and lower-case Greek letters, respectively, we write

$$\sigma_j^2 = R_j, \quad h_j = H_j^T, \quad k_j = K_j. \quad (4.8a,b,c)$$

Equations (4.6) then become

$$k_j = P_{j-1} h_j (h_j^T P_{j-1} h_j + \sigma_j^2)^{-1}, \quad (4.9a)$$

$$P_j = (I - k_j h_j^T) P_{j-1}, \quad (4.9b)$$

$$w_j = w_{j-1} + k_j (w_j^0 - h_j^T w_{j-1}). \quad (4.9c)$$

Noting that the factors in parentheses in (4.9a) and (4.9c) are scalars, and that the vector $P_{j-1} h_j$ appears several times, we have finally the serial-processing algorithm:

$$v_j = P_{j-1} h_j, \quad (4.10a)$$

$$\alpha_j = h_j^T v_j + \sigma_j^2, \quad (4.10b)$$

$$k_j = \frac{1}{\alpha_j} v_j, \quad (4.10c)$$

$$\beta_j = w_j^0 - h_j^T w_{j-1}, \quad (4.10d)$$

$$w_j = w_{j-1} + \beta_j k_j, \quad (4.10e)$$

$$P_j = P_{j-1} - k_j v_j^T. \quad (4.10f)$$

While equations (4.10) describe a computationally efficient algorithm for assimilating a single observation, one must still pay attention to details of implementation. For example, the vector h_j is usually sparse, and its sparsity must be taken into account. If the observation falls at a grid point, then h_j consists of a single one and the rest zeroes; v_j in (4.10a) is then simply a column of P_{j-1} . We note also that when the banded approximation is applied, the outer product in (4.10f) can increase the bandwidth in principle, but we do not calculate the extra bands. Finally we note that the numerical stability of algorithm (4.10) can be enhanced by some additional calculation (e.g., Bierman, 1977, p.28; Bucy and Joseph, 1968, pp.175-176), although we have not encountered stability problems with the algorithm as it stands.

It remains to show now only how to process correlated data, i.e., how to process a batch of observations w_j^o with $p_j > 1$. The idea is simply to decorrelate the observations. Suppose we let

$$R_j = L_j \tilde{R}_j L_j^T \quad (4.11)$$

be a Cholesky decomposition of R_j (e.g., Isaacson and Keller, 1966), i.e., L_j is a lower triangular matrix with all ones on the diagonal, and \tilde{R}_j is diagonal. If we set

$$\tilde{w}_j^o = L_j^{-1} w_j^o, \quad \tilde{H}_j = L_j^{-1} H_j, \quad \tilde{b}_j^o = L_j^{-1} b_j^o, \quad (4.12a,b,c)$$

then in terms of the tilde-quantities, the observation model (4.4a,b) can be written

$$\tilde{w}_j^o = \tilde{H}_j w^* + \tilde{b}_j^o, \quad (4.13a)$$

$$E(\tilde{b}_j^o)(\tilde{b}_j^o)^T = \tilde{R}_j. \quad (4.13b)$$

Eqs. (4.13) and (4.4) are identical in form, but the new observation error covariance matrix \tilde{R}_j is diagonal: the new observations \tilde{w}_j^o have uncorrelated errors and can therefore be processed serially.

The additional work involved in processing correlated observations, then, is that involved in (4.11), (4.12a) and (4.12b). This can in fact be done serially: for $i=1,2,\dots,p_j$ one calculates the i^{th} row of L_j , the i^{th} diagonal element of \tilde{R}_j , the i^{th} element of \tilde{w}_j^o , the i^{th} row of \tilde{H}_j , and finally processes the observation (equations (4.10), but with j replaced by i and with appropriate tildes). The amount of extra work is not large since p_j is usually rather small, as in the case of a radiosonde ascent. Further, R_j typically has a small bandwidth, which can be exploited in the Cholesky decomposition. In addition, the Cholesky decomposition for any R_j which is constant in time can be calculated once and for all.

As a final remark, we point out that in processing correlated observations we are, in fact, solving linear systems, as indicated by (4.11, 4.12a,b). They are small, however, compared with (4.2a): $p_k \ll p$. They are also easier to solve than to solve (4.6a) directly: R_j itself has more properties to be exploited than does $H_j P_{j-1} H_j^T + R_j$.

V. The Shallow-Water Model

We have chosen to apply the Kalman filter for a linear, two-dimensional shallow-water system:

$$u_x + U u_x + \phi_x - (f - U_y) v = 0, \quad (5.1a)$$

$$v_x + U v_x + \phi_y + f u = 0, \quad (5.1b)$$

$$\phi_x + U \phi_x + \Phi(u_x + v_y) + \Phi_y v = 0. \quad (5.1c)$$

The coordinates x and y point eastward and northward, respectively, u and v are the eastward and northward perturbation velocities, and ϕ is the perturbation geopotential. The mean zonal current is a function of y , $U = U(y)$, which is in geostrophic balance with the mean geopotential $\Phi(y)$,

$$f U + \Phi_y = 0. \quad (5.2)$$

The Coriolis parameter is given by

$$f = f_0 + \beta y. \quad (5.3a)$$

The model domain is a 6000km by 6000km square extending from approximately 15 N to 75 N, and we take

$$f_0 = 2\Omega \sin 15^\circ, \quad \beta = 10^{-11} \text{ m}^{-1} \text{ s}^{-1} \quad (5.3b,c)$$

We specify periodic boundary conditions at the east and west boundaries, and tangential flow ($v=0$) at the north and south boundaries.

The energy E of this system,

$$E \equiv \frac{1}{2} \iint [\Phi(u^2 + v^2) + \phi^2] dx dy, \quad (5.4a)$$

satisfies

$$\frac{dE}{dt} = - \iint \Phi U_y uv \, dx \, dy. \quad (5.4b)$$

For the experiments reported here, we take \underline{U} to be constant,

$$U = U_0 = 20 \text{ m s}^{-1}, \quad (5.5a)$$

which, in light of (5.4), gives rise to a stable, energy-conserving flow.

We also take

$$\Phi = \Phi_0 - \int_0^y f U \, dy', \quad (5.5b)$$

$$\Phi_0 = 3 \times 10^4 \text{ m}^2 \text{ s}^{-2}; \quad (5.5c)$$

cf. (5.2). Experiments with a barotropically unstable profile $U(y)$ will be reported in forthcoming work.

To obtain discrete dynamics Ψ , we have chosen to apply the Richtmyer two step version of the Lax-Wendroff scheme (Richtmyer and Morton, 1967, Sec. 12.7 and 13.4) to the continuous system (5.1). This finite-difference scheme is second-order accurate both in time and in space. In addition it has the advantage of using only two time levels: a three-level scheme such as leap-frog would artificially double the number of state variables, leading to increased execution time and storage requirements, especially in the covariance evolution equation (2.4b). Instead of Lax-Wendroff, we plan eventually to incorporate a fully implicit scheme (Augenbaum et al., 1985; Cohn et al., 1985) which also uses only two time levels, and is second-order accurate in time and fourth-order accurate in space. The fully implicit scheme would allow a large time step and thereby reduce the cost of evolving the forecast error covariance.

While the version of Lax-Wendroff we have applied to (5.1) is rather standard, our implementation of it is somewhat unusual because, as discussed in

Section III, we actually compute and store the elements of Ψ . The details of this calculation are given in Appendix C. Since the coefficients in (5.1) are independent of time, so is Ψ , and Ψ is therefore calculated only once for a given model resolution. The cost of the forecast step (2.4a,b) is simply that of the indicated matrix-vector and matrix-matrix operations.

We have run the Kalman filter for our model at various spatial resolutions and for various specified bandwidths \underline{b} of $P_h^{f,a}$. Table II summarizes the CYBER 205 CPU time which was required to advance P_{h-1}^a to P_h^f , eq. (2.4b), as a function of \underline{b} and of spatial resolution. The entries show that, in all cases, the CPU times are well within acceptable limits. In fact, it is only the lack of adequate storage capacity on the NMC CYBER 205 that limits the resolution of the model: missing entries in the table correspond to experiments which are not feasible due to the excessive paging they would require. The MFLOP rates indicated in the table demonstrate the efficiency of the diagonalwise matrix multiplication algorithm described in Appendix A. The MFLOP rates are nearly independent of bandwidth, since the average vector (diagonal) length is, and the MFLOP rates increase rapidly with resolution, as the average vector length increases. At 60 x 61 resolution, nearly 90% of the peak efficiency of the CYBER 205 is attained.

Table II demonstrates both the necessity and the success of the banded approximation. Only a very low resolution model is possible without the banded approximation, and one time step on the 60 x 61 grid with bandwidth $b=3$ takes only 2/3 the CPU time of one time step on the 20 x 21 grid without the banded approximation. On the 60 x 61 grid, a conservative CFL calculation yields a maximum time step of $\frac{100 \times 10^3 \text{ m}}{400 \text{ m s}^{-1}} = 250 \text{ sec} = 4.17 \text{ min}$. With a 4-minute timestep and the 2.15 CPU seconds per time step for the 60 x 61 grid with bandwidth $b=3$ indicated in Table II, this means it takes only 12.9 minutes to

evolve the covariance matrix over one day of data assimilation. The analysis step (2.4c-e) requires only a small additional amount of time, which depends on how many observations are to be assimilated.

VI. Numerical Experiments

The Kalman filter differs from OI schemes in that it uses forecast model dynamics to attempt an accurate determination of the forecast error covariance matrix P^f . Since we take Ψ , H , Q and R to be known here, the Kalman filter determines P^f exactly. The focus of our numerical experiments will be to examine this P^f , and its dependence upon data distribution, model error, and the banded approximation. This is done by plotting the forecast error standard deviation fields, obtained from the main diagonal of P^f , and the correlation fields obtained by dividing off-diagonal elements of P^f by the standard deviations.

Most of the experiments were run on a 16 x 17 grid, with $\Delta t = 18$ min, so that $\Delta x = \Delta y = 375$ km and there are $n = 3 \times 16 \times 17 = 816$ state variables. Results of high-resolution experiments were qualitatively similar to those for the 16 x 17 grid, and will be discussed at the end of this section. We take the observational errors to be uncorrelated, and to have standard deviations

$$\sigma_u^o = \sigma_v^o = 2 \text{ m s}^{-1} \quad (6.1a,b)$$

for wind observations and

$$\sigma_\phi^o = 20 \text{ m} \times 10 \text{ m s}^{-2} \quad (6.1c)$$

for geopotential observations: R is a diagonal matrix with entries $(\sigma_u^o)^2$, $(\sigma_v^o)^2$ and $(\sigma_\phi^o)^2$ placed according to the observing pattern H , which we will vary in the experiments. Observations are made every 12 hours; in between observations we have simply $w_h^a \equiv w_h^f$ and $P_h^a = P_h^f$ in place of (2.4c-e), i.e., only (2.4a,b) are applied. The initial (analysis) error covariance matrix, P_0^a , is taken to be diagonal, with the square roots of the diagonal elements given by

$$\sigma_u^a = \sigma_v^a = 8 \text{ m s}^{-1}, \quad \sigma_\phi^a = 100 \text{ m} \times 10 \text{ m s}^{-2}; \quad (6.2a,b,c)$$

each of $(\sigma_u^a)^2$, $(\sigma_v^a)^2$ and $(\sigma_\phi^a)^2$ appears 16 x 17 times along the diagonal.

Solutions of the shallow-water system (5.1) are independent of the x-coordinate if the initial data are, and our discretization Ψ of (5.1) retains this property. It follows that if the observing pattern is independent of \underline{x} and if Q_h and P_0^a are homogeneous in \underline{x} , then P_h^f and P_h^a will be homogeneous in \underline{x} , i.e., the standard deviations will be independent of \underline{x} and the correlations will be independent of the x-coordinate of the base point. Clearly we have chosen P_0^a to be homogeneous in \underline{x} (it is homogeneous in \underline{y} also). As described below, we will always take Q_h to be homogeneous in \underline{x} also. In all of our experiments, then, inhomogeneity of P_h^f in the x-direction is due solely to x-direction dependence of the observing pattern H_h : we will be able to study the dependence of P_h^f upon data distribution in a very precise way. A summary of the experiments appears in Table III.

As a simple test of the Kalman filter, in our first experiment we choose an x-independent observing pattern: we observe each variable \underline{u} , \underline{v} , ϕ at each gridpoint along the center of the channel, i.e., at (1,9), (2,9), ..., (16,9). Thus there are 48 observations; H is a 48 x 816 matrix, each row of which contains a single one and the rest zeros. In this experiment we also take

$$Q_h = 0, \quad (6.3)$$

i.e., we assume that the forecast model is perfect. The banded approximation is not used in this experiment: we calculate the Kalman filter exactly.

Figures 1, 2, and 3 show forecast error standard deviations and correlations for this experiment at 10 days, well after the filter has settled into steady state (cf. Ghil et al., 1981, esp. Sec. 4.3). Figures 1a, b, c show

the forecast error standard deviations for $h = \phi / 10 \text{ m s}^{-2}$, \underline{u} , and \underline{v} , respectively. Each field is independent of \underline{x} , and has a local minimum along the center line $j=9$, where observations were taken 12 hours previously. In each case this local minimum is less than the corresponding observation error standard deviation (6.1); in the case of \underline{h} it is far less. The initial error (6.2) has been reduced dramatically. The standard deviations for \underline{u} and \underline{v} reach local maxima on both sides of the center line, then decrease smoothly toward the north and south boundaries: the boundary condition $v=0$ is equivalent to observing \underline{v} there. On the other hand, the standard deviations for \underline{h} increase monotonically from the center line to the boundaries, faster toward the north boundary than toward the south boundary, reaching the observational error level of 20m at the north boundary. Evidently, "observing" \underline{v} at the walls has little effect upon \underline{h} there.

In Figure 2 we show $\phi - \phi$ forecast error correlations at 10 days for the first experiment. Since P_h^f is homogeneous in \underline{x} , we only vary the y-coordinate of the base point: Figs. 2a-e show, respectively, the forecast error correlations centered at points (9,7), (9,8), (9,9), (9,10) and (9,11). One notices immediately that the correlations are not very homogeneous in \underline{y} : the five panels are rather dissimilar. The correlations are also quite different from the homogeneous correlation functions specified, for example, by the OI system at NMC, whose contours are concentric circles. Significant correlations exist well away from the base point. In Fig. 2a, for example, points along the entire south wall are correlated with the base point with a coefficient of greater than 0.5. The correlations for base points at and above the line of observations (Figs. 2c,d,e) are somewhat less spread out than those for base points below the center line (Figs. 2a,b).

