Forecasting Research

Forecasting Research Division
Technical Report No. 108

TWO-DIMENSIONAL GRID-POINT VARIATIONAL ANALYSIS

by

Jon SHURLOCK & Andrew LORENC

25th August 1994

Meteorological Office
London Road
Bracknell
Berkshire
RG12 2SZ
United Kingdom

Forecasting Research Division
Technical Report No. 108

TWO-DIMENSIONAL GRID-POINT VARIATIONAL ANALYSIS.

Jon SHURLOCK & Andrew LORENC

25th August 1994

Forecasting Research
Meteorological Office
London Road
Bracknell
RG12 2SZ

This paper has not been published. Permission to quote from it must be obtained from an assistant director at the above address.

CONTENTS

1. INTRODUCTION

Variational analysis is a method for achieving a balance between certain constraints. In meteorology this translates to minimising a cost function that describes the distance from the analysed field to a background field and to a set of spatially and temporally varying observations. It is an iterative algorithm and in its full 4DVAR form not only allows a consistent global use of all observations (including satellites and non-linear variables), but also constrains the analysed field to be consistent with the forecast model. Unlike many other Met Services, the UK Met Office uses a grid point model, so whilst research using spectral models done at ECMWF and NMC (amongst others) can provide useful ideas, only research explicitly carried out on a grid can be of any concrete use to the UK Met Office.

2DVAR is a univariate analysis that, from a truth field, randomly generates a background field and a set of observations and, using their correlated errors, calculates an optimal analysis. It can be run on several grids - global, u-v (a staggered global grid with no points at the pole), Limited Area Model and polar stereographic - and at several resolutions (the highest of which is 288x217 for the global grids, and 91x91 for the LAM grid). It performs the analysis on the surface of the globe (hence 2D) and uses concepts from variational analysis (hence VAR). The code is written in FORTRAN 77 with PP package additions, and can be run on either the HP work stations or on the HDS mainframe. The actual program is a highly modular collection of subroutines, all controlled by a top level routine. This enables the user to have complete control over the type of experiments by overriding default values set up in common blocks.

The purpose of conducting these experiments is to show that a grid-point version of variational assimilation is feasible, and to provide some pointers for the direction in which the operational 3DVAR should go. There are therefore two main objectives for the 2dvar project: to show it works; and to develop an iteration strategy for minimising the time taken to reach convergence. By using 2DVAR we have a simple model that is easy and quick to program but that nevertheless contains most of the important features of 3DVAR.

2. SHOW THAT 2DVAR WORKS

For the 2DVAR prototype to work, it must be shown that it improves the analysis, and that the results that it provides are consistent with data assimilation. The method used for this experiment was to define a truth field, and then to generate observations and a background field using correlated errors and a random generator. We therefore know what the analysis should be aiming for and can compare it with the truth field by looking at the Root Mean Squared Differences between the two. We use the background field as the initial field, although there is no need for this to be the case.

2.1 RMS differences

One way to find out if 2DVAR works is to compare the root mean square value (RMS) of the difference between the background field and the truth field, with the RMS of the difference between the analysis and the truth field. We would expect the RMS of the difference of the analysis field to be less than that for the background field; we expect 2DVAR to improve the analysis by incrementing the analysed field closer to the truth field. We can also predict that the more observations we include in the assimilation the better the analysis should be; by including more observations we introduce more information into the assimilation. The RMS is equal to

$$RMS = \sqrt{(\bar{x})^2 + s^2}$$

so since x bar is small (random variations around zero tend to cancel each other out) the RMS of the differences should start off equal to the standard deviation s (which is equal to 1.0 in this case). We therefore expect that the RMS difference, for an analysis using a zero truth field and correlated errors set to 1.0, will asymptotically decrease from 1.0 to zero. We therefore have some criteria by which to judge 2DVAR.

A set of experiments were run on a global grid at high resolution with a varying number of observations, ranging from 0 to 20 000. The correlated errors were set at 1.0 and a zero truth field was chosen, although there are plans to use real fields. The results are shown in Fig 1. The RMS difference decreases from 1.0 (the value of the standard deviation), and reaches a value of 0.33 with 20 000 observations. A separate experiment using a similar setup has shown that with 50 000 observations the RMS difference is reduced to .26 and with 100 000 observations it is reduced to 0.23. We can therefore say that 2DVAR does improve the analysis, although it is heavily dependent on the number of observations (as is any assimilation scheme).

2.2 Penalty Function

Another way to verify that 2DVAR works is to study how the penalty function behaves against the number of inner iterations. We expect the penalty function to decrease at each new iteration of the descent algorithm, and finally reach convergence. Fig 2 shows the results of an experiment on a u-v grid with 100 observations. (In fact it is easier to count function evaluations than iterations; the graph is plotted against this. Each iteration has one or two penalty evaluations). The penalty function decreases rapidly then slowly decreases to a minimum where convergence is reached. The background penalty function term starts at zero because the analysis starts at the background field, as in a real assimilation cycle a previous forecast would be used. The observation penalty term decreases until balanced by the background penalty term. Fig 3 shows what happens to the norm of gradients of the penalty function and penalty terms. The total gradient decreases to zero, whereas the two component terms converge to equal but opposite values.