Figure 3 shows the $u-\phi$ correlations corresponding to the $\phi-\phi$ correlations shown in Fig. 2. Again the correlations are not at all homogeneous in y . Figure 3c, in which the base point is the central observing station (9,9), is somewhat reminiscent of the $u-\phi$ correlation used in the OI system at NMC, although its amplitude is decreased and two lobes are present downwind of the base point. The other four panels are strikingly different from Fig. 3c, even though the base points are only 1-2 grid points away from that of Fig. 3c. This inhomogeneity is explained, at least in part, by the fact that the base points in Figs. 3a,b,d,e are located in the midst of substantial gradients in the height-field forecast error standard deviation (Fig. 1a), while this gradient is (nearly) zero at the base point (9,9). Cohn and Morone (1984) have shown that such gradients radically alter the shape of wind-height and wind-wind forecast error correlations. Comparison of Fig. 6a of Cohn and Morone (1984) with Figs 3a,b here suggests that this effect is a dominant cause of the inhomogeneity seen in Figs. 3a-e. Whenever it is true that wind-field forecast errors are related diagnostically to height-field forecast errors, there can in fact be only one other cause of wind-height and wind-wind forecast error correlation inhomogeneity, namely inhomogeneity of the height-height correlation itself (Cohn and Morone, 1984, Eqs. 2.15).

Experiment 1 is unrealistic in that a perfect model was assumed. Experiment 2 is identical to Experiment 1, except that model error is present. We take the model error to be uncorrelated, i. e., Q_h is diagonal. We take the model error standard deviations to be independent of time and space: the squares of

$$\sigma_u^2 = \sigma_v^2 = \frac{\Delta t}{12 \times 3600 \text{ s}} \times 0.5 \text{ m s}^{-1}, \quad (6.4a,b)$$

$$\sigma_\phi^2 = \frac{\Delta t}{12 \times 3600 \text{ s}} \times 6 \text{ m} \times 10 \text{ m s}^{-2}, \quad (6.4c)$$

appear repeatedly along the diagonal of $Q_A = Q$. Thus the (constant) growth rate of model error for wind components is 0.5 ms^{-1} per 12 hours, and is 6 m per 12 hours for heights.

Figures 4-6 are the analogues of Figs. 1-3 for the second experiment. The panels of Fig. 4 show the same general patterns as those of Fig. 1 while, as expected, the forecast error standard deviations are now much larger. Along the line of observations, the forecast error standard deviations are now somewhat larger than the observational error standard deviations (6.1).

The $\phi-\phi$ forecast error correlations for Experiment 2, shown in Fig. 5, are much more homogeneous than for the first experiment (Fig. 2). They are also much less spread out, and nearly circular. The model error, which is homogeneous and uncorrelated, has the effect of homogenizing and tightening the $\phi-\phi$ forecast error correlations. The correlations of Fig. 5 are in fact, remarkably similar to the homogeneous $\phi-\phi$ correlation prescribed in the OI system at NMC, even in scale. The mean e-folding distance in Figs. 5a-e is about 2 grid points, i.e., 750 km, while the e-folding distance used at NMC is 707 km. One can conclude that if the model error of current operational models is homogeneous enough and uncorrelated enough to outweigh the inhomogeneity of observing patterns and the correlations that arise from dynamical causes, then a homogeneous, localized $\phi-\phi$ correlation like that used in OI may be adequate. The extent to which this is true of current and planned models and observing systems remains to be seen.

The $u-\phi$ forecast error correlations shown in Fig. 6 are again rather inhomogeneous, and the inhomogeneity can be explained by the same argument used in reference to Fig. 3. The correlations now, however, are all quite small. One concludes that forecast errors in this case are not geostrophic, or even nearly so. If they were, the correlations in Fig. 6 would more nearly resemble

the usual OI $u-\phi$ correlation, or OI $u-\phi$ correlations accounting for gradients of height-field forecast error standard deviations (Cohn and Morone, 1984, Figs. 5a,6a), since the $\phi-\phi$ correlations of Fig. 5 already resemble strongly the usual OI $\phi-\phi$ correlation. This is probably because our homogeneous model error (6.4) feeds energy into ageostrophic, as well as geostrophic modes. We have not yet attempted to specify separately the amount of model error in geostrophic and ageostrophic modes, or rotational and divergent modes.

In the remaining experiments, numbered 3-6, we observe along a line perpendicular to the mean zonal current: u , v and ϕ are observed at the 15 points (9,2), (9,3), ..., (9,16). Each of these experiments uses the same model error covariance Q that was used in Experiment 2, eqs. (6.4). We will study the effect of the banded approximation: Experiment 3 does not use the banded approximation, while in Experiments 4-6 we specify the bandwidth $b = 5, 4, \text{ and } 3$, respectively. Apart from the bandwidth, Experiments 3-6 are identical (see Table III). Again we plot the forecast error standard deviations and correlations at 10 days.

Figure 7 shows the height-field forecast error standard deviations for Experiments 3-6. The banded approximation appears to be a good one: differences between Figs. 7a-d are generally less than one meter. The only large differences are confined near the north wall. Forecast errors are actually highly correlated along the entire length of this boundary, as other plots (not shown) have demonstrated, so that the banded approximation breaks down there. Still, there is no effect more than 1-2 grid points from the boundary. Notice also that the standard deviation at each grid point decreases slightly as the bandwidth is decreased, whereas we know that the true standard deviation actually increases slightly: the Kalman filter without banded approximation (Fig. 7a) is optimal.

The banded approximation removes some variance from the system: zeroing out off-diagonals of the covariance matrix makes it less positive definite. In practice this should not be a problem if the decrease is small, and it can be accounted for, if necessary, by increasing the model error covariance matrix Q .

For such a simple experiment, Fig. 7 demonstrates a surprising degree of complexity. Of course, one no longer expects independence of \underline{x} or \underline{y} , as in Figs. 1 and 4, since the observing pattern now depends on \underline{x} (it is a discrete δ -function), and the coefficients of the model (5.1) depend on \underline{y} . The standard deviations are smaller to the southeast of the observing line than to the northwest, indicating that this is the direction in which most of the information propagates. Such effects of propagation of information are not accounted for in present-day OI systems. Notice also that observing perpendicular to the zonal current is more effective at reducing height forecast errors than observing parallel to it: away from the north boundary, the standard deviations in Fig. 7a are all 20-25 m, while in Fig. 4a they are 25-35 m over much of the domain. This is what one might expect comparing these two simple observing patterns. It also demonstrates the usefulness of the Kalman filter in performing observing systems simulation experiments (OSSE) for much more complicated observing patterns and systems.

Figures 8 and 9, corresponding to Fig. 7, show the forecast error standard deviations in \underline{u} and \underline{v} , respectively, for Experiments 3-6. Again the degree of complexity is greater than in the first two experiments (Figs. 1 b, c, and 4 b, c), but less than in Fig. 7. The wind information appears to propagate zonally, rather than northwest-to-southeast, and is symmetric with respect to the latitude line $j=9$. The perpendicular observing system appears roughly as effective as the parallel one in reducing forecast errors in \underline{u} and \underline{v} : compare Figs. 8a, 9a with Figs. 4b, 4c. Figures 8 and 9 also show the success of the

banded approximation: everywhere in Fig. 9, and away from the boundaries in Fig. 8, the differences between panels are very slight.

Figure 10 shows the $\phi - \phi$ correlation at the central point (9,9) for each of Experiments 3-6. The small square in Figs. 10b-d indicate the bandwidth employed. The banded approximation handles the correlations quite accurately. Even for bandwidths as small as $b=3$ (Fig. 10d), contours from 0.3 upwards are placed well (cf Fig. 10a). Only the 0.1 contour, indicating a negligible correlation for the purposes of data analysis, is misplaced slightly. Figures 11 and 12, which show the $u-u$ and $v-v$ correlations corresponding to Fig. 10, also demonstrate the accuracy of the banded approximation: only the ± 0.1 contours are distorted slightly.

We have carried out to date a number of experiments at resolutions of up to 60×60 points over our 6000×6000 km grid. As shown in Table I, the banded approximation is of utmost importance in these high-resolution experiments. The results are encouraging: the banded approximation appears to retain its accuracy, independently of resolution. The reason for this is that, as resolution is increased, we have found that the characteristic length scales of the forecast error correlations decrease proportionately. One can expect this to be strictly the case only when model errors are uncorrelated (as they were in our experiments), or at least have correlation length scales which decrease proportionately with resolution, and are large enough to overwhelm the spreading of correlations induced by dynamical effects (compare Figs. 2 and 5). Theoretical work in progress is aimed toward determining the circumstances under which model error due to discretization of the governing differential equations satisfies these two requirements.

VII. Concluding Remarks

We have devised and implemented a number of techniques which, taken together, render the Kalman filter practical for assimilating data in two spatial dimensions. Numerical experiments using a two-dimensional shallow-water model demonstrate, in a fairly realistic setting, some of the advantages offered by the Kalman filter over the OI methods in current operational use.

Our ultimate goal in this work is to develop a Kalman filter which is suitable for routine assimilation of data into a fully three-dimensional model. Experience gained with the two-dimensional code suggests that this goal is well within reach, although not without additional refinements to our method. Improvements mentioned in the text include using a fully implicit forecast model, storing covariances compactly by separating variances and correlations, and using extrapolation techniques in the banded approximation. We are currently exploring these possibilities, as well as additional ones, which should further reduce the amount of work required by the Kalman filter, with little effect on its accuracy.

Acknowledgements. It is a pleasure to acknowledge fruitful discussions with Dick Dee and Michael Ghil. This work has benefitted enormously from the enthusiastic support of Ron McPherson. We are also grateful to Lauren Morone, who assisted in testing the code.

Appendix A. Diagonalwise Matrix Triple Product

Here we develop an efficient algorithm for evaluating the matrix triple product appearing in the covariance forecast equation (2.4b), which is of the form

$$C = ABA^T \quad (\text{A.1})$$

where A and B are large square matrices, stored by diagonals, and where B and C are symmetric, $B^T=B$ and $C^T=C$. The algorithm obtains each diagonal of C in turn, by multiplying appropriate diagonals of A and B. For full matrices A and B, the diagonalwise algorithm has about the same computational complexity as standard row-column algorithms. In our application, A has only a small number of nonzero diagonals and B does also when the banded approximation is employed. The diagonalwise algorithm gains considerable efficiency by exploiting this fact. Vectorization of the algorithm is over the elements of each nonzero diagonal. The vectors (diagonals) are long since A and B are large, an ideal situation for calculation on the CYBER 205.

First we introduce a notation which describes how the matrices are stored. For a general matrix G of dimension N x N, we number from 1 to N the diagonals from the main diagonal downwards. Diagonals above the main diagonal are numbered 2 to N. Thus, suppose G^L and G^U are the lower and strictly upper triangular parts of G,

$$G = G^L + G^U, \quad (\text{A.2})$$

$$G_{ij}^L = G_{ij} \text{ for } i=1, \dots, N \text{ and } j=1, \dots, i, \quad (\text{A.3a})$$

$$G_{ij}^U = G_{ij} \text{ for } j=2, \dots, N \text{ and } i=1, \dots, j-1. \quad (\text{A.3b})$$

Letting

$$n_L = i - j + 1, \quad n_U = j - i + 1, \quad (\text{A.4a,b})$$

we then define

$$G^L[j, n_L] \equiv G_{ij}^L \quad \text{for } n_L = 1, \dots, N \quad \text{and } j = 1, \dots, N - n_L + 1, \quad (\text{A.5a})$$

$$G^U[i, n_U] \equiv G_{ij}^U \quad \text{for } n_U = 2, \dots, N \quad \text{and } i = 1, \dots, N - n_U + 1; \quad (\text{A.5b})$$

$G^L[j, n_L]$ is the j^{th} element of the n_L^{th} diagonal of G^L , and similarly for $G^U[i, n_U]$. Notice that $G^L[k, n] = G^U[k, n]$ if G is symmetric. To store G^L , say, we retain an array $\alpha^L(n_L)$ containing the storage address of the first element of each nonzero (i.e., stored) diagonal of G^L ; the address of the element $G^L[j, n_L]$ is then simply $\alpha^L(n_L) + j - 1$ if the n_L^{th} diagonal is stored. If the n_L^{th} diagonal is zero (not stored), it is flagged by setting $\alpha^L(n_L) = 0$.

To describe the diagonalwise matrix multiplication algorithm, we first expand (A.1) in row-column form and split the result into upper and lower triangular parts. Letting the dimension of A , B and C be $I \times I$, the $(i, j)^{\text{th}}$ element of C in (A.1) is simply

$$C_{ij} = \sum_{k=1}^I \sum_{l=1}^I A_{ik} B_{kl} A_{jl}, \quad (\text{A.6})$$

or

$$C_{ij} = \sum_{k=1}^I A_{ik} T_{kj}, \quad (\text{A.7a})$$

where

$$T_{kj} = \sum_{l=1}^I B_{kl} A_{jl}. \quad (\text{A.7b})$$

Since C is symmetric, we need only calculate its lower triangular part $C_{ij}^L, i \geq j$.
 For $i \geq j$, (A.7a) can be split into three parts

$$C_{ij}^L = \sum_{k=1}^{j-1} A_{ik}^L T_{kj}^L + \sum_{k=j}^i A_{ik}^L T_{kj}^L + \sum_{k=i+1}^I A_{ik}^U T_{kj}^L. \quad (\text{A.8})$$

Notice that the first two sums in (A.8) reference only the lower triangular part of A and the last one references only the strictly upper triangular part of A while the first sum references only the strictly upper triangular part of T and the last two reference only the lower triangular part of T: we have

$$C_{ij}^L = \sum_{k=1}^{j-1} A_{ik}^L T_{kj}^U + \sum_{k=j}^i A_{ik}^L T_{kj}^L + \sum_{k=i+1}^I A_{ik}^U T_{kj}^L. \quad (\text{A.9a})$$

Similarly, (A.7b) can be rewritten as

$$T_{kj}^L = \sum_{l=1}^j B_{kl}^L A_{jl}^L + \sum_{l=j+1}^k B_{kl}^L A_{jl}^U + \sum_{l=k+1}^I B_{kl}^L A_{jl}^U, \quad (\text{A.9b})$$

$$T_{kj}^U = \sum_{l=1}^k B_{kl}^L A_{jl}^L + \sum_{l=k+1}^j B_{kl}^L A_{jl}^L + \sum_{l=j+1}^I B_{kl}^L A_{jl}^U. \quad (\text{A.9c})$$

The symmetry of B was used in the last sum in (A.9b) and the last two sums in (A.9c); B^U is not stored or referenced.

The diagonalwise multiplication algorithm evaluates each of the nine sums in (A.9) by diagonals. We describe in detail how the first sum in (A.9b),

$$T_{kj}^{L1} = \sum_{l=1}^j B_{kl}^L A_{jl}^L \quad (\text{A.10})$$

is obtained; the procedure for the remaining sums will be clear by analogy.

To begin we write (A.10) in cumulative sum notation, i.e.

$$\text{for } k=1, \dots, I \quad (\text{A.11a})$$

$$\text{for } j=1, \dots, k \quad (\text{A.11b})$$

$$\text{for } l=1, \dots, j \quad (\text{A.11c})$$

$$T_{kj}^{L1} := T_{kj}^{L1} + B_{kl}^L A_{jl}^L. \quad (\text{A.11d})$$

The notation $\underline{:=}$ means to replace the left-hand-side by the right-hand-side; thus (A.11c,d) are equivalent to (A.10) if $T_{kj}^{L1} = 0$ initially, which we take to be the case.