2.3 Observation values

If only one observation of value 1.0 is analysed on a zero background field and diag(B)=diag(O+F)=1, then since

$$J = J_b + J_o$$

$$J = \frac{1}{2}(x_b-x)^T B^{-1}(x_b-x) + \frac{1}{2}(y_o-K(x))^T(O+F)^{-1}(y_o-K(x))$$

a value of 0.5 is expected to be returned at the observation. Fig 4 shows an experiment with one observation at the equator, using a correlation scale of 400km. A value of 0.506 is returned which confirms expectations. The background error correlation μ, between two points a distance r apart is given by

$$\mu\,(r) = (1+\frac{r}{s})\exp(\frac{-r}{s})$$

whose spectral response is

$$S_{SOAR}(k) = \varepsilon\,^2_b\frac{4s}{(1\ +\ k^2s^2)^2}$$

where s is the correlation scale, ϵ^2_b is the variance and k is the wave number of the appropriate wave component. If s is taken to be 400km, and μ is taken to be 0.4, then r is given as approximately 810km. By inspection from Fig 4 a value of about 800km is evident.

2.4 Boundary points and polar points

Because the Met Office uses a grid point model (so the model for the LAM is the same as for a global grid), there are always going to be problems with: the poles, where the pole is represented by 288 points each with a different value; the boundaries of a LAM grid due to the constraint that edge increments must be zero.

The LAM grid has boundaries and these present a problem for 2DVAR. It is desirable that the boundaries are constrained to remain fixed, that is that boundary increments are zero. A simple way is to remove the boundary points before the analysis, and then add them on when the analysis is finished, using some appropriate ramping function to fill in the missing increments. This is schematically shown in Fig 5.

The results of an experiment run with 1 ob at the boundary of a LAM grid can be seen in Fig 6. The ramp has constrained the boundary to zero. In other words the observation has only minimal effect with a maximum value of 0.069 as opposed to the expected value of 0.5.

Another way of dealing with the boundary is to use a spectral filter. The latter has the advantage of automatically keeping boundary increments to zero, although this can also be achieved by using an S-shaped ramp (see fig. 5). This works with a double sine transform and means that the control variable now takes the form of spectral coefficients of a spatial field. This has been implemented in 2DVAR using the spectral response given by eqn 5, but it is not yet working correctly; analysing 1 observation on a LAM grid gives a totally unacceptable result. This can be seen by comparing fig 7 (analysed with normal recursive filter) with fig 8 (analysed with spectral filter).

The pole presents a different problem. Because on the global grid there are 288 points at the pole, each of which is slightly different, the 2DVAR program averages them out.

There is another component to this which is the filter. The filter is used to precondition the control variable. It is a two pass recursive filter that approximates a SOAR. Lorenc has constructed a routine which is its inverse (except at the poles). This could introduce small scale noise at the pole into the analysis.

Another point to bear in mind is that the filter treats a grid as if it were regular, and so the filter on a sphere has been designed to be uniform. This has not been completely achieved, and points near the pole do not get exactly the desired filtering. A good test of the 2DVAR code would therefore be to analyse a grid with just one observation at the pole. Fig 9 shows the resulting analysed field, and Fig 10 shows the same field interpolated to a polar stereographic grid, and Fig 11 shows the same experiment but analysed on a polar stereographic grid. It can be seen that whereas the analysis on the global analysis interpolated to the stereographic grid is circular (as the correlation function should be), the polar stereographic grid is rather square. This is a result of the analysis dragging the observation around the top of the global grid.

The fact that the filter has problems at the poles can be seen from Fig 9 It can be seen that the value attributed to the ob is 0.649 but equation 3 predicts a value of 0.5. This is compared to Fig 4 where the value is 0.504, very close to 0.5. The problem is that the filter does not weight observations properly at the pole. This is not seen to be a problem since the weighting depends on the background error variance, which is imprecisely known.

Another factor is the interpolation routine that is used to calculate the w-field value at an observation. Fig 12 shows the analysis of one observation at the pole for a u-v grid. Although the observation was placed at the pole, the analysis has placed it slightly off centre. This is a result of the way the routine was coded, which extrapolates beyond the boundaries of the grid, and if considered a serious problem can easily be rectified, by interpolating across the pole.

3. ITERATION STRATEGY FOR 3DVAR

Because 3DVAR, and even more so 4DVAR, is so expensive in computer time, there is a necessity to cut down on time. There are two main ways of doing this: the first is to use the multigrid approach where the initial work is done on low resolution grids; the second is convergence interruption where the descent algorithm is stopped before convergence is reached, but after a good approximate solution has been achieved.