The two loops (A.11a,b) indicate that the operation (A.11c,d) is to be carried out for each element (k,j) of the lower triangular matrix T^{L1} , cf. (A.3a). These two loops can be replaced by one loop over the diagonals of T^{L1} and a second over the elements along a diagonal of T^{L1} : comparing (A.3a) with (A.5a), we see that (A.11) is equivalent to

$$\text{for } n_T = 1, \dots, I \quad (\text{A.12a})$$

$$\text{for } j = 1, \dots, I - n_T + 1 \quad (\text{A.12b})$$

$$\text{for } l = 1, \dots, j \quad (\text{A.12c})$$

$$T^{L1}[j, n_T] := T^{L1}[j, n_T] + B_{j+n_T-1}^L A_{j,l}^L. \quad (\text{A.12d})$$

The first subscript on B in (A.12d) follows from the definition of the diagonal index n_T of T_{kj}^{L1} , i.e.,

$$n_T = k - j + 1, \quad (\text{A.13a})$$

cf (A.4a), so that

$$k = j + n_T - 1 \quad (\text{A.13b})$$

is the first subscript of B^L . The index k in (A.11) has been replaced entirely by the index n_T in (A.12).

The inner loops (A.12b,c) reference part of the lower triangular matrix A^L in (A.12d), namely they reference the upper-left-most submatrix of A^L of dimension $(I - n_T + 1) \times (I - n_T + 1)$. Again comparing (A.3a) with (A.5a), we find that (A.12) is equivalent to

$$\text{for } n_T = 1, \dots, I \quad (\text{A.14a})$$

$$\text{for } n_A = 1, \dots, I - n_T + 1 \quad (\text{A.14b})$$

$$\text{for } l = 1, \dots, I - n_T - n_A + 2 \quad (\text{A.14c})$$

$$T^{L1}[l+n_A-1, n_T] := T^{L1}[l+n_A-1, n_T] + B[l, n_A+n_T-1] A^L[l, n_A]. \quad (\text{A.14d})$$

The index j has been eliminated in passing from (A.12d) to (A.14d) by using the definition of the diagonal index n_A of $A_{j\lambda}^L$, i.e.,

$$n_A = j - \lambda + 1, \quad (\text{A.15a})$$

cf. (A.4a), which implies that

$$j = \lambda + n_A - 1. \quad (\text{A.15b})$$

Equations (A.14a-d) represent the algorithm for calculating the first sum (A.10) in (A.9b). The algorithm is vectorized over the innermost loop (A.14c): the vector length is $I - n_T - n_A + 2$. Equation (A.14d) states that, for given n_T and n_A , multiply together diagonals n_A of A^L and $n_A + n_T - 1$ of B^L and add the result to diagonal n_T of T^{L1} , offset by $n_A - 1$ elements. The remaining sums in (A.9a-c) are obtained in a similar manner.

The overall algorithm for (A.9a-c) requires storage for only one diagonal of T^L or T^U , rather than for the entire matrices T^L and T^U themselves: the diagonals of T^L or T^U are calculated only as they are needed in (A.9a). This is possible because the loop on n_T is the outermost loop in (A.14), as it is for the other sums in (A.9b,c), and the (vectorized) loop over the elements of a diagonal of T^L or T^U is the innermost loop for the sums in (A.9a). When the banded approximation is used, it is only (A.9a), and not (A.9b,c), which is approximated: only those diagonals of C within the specified bandwidth are calculated, of course, but all nonzero diagonals of T are calculated.

The algorithm, as it stands, exploits the symmetry of B and C to reduce both storage and execution-time requirements: only B^L and C^L are calculated and stored. It also reduces execution time by using the fact that A (the model dynamics) is a banded matrix with small bandwidth: only those diagonals of T^L and T^U which multiply nonzero diagonals of A^L and A^U in (A.9a) are calculated in (A.9b,c). It is also possible to reduce storage

requirements further by exploiting the small bandwidth of A. If A were diagonal then obviously C^L could be overwritten on B^L . More generally, for any A, C^L can be overwritten on B^L if auxiliary storage roughly the same size as that for A is provided. We have not yet implemented this idea.

Appendix B. Serial Processing of Observations

Here we prove the equivalence of (4.2) and (4.6). It suffices to show the equivalence for $J=2$; the general case follows by an inductive argument.

We begin by writing (4.2) in a more convenient form. From (4.2a) we have

$$KHP^f H^T + KR = P^f H^T,$$

or

$$KHP^f H^T R^{-1} + K = P^f H^T R^{-1},$$

or

$$K = (I - KH) P^f H^T R^{-1},$$

which, from (4.2b), can be written as

$$K = P^a H^T R^{-1}, \tag{B.1}$$

so that (4.2c) becomes

$$w^a = w^f + P^a H^T R^{-1} (w^o - H w^f) \tag{B.2}$$

Now, according to (4.3),

$$H = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix}, \quad R = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}, \quad w^o = \begin{bmatrix} w_1^o \\ w_2^o \end{bmatrix}, \tag{B.3a,b,c}$$

so that (B.2) can be written as

$$w^a = w^f + P^a \begin{bmatrix} H_1^T R_1^{-1} & H_2^T R_2^{-1} \end{bmatrix} \begin{bmatrix} w_1^o - H_1 w^f \\ w_2^o - H_2 w^f \end{bmatrix},$$

or

$$w^a = w^f + P^a \left[H_1^T R_1^{-1} (w_1^o - H_1 w^f) + H_2^T R_2^{-1} (w_2^o - H_2 w^f) \right]. \tag{B.4}$$

The Woodbury formula states the two equations

$$C = B - BA^T (ABA^T + R)^{-1} AB, \tag{B.5a}$$

$$C^{-1} = B^{-1} + A^T R^{-1} A \tag{B.5b}$$

are equivalent for arbitrary conformable matrices A, B, C, R , provided the indicated inverses exist. Since (4.2a) substituted into (4.2b) gives

$$P^a = P^f - P^f H^T (H P^f H^T + R)^{-1} H P^f,$$

it follows that

$$(P^a)^{-1} = (P^f)^{-1} + H^T R^{-1} H, \quad (\text{B.6})$$

which is a statement that the analysis accuracy is the sum of the forecast and observation accuracies. Substituting (B.3a,b) into (B.6) gives

$$(P^a)^{-1} = (P^f)^{-1} + H_1^T R_1^{-1} H_1 + H_2^T R_2^{-1} H_2. \quad (\text{B.7})$$

We will show the desired equivalence by showing that (B.4) and (B.7) hold for the algorithm (4.5, 4.6, 4.7).

To show that (B.7) holds, we simply apply the Woodbury formula again, to (4.6a,b) with $j=1$ and $j=2$, to get

$$(P_1)^{-1} = (P_0)^{-1} + H_1^T R_1^{-1} H_1, \quad (\text{B.8a})$$

$$(P_2)^{-1} = (P_1)^{-1} + H_2^T R_2^{-1} H_2; \quad (\text{B.8b})$$

substituting (4.5a, 4.7a) here then gives (B.7). To verify that (B.4) holds, we write (4.6c) for $j=1$ and $j=2$:

$$w_1 = w^f + P_1 H_1^T R_1^{-1} (w_0^o - H_1 w^f), \quad (\text{B.9a})$$

$$w_2 = w_1 + P_2 H_2^T R_2^{-1} (w_2^o - H_2 w^f), \quad (\text{B.9b})$$

where we have used an argument similar to that leading to (B.2). Substituting (B.9b) into (B.9a) gives

$$w_2 = w^f + P_1 H_1^T R_1^{-1} (w_1^o - H_1 w^f) + P_2 H_2^T R_2^{-1} \{ w_2^o - H_2 [w^f + P_1 H_1^T R_1^{-1} (w_1^o - H_1 w^f)] \},$$

or

$$w_2 = w^f + (I - P_2 H_2^T R_2^{-1} H_2) P_1 H_1^T R_1^{-1} (w_1^o - H_1 w^f) + P_2 H_2^T R_2^{-1} (w_2^o - H_2 w^f).$$

(B.10)

Now, premultiplying (B.8b) by P_2 and postmultiplying by P_1 gives

$$P_1 = P_2 + P_2 H_2^T R_2^{-1} H_2 P_1,$$

or

$$P_2 = (I - P_2 H_2^T R_2^{-1} H_2) P_1. \quad (\text{B.11})$$

Using (B.11) in (B.10), and using (4.7a,b), gives (B.4).

Appendix C. Finite-Difference Scheme

Letting $w = (u, v, \phi)^T$, the shallow-water system (5.1) can be written in flux form as

$$w_x + \frac{\partial}{\partial x} Aw + \frac{\partial}{\partial y} Bw + Cw = 0 \quad (C.1a)$$

where

$$A = \begin{bmatrix} U & 0 & 1 \\ 0 & U & 0 \\ \Phi & 0 & U \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & \Phi & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & -(f - U_y) & 0 \\ f & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (C.1b,c,d)$$

We will discretize (C.1) on a uniform grid of $I \times J$ points and denote the elements of the discrete state vector by

$$w_{i,j}^k \equiv w[(i-1)\Delta x, (j-1)\Delta y, k\Delta t] \quad (C.2a)$$

for $i=1,2,\dots,I$, $j=1,2,\dots,J$, and $k=0,1,2,\dots$, where

$$\Delta x = X/I, \quad \Delta y = Y/(J-1) \quad (C.2b,c)$$

and

$$X = Y = 6000 \text{ km}. \quad (C.2d,e)$$

We let

$$\lambda_x = \frac{\Delta t}{\Delta x}, \quad \lambda_y = \frac{\Delta t}{\Delta y} \quad (C.3a,b)$$

and define the difference operators δ_x and δ_y by

$$\delta_x w_{i,j} = w_{i+\frac{1}{2},j} - w_{i-\frac{1}{2},j}, \quad \delta_y w_{i,j} = w_{i,j+\frac{1}{2}} - w_{i,j-\frac{1}{2}} \quad (C.4a,b)$$

and the averaging operators μ_x and μ_y by

$$\mu_x w_{i,j} = \frac{1}{2} (w_{i+\frac{1}{2},j} + w_{i-\frac{1}{2},j}), \quad \mu_y w_{i,j} = \frac{1}{2} (w_{i,j+\frac{1}{2}} + w_{i,j-\frac{1}{2}}) \quad (C.4c,d)$$

The Richtmyer two-step version of the Lax-Wendroff scheme leads readily to algebraic formulas for the elements of the dynamics matrix Ψ . The first step of the scheme, as it is usually applied, calculates provisional values at centers of grid boxes,

$$w_{i+\frac{1}{2}, j+\frac{1}{2}}^{k+\frac{1}{2}} = \mu_x \mu_y w_{i+\frac{1}{2}, j+\frac{1}{2}}^k - \frac{1}{2} L_{j+\frac{1}{2}} w_{i+\frac{1}{2}, j+\frac{1}{2}}^k \quad (C.5a)$$

for $i=1, 2, \dots, I$ and $j=1, 2, \dots, J-1$, where the operator L is defined as

$$L_j = \lambda_x \mu_y \delta_x A_j + \lambda_y \mu_x \delta_y B_j + \Delta t \mu_x \mu_y C_j \quad (C.5b)$$

The second step uses these values to calculate

$$\tilde{w}_{ij}^{k+1} = w_{ij}^k - L_j w_{ij}^k \quad (C.5c)$$

for $i=1, 2, \dots, I$ and $j=2, 3, \dots, J-1$ i.e., the values at the new time step on

the original grid, excluding the north and south boundaries. We will combine

(C.5a) and (C.5c) along with the boundary treatment for \underline{u} and \underline{v} (described

later), to give provisional values \tilde{w}^{k+1} at all grid points. Then a bound-

ary treatment for ϕ will be applied to give the final values w^{k+1} .

Thus Ψ will be a composition of an interior operator Ψ^I and a boundary

operator Ψ^B , $\Psi = \Psi^B \Psi^I$ and

$$w^{k+1} = \Psi^B \tilde{w}^{k+1} = \Psi^B \Psi^I w^k \quad (C.6)$$

Most of Ψ^B is just the identity; Ψ^I does most of the work.

To arrive at formulas for the elements of Ψ^I , we first expand (C.5c)

using (C.5b) and (C.4a-d), to find

$$\begin{aligned} \tilde{w}_{ij}^{k+1} = w_{ij}^k &- M_{\frac{1}{2}, \frac{1}{2}}^j w_{i+\frac{1}{2}, j+\frac{1}{2}}^{k+\frac{1}{2}} - M_{\frac{1}{2}, -\frac{1}{2}}^j w_{i+\frac{1}{2}, j-\frac{1}{2}}^{k+\frac{1}{2}} \\ &- M_{-\frac{1}{2}, \frac{1}{2}}^j w_{i-\frac{1}{2}, j+\frac{1}{2}}^{k+\frac{1}{2}} - M_{-\frac{1}{2}, -\frac{1}{2}}^j w_{i-\frac{1}{2}, j-\frac{1}{2}}^{k+\frac{1}{2}} \end{aligned} \quad (C.7a)$$

where

$$M_{\pm \frac{1}{2}, \frac{1}{2}}^j = \frac{1}{2} \left(\pm \lambda_x A_{j+\frac{1}{2}} + \lambda_y B_{j+\frac{1}{2}} + \frac{\Delta t}{2} C_{j+\frac{1}{2}} \right), \quad (C.7b)$$

$$M_{\pm \frac{1}{2}, -\frac{1}{2}}^j = \frac{1}{2} \left(\pm \lambda_x A_{j-\frac{1}{2}} - \lambda_y B_{j-\frac{1}{2}} + \frac{\Delta t}{2} C_{j-\frac{1}{2}} \right). \quad (C.7c)$$

Substituting (C.5a) into (C.7a) and again expanding operators gives finally

$$\begin{aligned}
 \tilde{w}_{ij}^{k+1} = & \left[I - M_{\frac{1}{2}, \frac{1}{2}}^j N_{00}^j - M_{\frac{1}{2}, -\frac{1}{2}}^j N_{01}^{j-1} - M_{-\frac{1}{2}, \frac{1}{2}}^j N_{10}^j - M_{-\frac{1}{2}, -\frac{1}{2}}^j N_{11}^{j-1} \right] w_{ij}^k \\
 & - \left[M_{\frac{1}{2}, \frac{1}{2}}^j N_{10}^j + M_{\frac{1}{2}, -\frac{1}{2}}^j N_{11}^{j-1} \right] w_{i+1, j}^k - \left[M_{\frac{1}{2}, \frac{1}{2}}^j N_{01}^j + M_{-\frac{1}{2}, \frac{1}{2}}^j N_{11}^j \right] w_{i, j+1}^k \\
 & - \left[M_{-\frac{1}{2}, \frac{1}{2}}^j N_{00}^j + M_{-\frac{1}{2}, -\frac{1}{2}}^j N_{01}^{j-1} \right] w_{i-1, j}^k - \left[M_{\frac{1}{2}, -\frac{1}{2}}^j N_{00}^j + M_{-\frac{1}{2}, -\frac{1}{2}}^j N_{10}^j \right] w_{i, j-1}^k \\
 & - \left[M_{\frac{1}{2}, \frac{1}{2}}^j N_{11}^j \right] w_{i+1, j+1}^k - \left[M_{\frac{1}{2}, -\frac{1}{2}}^j N_{10}^{j-1} \right] w_{i+1, j-1}^k \\
 & - \left[M_{-\frac{1}{2}, \frac{1}{2}}^j N_{01}^j \right] w_{i-1, j+1}^k - \left[M_{-\frac{1}{2}, -\frac{1}{2}}^j N_{00}^{j-1} \right] w_{i-1, j-1}^k
 \end{aligned} \tag{C.8a}$$

where

$$N_{\frac{1}{2} \pm \frac{1}{2}, \frac{1}{2} \pm \frac{1}{2}}^j = \frac{1}{4} I - \frac{1}{2} M_{\pm \frac{1}{2}, \pm \frac{1}{2}}^{j+\frac{1}{2}}. \tag{C.8b}$$

Equation (C.8a) gives for each interior point $i=1, 2, \dots, I$, $j=2, 3, \dots, J-1$,

the value of \tilde{w}_{ij}^{k+1} directly in terms of values at neighboring grid points at the previous time step. Each bracketed expression in (C.8a) is a 3×3 matrix, and Ψ^I is composed of these 3×3 matrices. The symbol I in (C.8a,b) denotes the 3×3 identity matrix, not the number of grid points in the x -direction.

The boundary condition

$$v = 0 \tag{C.9a}$$

at the north and south boundaries implies, according to (5.1a,b), that

$$u_x + U u_x + \phi_x = 0 \tag{C.9b}$$

and

$$\phi_y + f u = 0 \tag{C.9c}$$

there also. Equation (C.9b) is treated by the ordinary Lax-Friedrichs

scheme

$$u_{ij}^{k+1} = (2\mu_x^2 - 1) u_{ij}^k - \lambda_x \mu_x \delta_x (U_j u_{ij}^k + \phi_{ij}^k). \tag{C.10}$$

The diagnostic equation (C.9c) is treated by first-order uncentered differences:

$$\frac{1}{\Delta y} \delta_y \phi_{i, J-1/2}^{k+1} + f_J u_{i, J}^{k+1} = 0, \quad (\text{C.11a})$$

$$\frac{1}{\Delta y} \delta_y \phi_{i, 3/2}^{k+1} + f_1 u_{i, 1}^{k+1} = 0. \quad (\text{C.11b})$$

According to (C.11), the boundary values of ϕ depend on the boundary values of \underline{u} at the same time step, which necessitates our solving (C.10) before (C.11): equations (C.9a, C.10) will be formulated as part of the interior operator Ψ^I , and (C.11) will be formulated in the boundary operator Ψ^B .

We write (C.9a, C.10) as

$$\tilde{w}_{i, j}^{k+1} = \frac{1}{2} [I^B - \lambda_x A_j^B] w_{i+1, j}^k + \frac{1}{2} [I^B + \lambda_x A_j^B] w_{i-1, j}^k \quad (\text{C.12a})$$

for $i=1, 2, \dots, I$, $j=1$ and $j=J$, where

$$I^B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad A_j^B = \begin{bmatrix} U_j & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (\text{C.12b,c})$$

Equations (C.8, C.12) complete the description of the interior operator Ψ^I , which updates \underline{u} , \underline{v} and ϕ at interior points, and \underline{u} and \underline{v} at the boundaries.

Note from (C.12) that one application of Ψ^I also zeros out ϕ at the boundaries. Subsequent multiplication by Ψ^B , described next, will refresh ϕ at the boundaries. Equations (C.11) can be written as

$$\phi_{i, J}^{k+1} = \phi_{i, J-1}^{k+1} - \Delta y f_J u_{i, J}^{k+1}, \quad (\text{C.13a})$$

$$\phi_{i, 1}^{k+1} = \phi_{i, 2}^{k+1} + \Delta y f_1 u_{i, 1}^{k+1}, \quad (\text{C.13b})$$

so that the boundary part of Ψ^B is given by

$$\omega_{iJ}^{k+1} = [I - E^B - \Delta y D_j^B] \tilde{\omega}_{iJ}^{k+1} + E^B \tilde{\omega}_{i,J-1}^{k+1}, \quad (C.14a)$$

$$\omega_{i1}^{k+1} = [I - E^B + \Delta y D_i^B] \tilde{\omega}_{i1}^{k+1} + E^B \tilde{\omega}_{i2}^{k+1}, \quad (C.14b)$$

for $i=1,2,\dots,I$, where

$$D_j^B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ f_j & 0 & 0 \end{bmatrix}, \quad E^B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (C.14c,d)$$

The interior part of Ψ^B is simply

$$\omega_{ij}^{k+1} = \tilde{\omega}_{ij}^{k+1} \quad (C.15)$$

for $i=1,2,\dots,I$ and $j=2,3,\dots,J-1$.

REFERENCES

- Augenbaum, J. M., S. E. Cohn, D. P. Dee, E. Isaacson and D. Marchesin, 1985: A fully implicit scheme for global numerical weather prediction. Seventh Conference on Numerical Weather Prediction, preprint volume, Amer. Meteor. Soc., Boston, to appear.
- Bergman, K. H., 1979: Multivariate analysis of temperatures and winds using optimum interpolation. Mon. Wea. Rev., 107, 1423-1444.
- Bierman, G. J., 1977: Factorization Methods for Discrete Sequential Estimation. Academic Press, New York 241 pp.
- Bucy, R. S., and P. D. Joseph, 1968: Filtering for Stochastic Processes with Applications to Guidance. Wiley-Interscience, New York, 195 pp.
- Cohn, S. E., 1982: Methods of Sequential Estimation for Determining Initial Data in Numerical Weather Prediction. Ph.D. dissertation, Courant Institute of Mathematical Sciences Report No. CI-6-82, New York University, 183 pp.
- Cohn, S. E., and L. L. Morone, 1984: The effect of horizontal gradients of height-field forecast error variances upon OI forecast error statistics. Office Note 296, National Meteorological Center, Washington, D.C., 20233, 37pp.
- Cohn, S. E., M. Ghil and E. Isaacson, 1981: Optimal interpolation and the Kalman filter. Fifth Conference on Numerical Weather Prediction, preprint volume, Amer. Meteor. Soc., Boston, pp. 36-42.
- Cohn, S. E., D. Dee, E. Isaacson, D. Marchesin and G. Zwas, 1985: A fully implicit scheme for the barotropic primitive equations. Mon. Wea. Rev., to appear in March.
- Dee, D., S. E. Cohn and M. Ghil, 1984: An efficient algorithm for estimating noise covariances in distributed systems. IEEE Trans. on Automatic Control, accepted for publication.
- Dee, D., S.E. Cohn and M. Ghil, 1985: Systematic estimation of forecast and observation error covariances in four-dimensional data assimilation. Seventh Conference on Numerical Weather Prediction, preprint volume, Amer. Meteor. Soc., Boston, to appear.
- Gelb, A. (ed.), 1974: Applied Optimal Estimation. The M.I.T. Press, Cambridge, MA, 374 pp.
- Ghil, M., S. Cohn, J. Tavantzis, K. Bube and E. Isaacson, 1981: Applications of estimation theory to numerical weather prediction. Dynamic Meteorology: Data Assimilation Methods, L. Bengtsson, M. Ghil and E. Kallen (eds.), Springer-Verlag, New York, pp. 139-224.

- Ghil, M., S. E. Cohn and A. Dalcher, 1982: Sequential estimation, data assimilation and initialization. The Interaction between Objective Analysis and Initialization: Proceedings of the Fourteenth Stanstead Seminar, National Center for Atmospheric Research, Boulder CO, pp. 83-97.
- Isaacson, E., and H. B. Keller, 1966: Analysis of Numerical Methods. Wiley, New York, 541 pp.
- Jazwinski, A. H., 1970: Stochastic Processes and Filtering Theory. Academic Press, New York, 376 pp.
- Lorenc, A., 1981: A global three-dimensional multivariate statistical interpolation scheme. Mon. Wea. Rev., 109, 701-721.
- Madsen, N. K., G. H. Rodrigue and J. I. Karush, 1976: Matrix multiplication by diagonals on a vector/parallel processor. Information Processing Letters, 5, 41-45.
- Morone, L. L., and S. E. Cohn, 1984: The effect of spherical distance approximations upon OI forecast error correlations. Office Note 297, National Meteorological Center, Washington DC 20233, 43 pp.
- Morone, L. L. and S. E. Cohn, 1985: The spatial variability of height-field forecast errors and its effect upon OI forecast error statistics. Seventh Conference on Numerical Weather Prediction, preprint volume, Amer. Meteor. Soc., Boston, to appear.
- Phillips, N. A., 1982: A very simple application of Kalman filtering to meteorological data assimilation. Office Note 258, National Meteorological Center, Washington DC 20233, 27 pp.
- Richtmyer, R. D., and K. W. Morton, 1967: Difference Methods for Initial-Value Problems, 2nd ed. Wiley-Interscience, New York, 405 pp.

<u>b</u>	<u>S/N</u>	<u>N</u>	$\sqrt{N} \times \sqrt{N}$
0	9	33.3×10^4	577 x 577
1	81	3.7×10^4	192 x 192
2	225	1.3×10^4	115 x 115
3	441	68.0×10^2	82 x 82
4	729	41.2×10^2	64 x 64
5	1089	27.6×10^2	52 x 52
6	1521	19.7×10^2	44 x 44
7	2025	14.8×10^2	38 x 38
8	2601	11.6×10^2	33 x 33
9	3249	9.2×10^2	30 x 30
full	9N	5.8×10^2	24 x 24

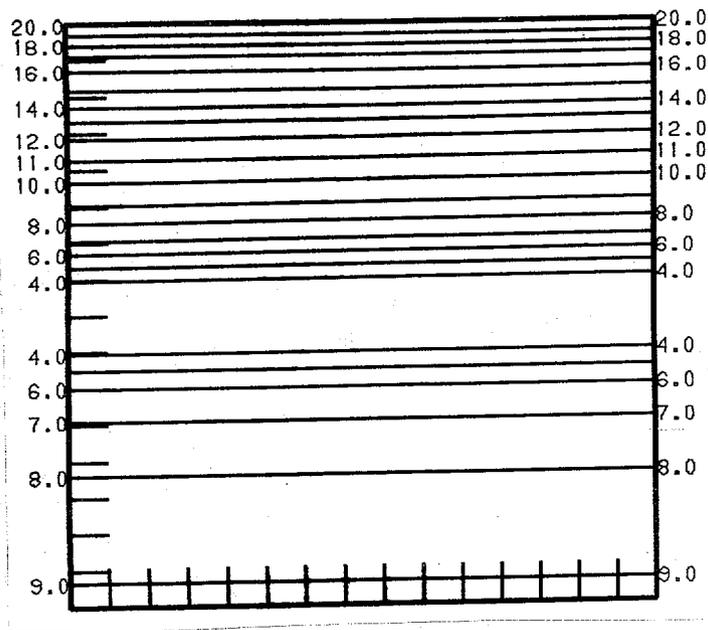
Table I. Storage requirements for $P^{f,a}$ as a function of bandwidth b, computed from the formula $S = Nv^2(2b+1)^d$, with the number of spatial dimensions $d = 2$ and the number of dependent variables $v = 3$; S is the total required storage and N is the total number of grid points. The first column gives the bandwidth and the second gives the storage required per grid point. For available storage $S = 3 \times 10^6$, the third column gives the maximum total number of grid points and the last gives the maximum grid dimensions for a uniform square grid. The last row gives requirements for storing the full $P^{f,a}$, using the formula $S = (Nv)^2$.

<u>bandwidth b</u>	<u>resolution</u>		
	<u>20 x 21</u>	<u>40 x 41</u>	<u>60 x 61</u>
1	0.11 (93)	0.27 (154)	0.52 (178)
3	0.48 (94)	1.10 (153)	2.15 (176)
5	1.05 (93)	2.51 (152)	
7	1.84 (93)		
full	3.24 (89)		

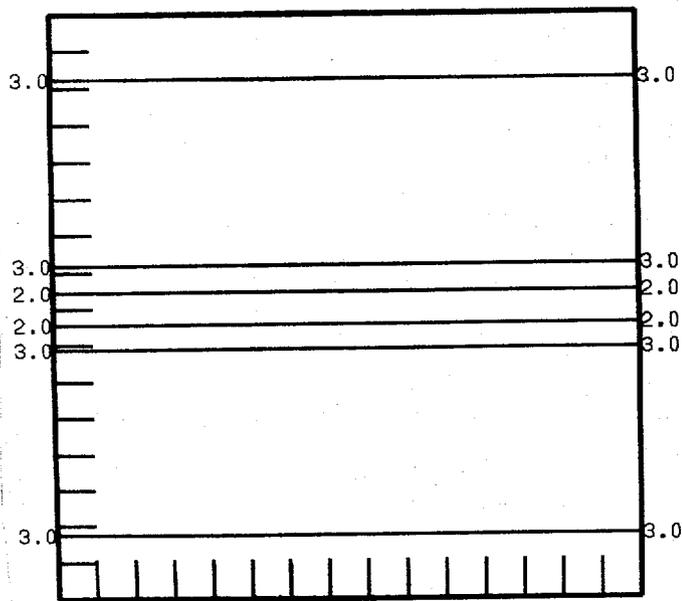
Table II. CPU seconds required by eq. (2.4b), as a function of bandwidth and model resolution. Numbers in parentheses are the observed megaflop (MFLOP) rates for this computation. Peak rate of the CYBER 205 is 200 MFLOPS.

<u>EXPERIMENT #</u>	<u>DATA DISTRIBUTION</u>	<u>MODEL ERROR COVARIANCE</u>	<u>BANDED APPROXIMATION</u>
1	parallel to basic flow	$Q = 0$	none
2	parallel to basic flow	$Q \neq 0$	none
3	perpendicular to basic flow	$Q \neq 0$	none
4	perpendicular to basic flow	$Q \neq 0$	$b = 5$
5	perpendicular to basic flow	$Q \neq 0$	$b = 4$
6	perpendicular to basic flow	$Q \neq 0$	$b = 3$

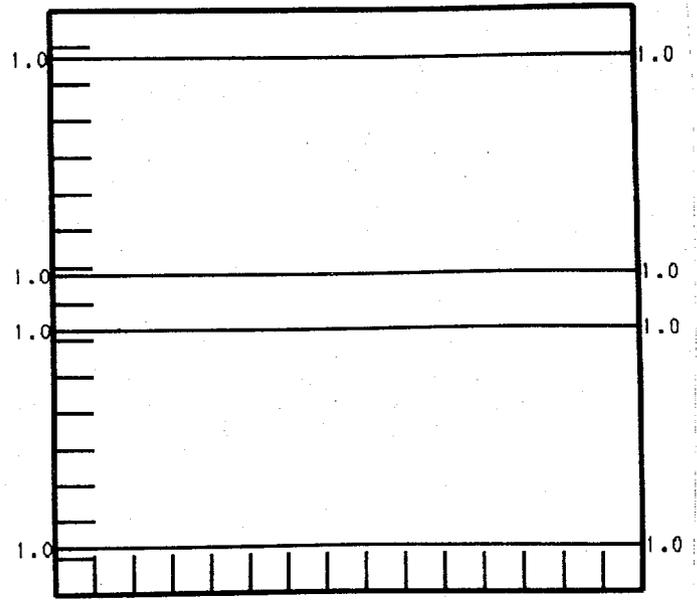
Table III. Summary of experiments discussed in Section VI.



(a)



(b)



(c)

Figure 1. Forecast error standard deviations for Experiment 1, (a) in the height $h = \phi / 10 \text{ ms}^{-2}$, (b) in the zonal wind u , and (c) in the meridional wind v , at 10 days. Contour interval is one meter in (a), and one meter per second in (b) and (c). Tick marks indicate grid points.

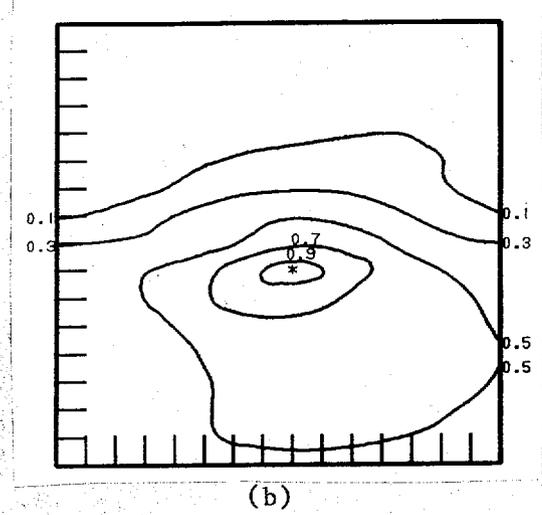
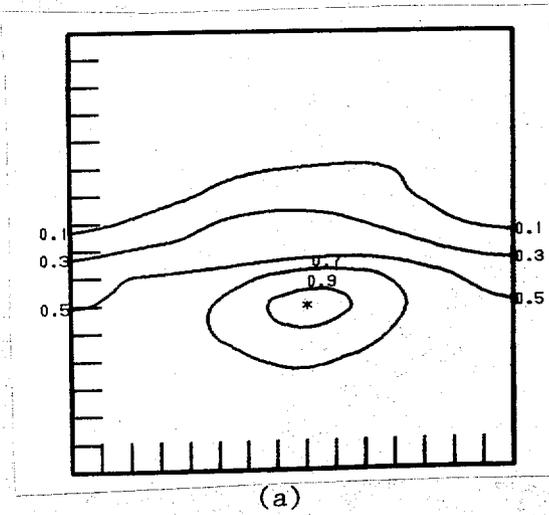
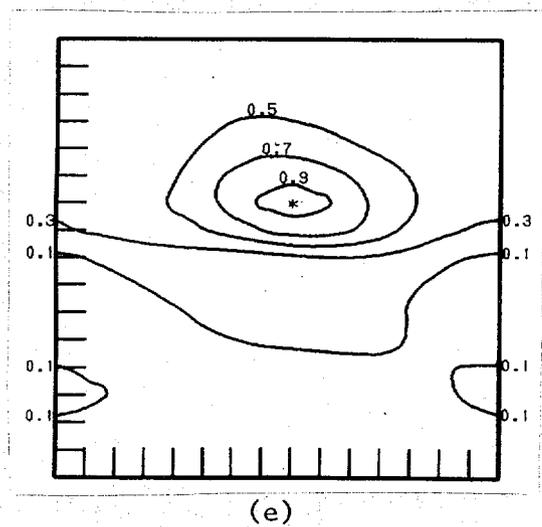
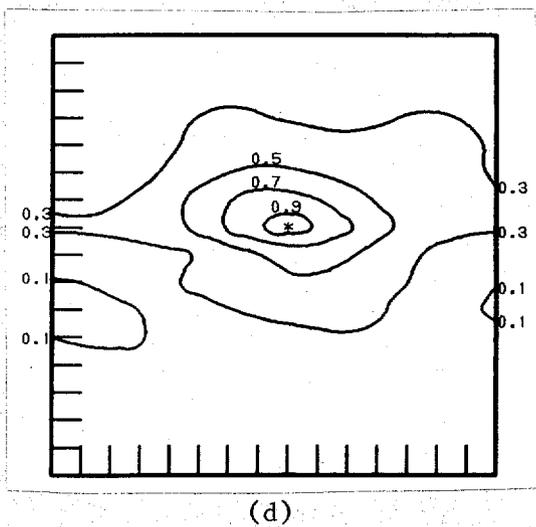
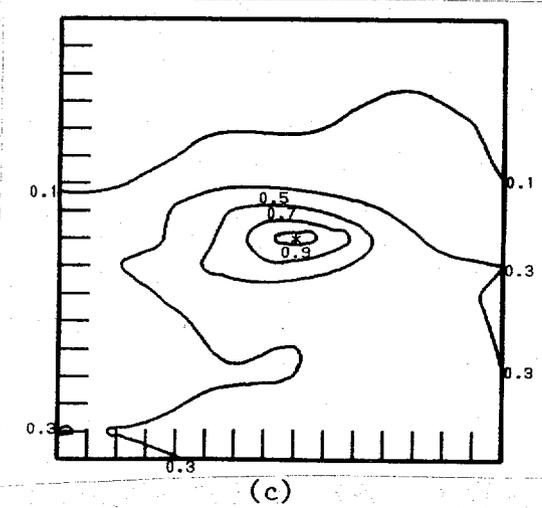


Figure 2. ϕ - ϕ forecast error correlations for Experiment 1 at 10 days, for base points (a) (9,7), (b) (9,8), (c) (9,9), (d) (9,10), and (e) (9,11). Asterisk in each panel denotes location of base point.



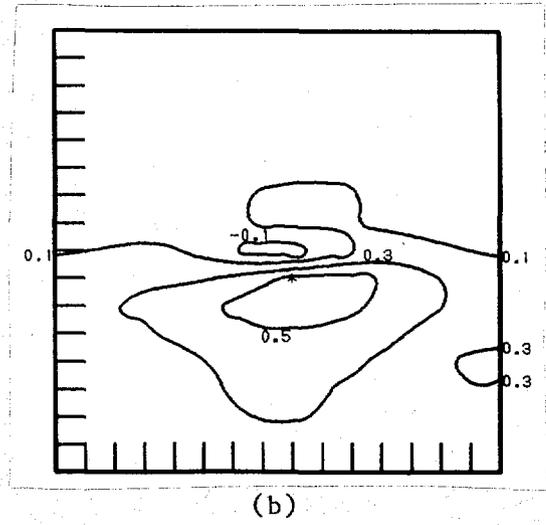
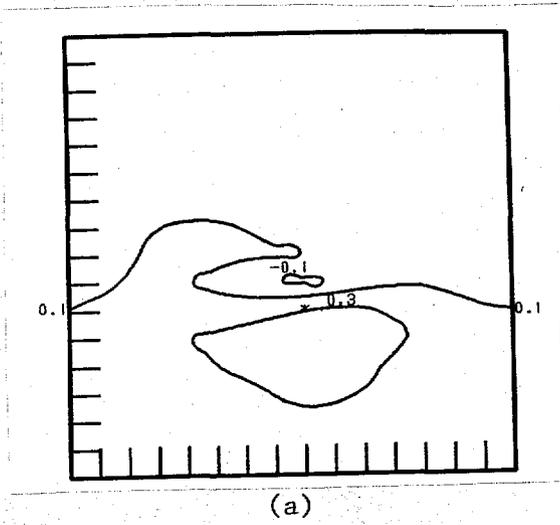
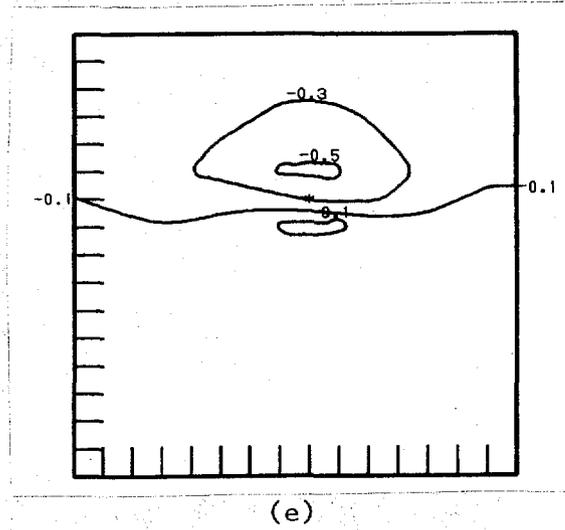
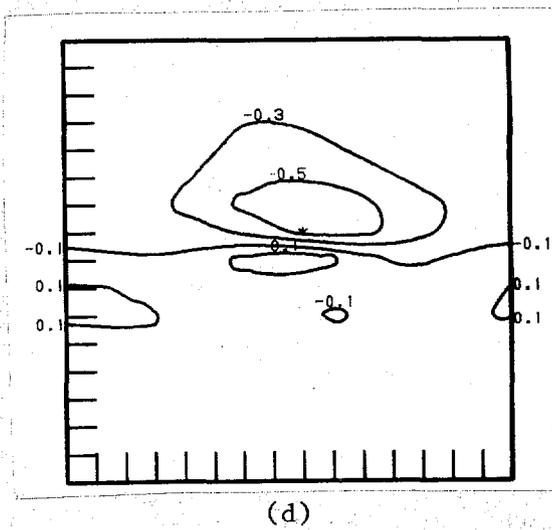
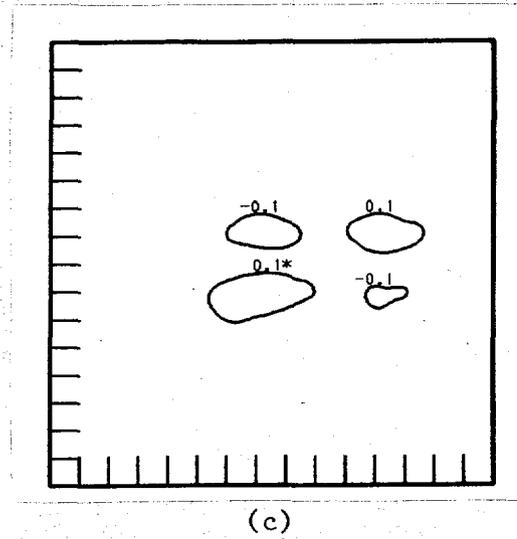
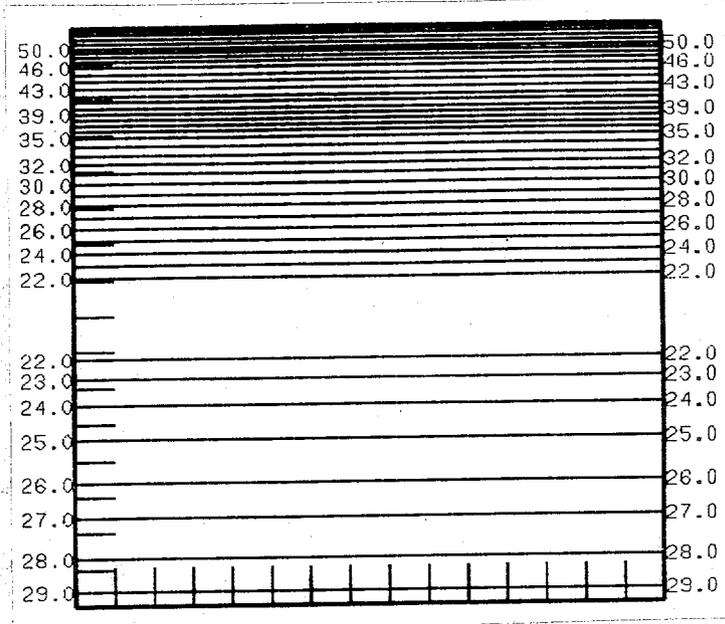
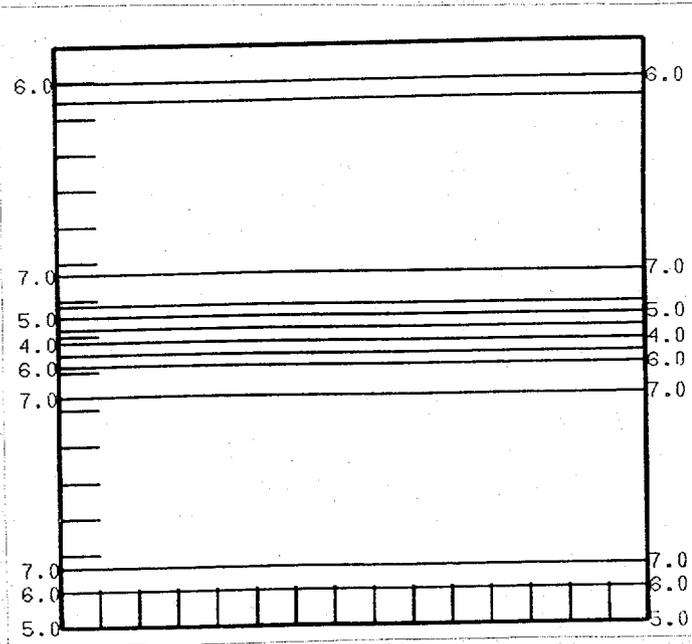


Figure 3. $u-\phi$ forecast error correlations for Experiment 1 at 10 days, for base points (a) (9,7), (b) (9,8), (c) (9,9), (d) (9,10), and (e) (9,11).

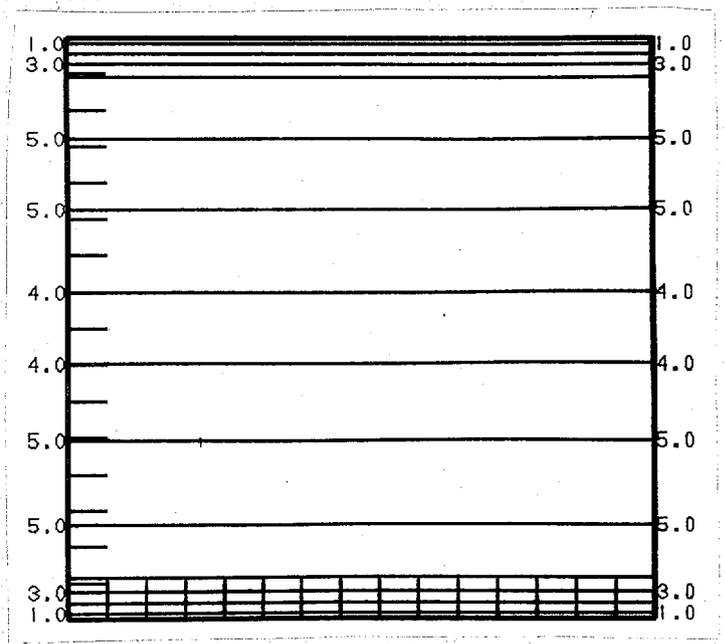




(a)



(b)



(c)

Figure 4. As in Figure 1, but for Experiment 2.

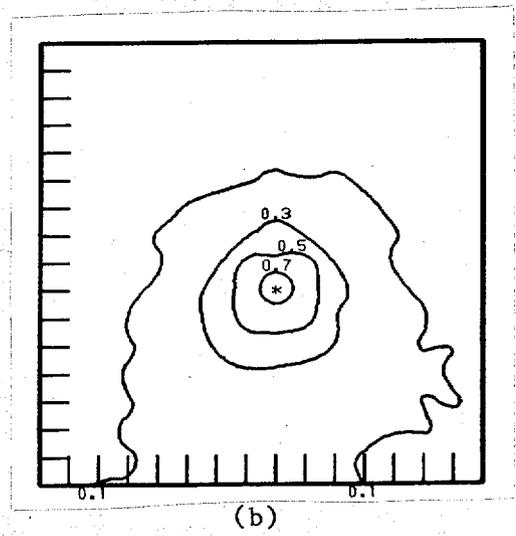
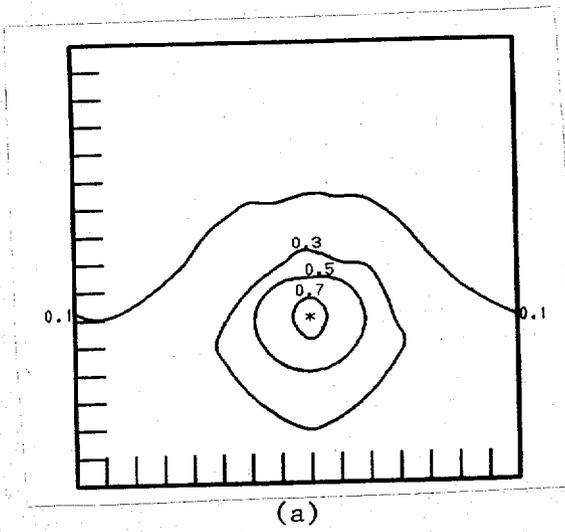
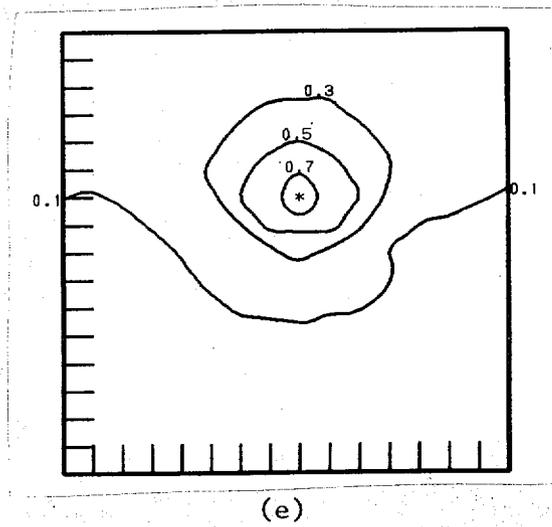
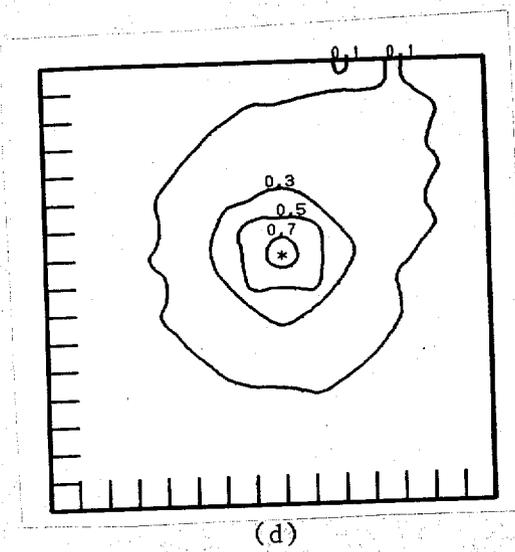
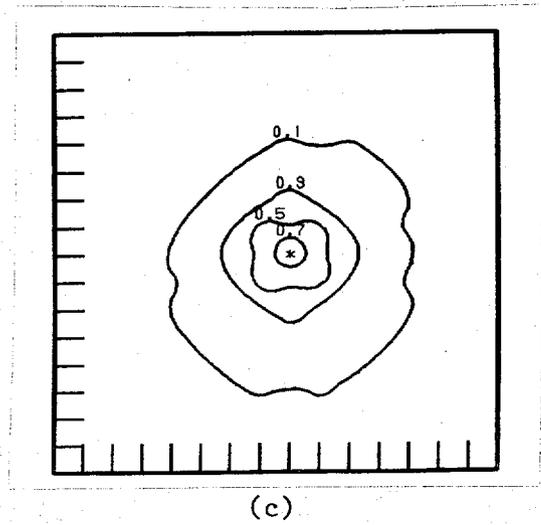


Figure 5. As in Figure 2, but for Experiment 2.



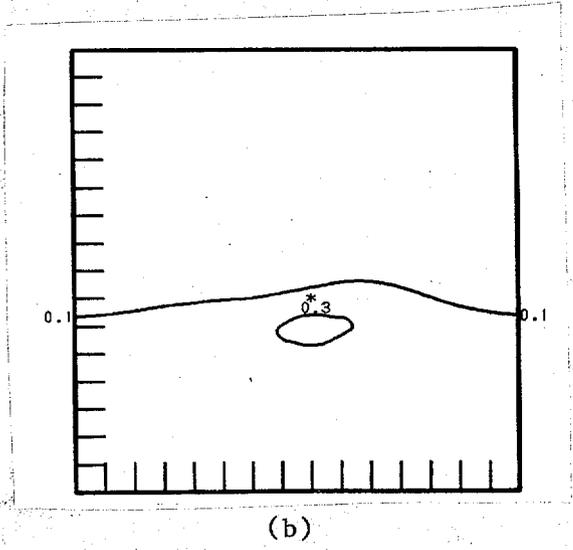
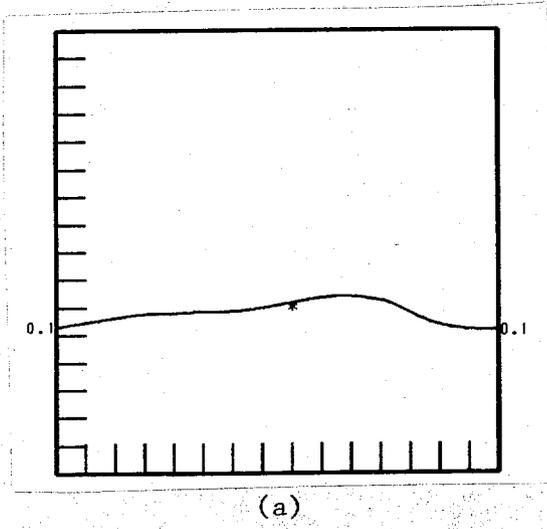
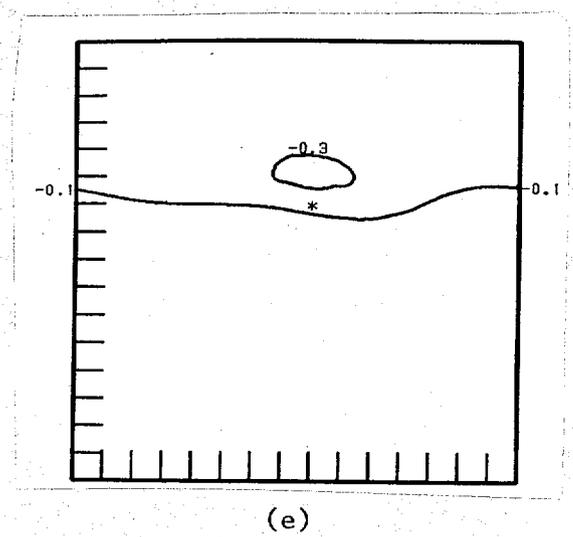
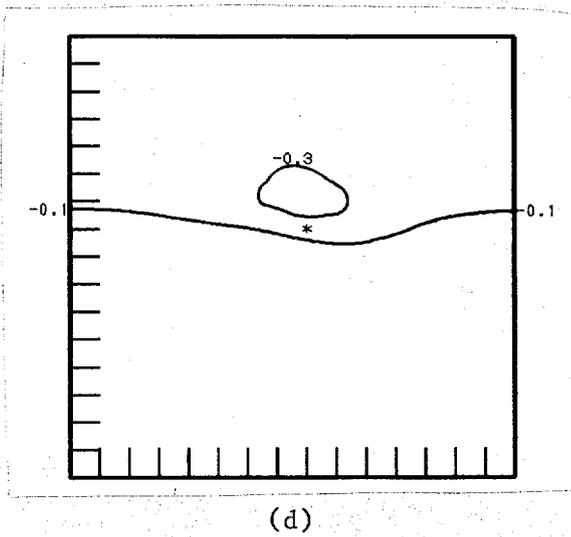
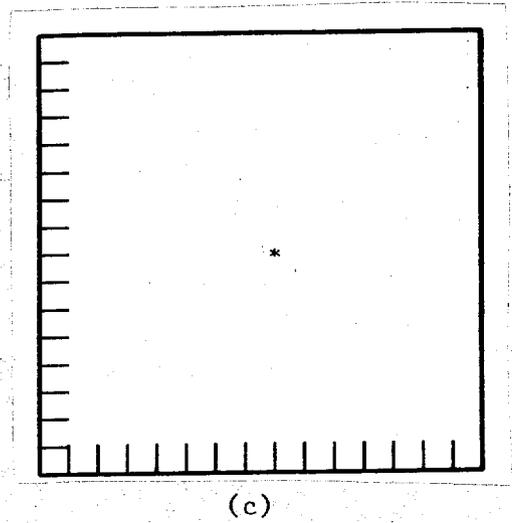


Figure 6. As in Figure 3, but for Experiment 2.



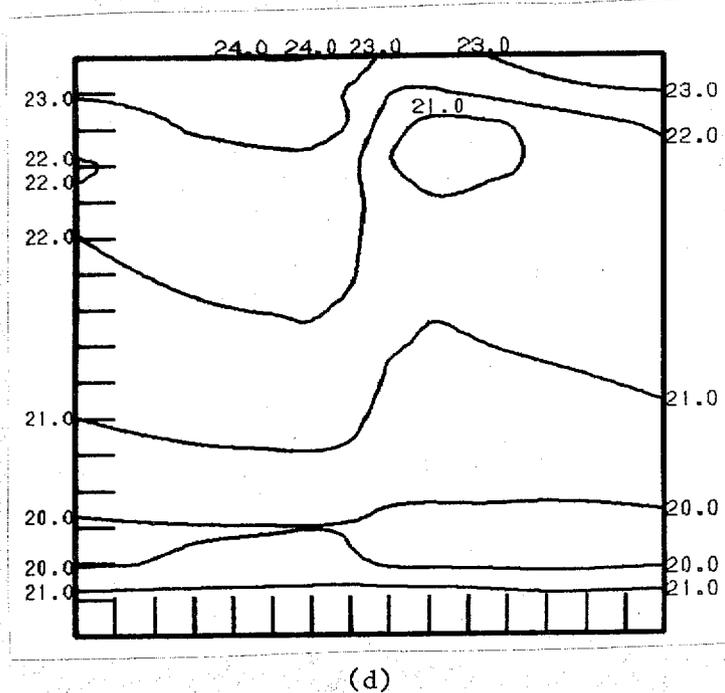
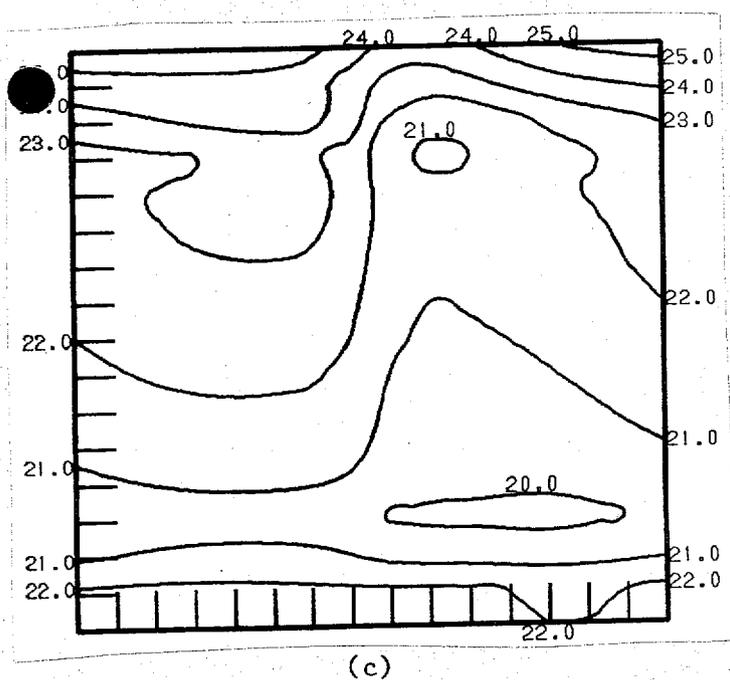
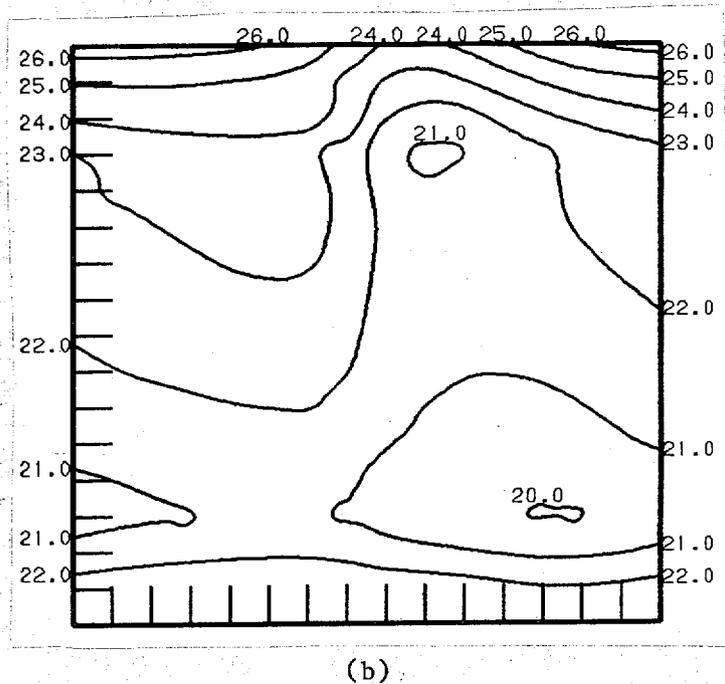
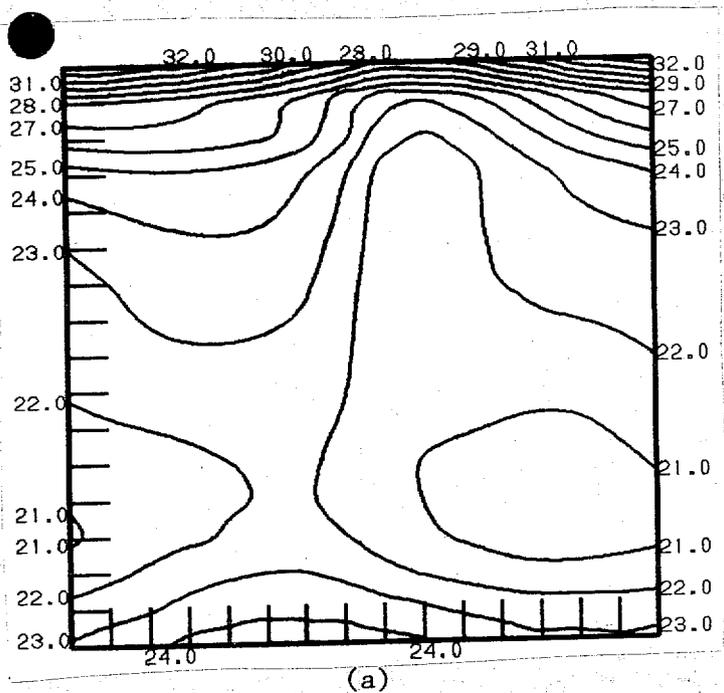


Figure 7. Forecast error standard deviations in the height field at 10 days for (a) Experiment 3 (no banded approximation), (b) Experiment 4 (bandwidth $\underline{b} = 5$), (c) Experiment 5 ($\underline{b} = 4$), and (d) Experiment 6 ($\underline{b} = 3$).

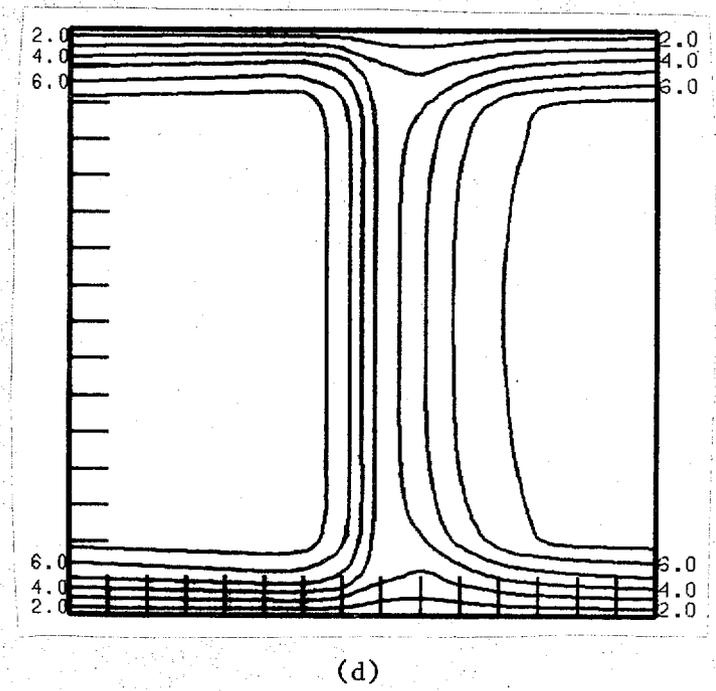
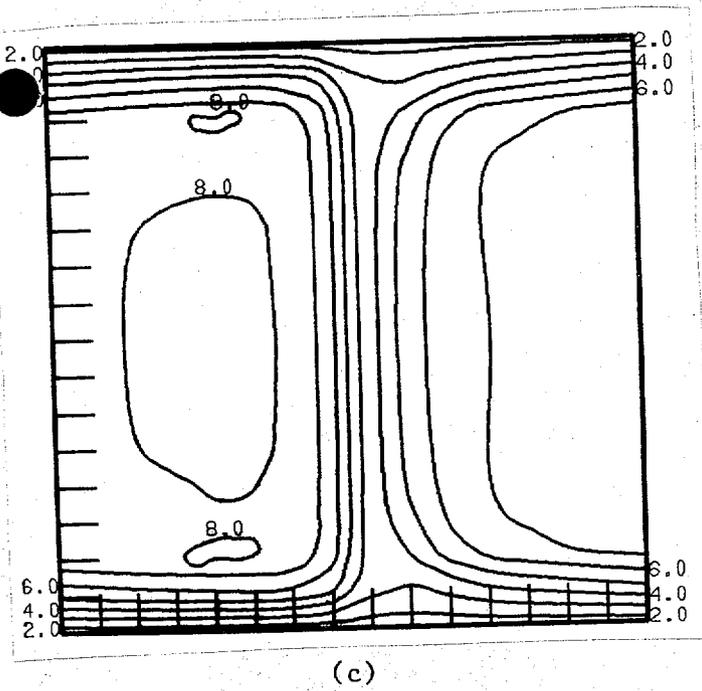
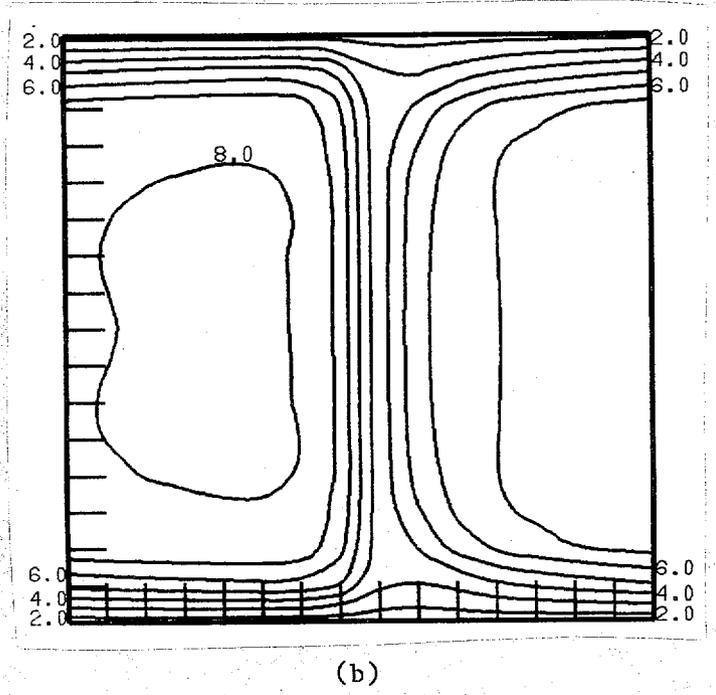
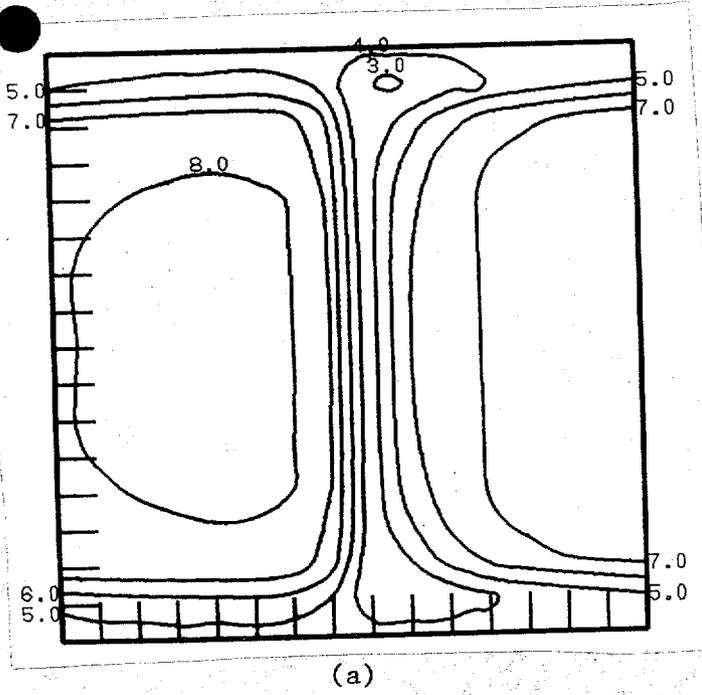
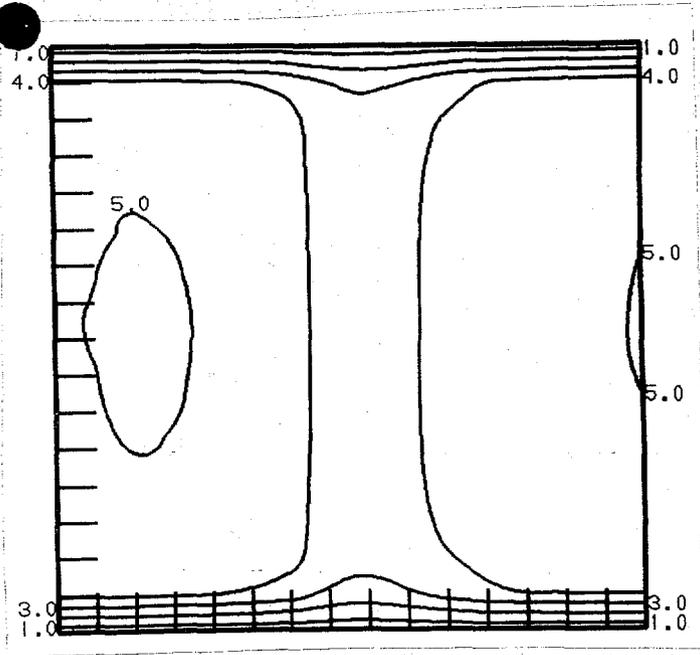
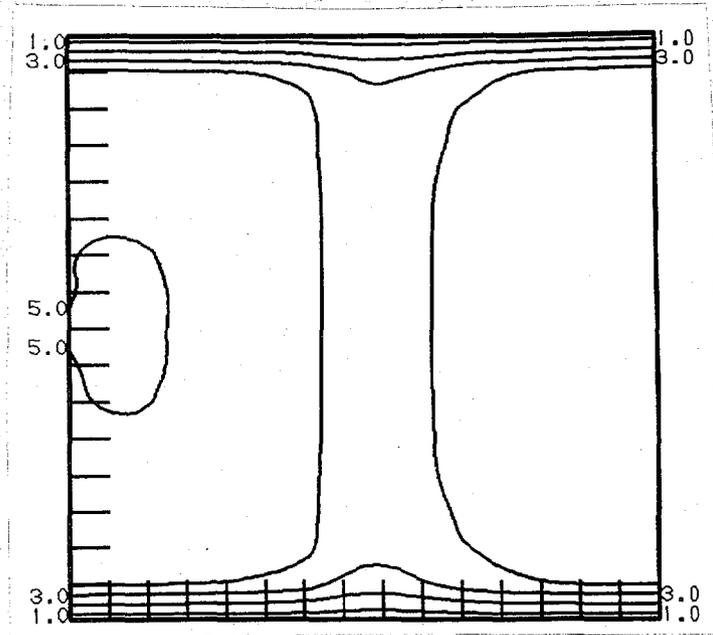


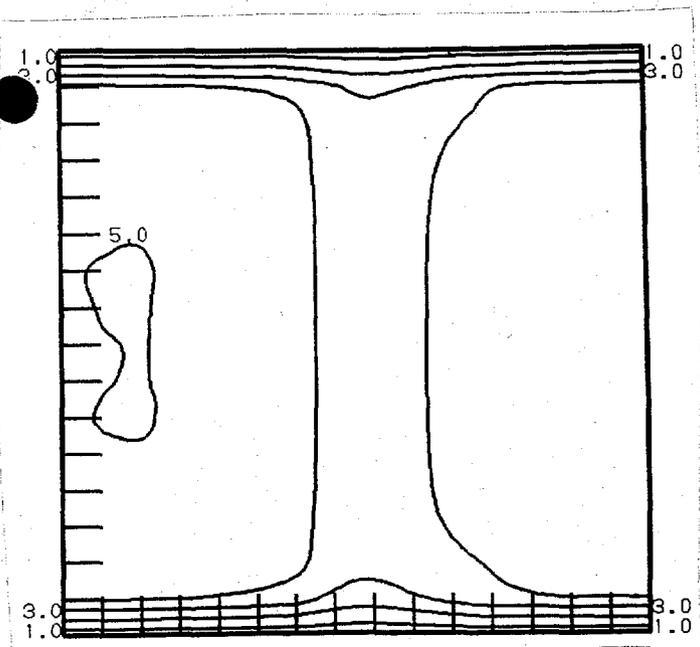
Figure 8. As in Figure 7, but for the zonal wind u .



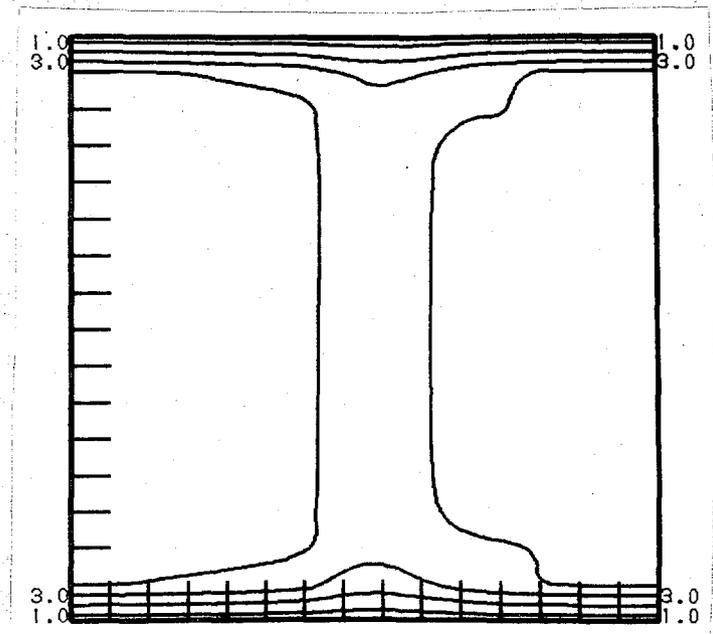
(a)



(b)

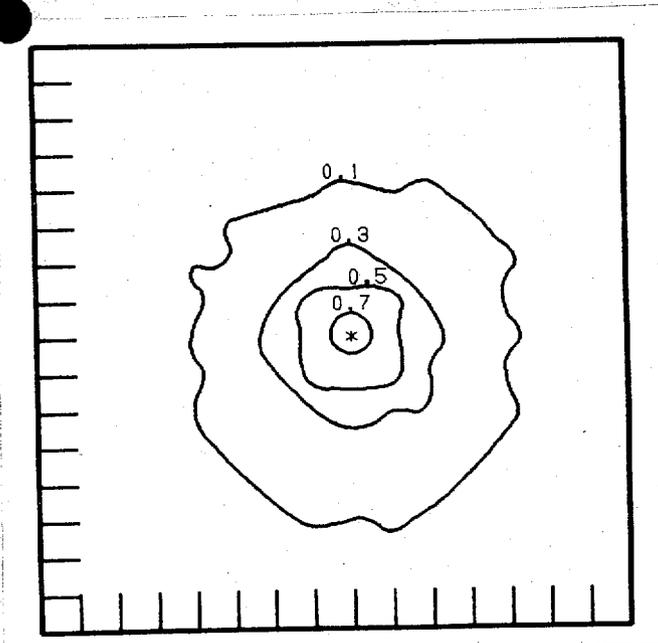


(c)

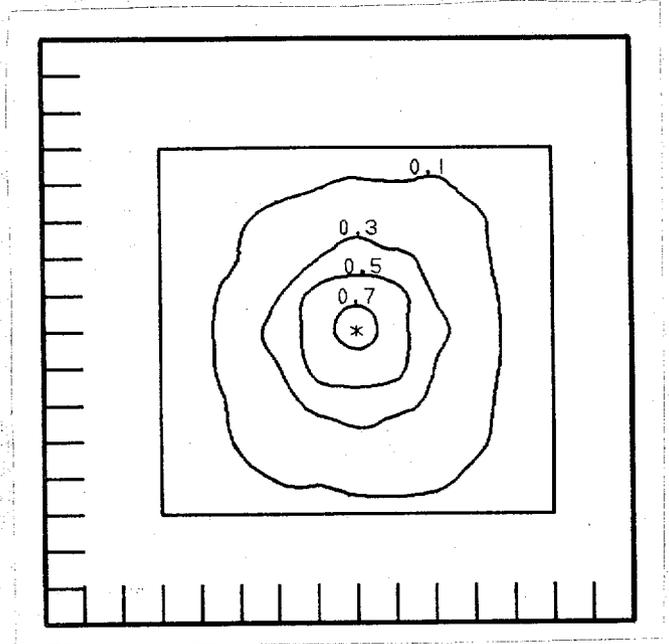


(d)

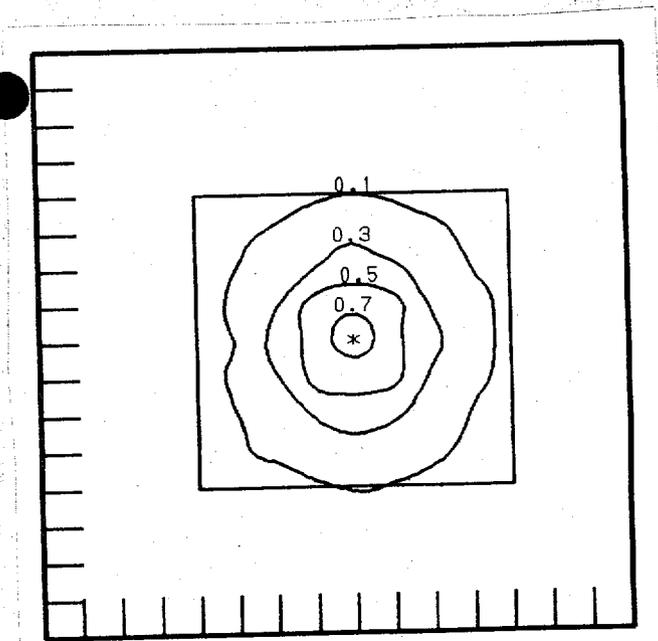
Figure 9. As in Figure 7, but for the meridional wind v .



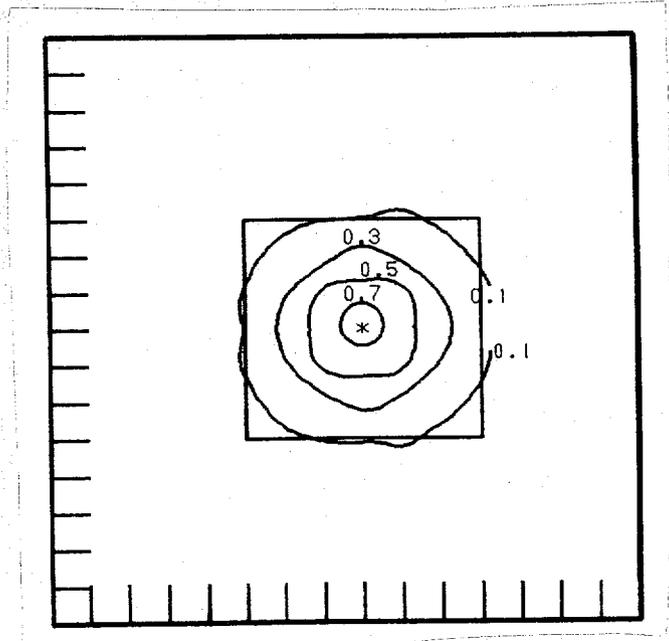
(a)



(b)

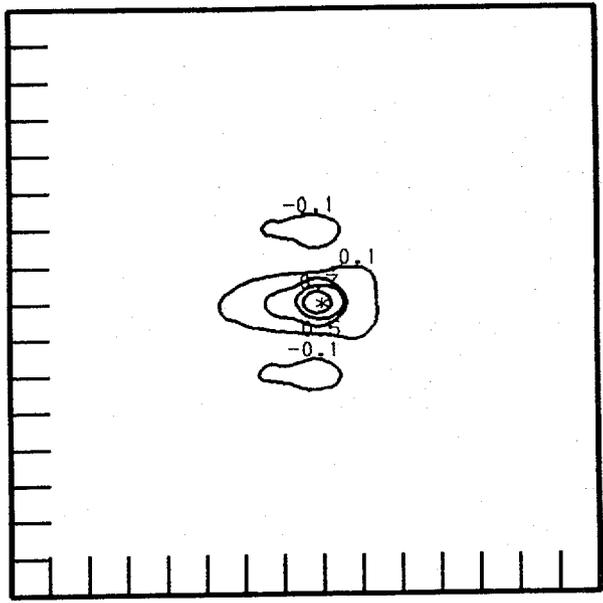


(c)

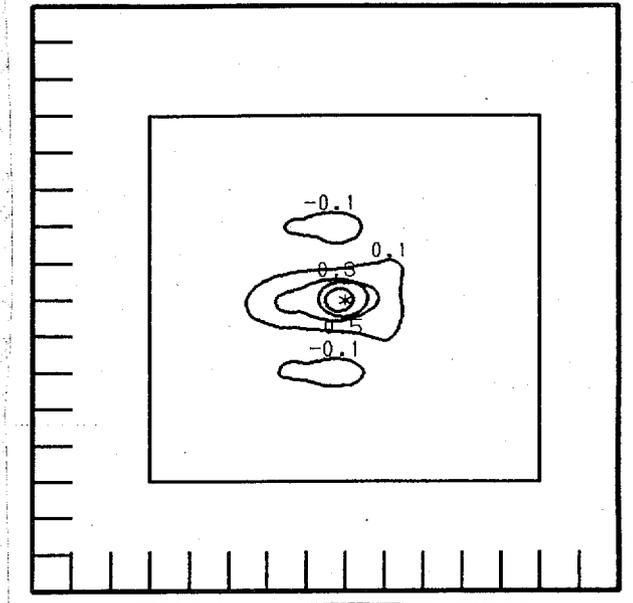


(d)

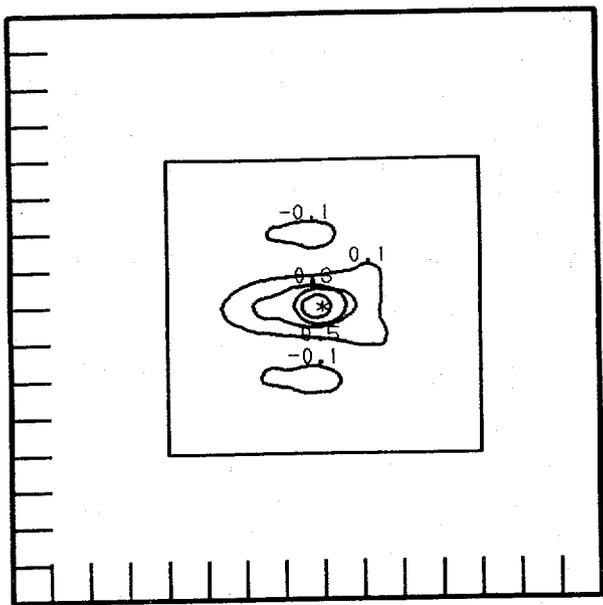
Figure 10. As in Figure 7, but for $\phi - \phi$ forecast error correlations at point (9,9). The squares in figures 10 b-d indicate the bandwidth. Contours extending outside the squares in Figures 10 c,d are an artifact of the contouring routine.



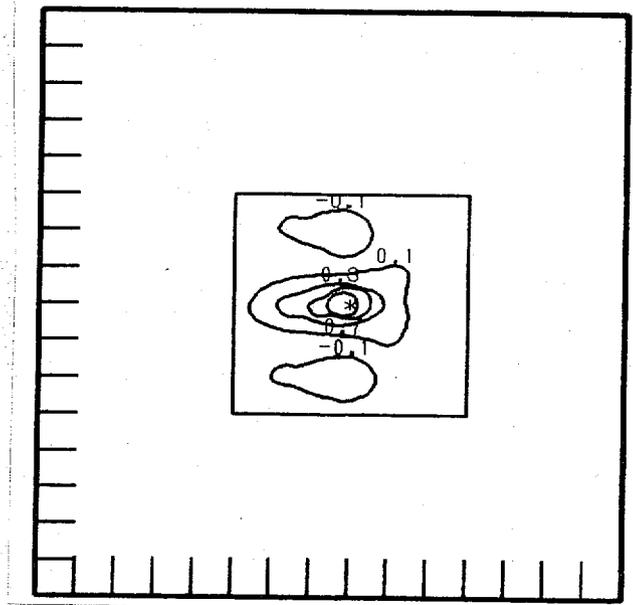
(a)



(b)

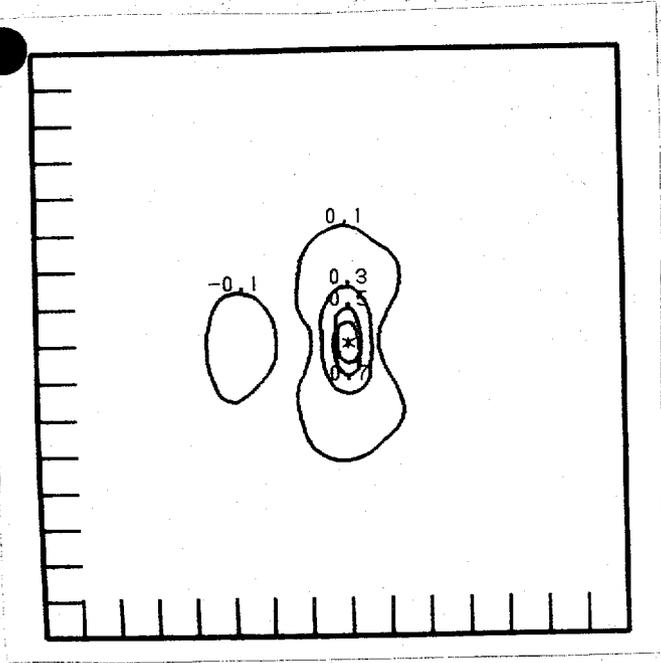


(c)

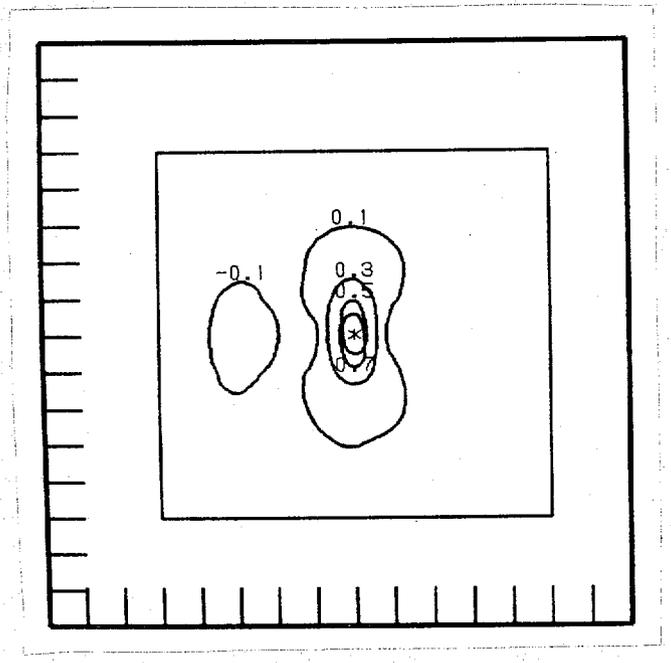


(d)

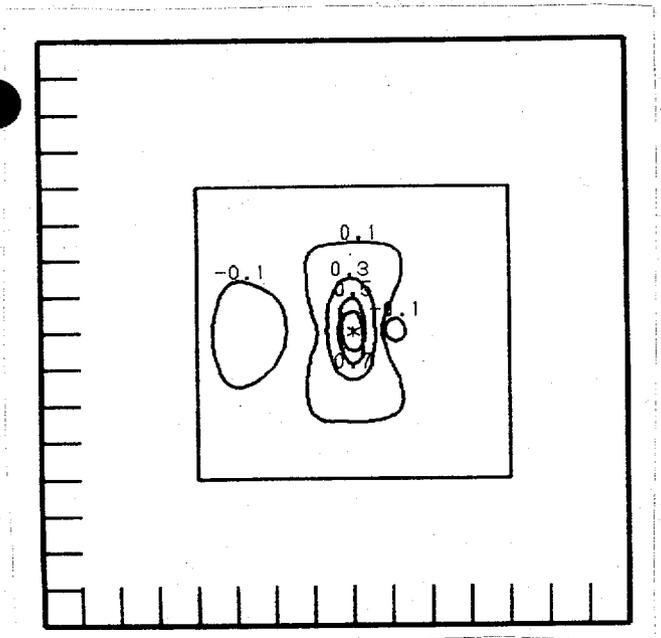
Figure 11. As in Figure 10, but for $u-u$ forecast error correlations at point (9,9).



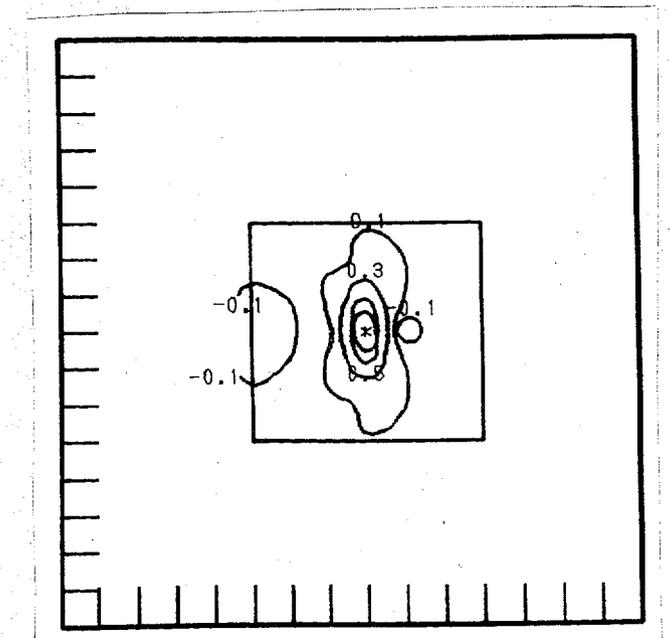
(a)



(b)



(c)



(d)

Figure 12. As in Figure 10, but for $v-v$ forecast error correlations at point (9,9).