3.1 Multigrid

One possible way to decrease the variable transformss' and background term's contributions to computer time is to use the multigrid approach. This involves working out an increment to minimise the cost function on a low resolution grid (which should be a good approximation to the full field answer), then interpolating the resultant field to a higher resolution and adding to the previous high-resolution estimate. This process can be iterated in an outer loop until a final solution is achieved on the full field. Since a lot of the donkey work is carried out on the low resolution grids, an approximate answer is obtained at low cost. The final solution on the full field then just adds fine detail and so should take little time to reach convergence. We thus have inner and outer iterations. Outer iterations are performed over resolution changes, whereas inner iterations are performed at one resolution and represent the steps needed by the descent algorithm to reach a solution. Because the analysis is for the most part carried out on low resolution grids, there must be some way of preventing aliasing, that is the loss of small scale features by undersampling the full field with a lower resolution grid. This works by working in increment space, so an analysis is produced of the increments needed to be added to the background field at low resolution. This field is then interpolated back to full resolution for use in the persistence background term for the next outer iteration.

There are two major parameters that can be changed: the number of outer iterations ie. the number of resolution changes, and the reduction in resolution of the low resolution grids. The latter is obviously in need of careful tuning but general schemes such as 1, 1/2, 1/4, 1/8, 1/16 or 1, 1/2, 1/3, 1/4, 1/5 etc can be used to gain a rough idea of what is needed. An important point is that if the gridlength of a reduced resolution grid is more than the filter scale then essentially any observation is not spread, and remains at the gridpoint. Assuming a filter scale of 400km,if we have a global grid 288x217 with a gridlength of 100 km at the equator then we cannot have a grid reduction larger than 1/4. For a global grid 96x73 (for use in climate models) we cannot reduce the grid by less than 2/7, and for a LAM grid 43x43 cannot reduce less than 1/4 . Tests were conducted on global, u-v at high and medium res, and on LAM at high, running through 1-4 outer iterations for each scheme. The results for the LAM grid are plotted in Fig 13. Similar results are obtained for global and u-v grids. One can see that although the number of inner iterations increases with increasing numbers of outer iterations, there is a minimum of computer time that comes after 2 outer iterations. For the U-V grid a scheme of 2 outer iterations with a resolution reduction of 1/4 for the initial iteration produces the quickest time, whereas for the global grid a scheme of 3 outer iterations with reductions of 1/3, 1/2 for the first two iterations works best. In all cases there is a strong correlation between the total time taken to reach convergence and the number of inner iterations in the final outer iteration. With 3DVAR, the background term is even more important since there are (in the global model) 19 levels to contend with and so I would expect the reduction to be even more dramatic.

If a graph, showing results from an analysis with 1000 observations, of purely penalty function against number of inner iterations is inspected as in Fig 14 then it is obvious that there is a rather large spike in the penalty function that corresponds to the beginning of the last outer iteration. Although this reduces fairly rapidly it is nonetheless a bit worrying. The reason it occurs is that when interpolating to a finer resolution, not only is the number of background contributions to the penalty function increased, but also there is a mismatch between scales due the coarser resolution being unable to display fine detail. Another reason why there is a spike is the interpolation routines used for interpolating from one grid to another. Experiments were run using instead a spectral interpolation routine. This was much slower but it did reduce the spike. One would expect the spike to be bigger for a bigger resolution jump and this appears to be the case. Running an analysis on a global grid at high resolution for a jump of 1:2 the spike has a maximum value of 1.5 10^8, whereas for a final jump of 1:4 the spike's maximum is 5.3 10^{11}.

Another method to reduce computer time is to use some type of convergence interruption of the descent algorithm. From Fig 2 it can be seen that after approximately 4 inner iterations there is no substantial reduction in the penalty function; a good approximate solution has been found and the computer is spending a lot of time to put the finishing touches to its solution, and so reach what the descent algorithm considers to be convergence. We should remember that even if the descent algorithm satisfies convergence criteria it is unlikely that an exact solution has been found (due to the descent algorithms own accuracy criteria). This leads to the idea of stopping the algorithm prematurely by introducing artificial convergence criteria. This is obviously only desirable in early outer iterations, and for the final outer iteration the descent algorithm should be stopped after different criteria (since there is evidence that the algorithm's own criteria introduce over-fitting of the observations to the analysis). Since we are only looking for approximate solutions to the initial outer iterations, the fact that convergence has not quite been reached should not be a problem. Several schemes can be designed, but I have used two: the first stops the descent algorithm when the total penalty function is less than some value, normally a fraction of the initial cost function; the second stops the descent algorithm when both components of the penalty function are less than a fraction of the total current penalty function. However it was found that convergence interruption had little effect on the final convergence, perhaps because in the earlier iterations it finished before a satisfactory solution was found. In this case more investigation might be needed.

3.2 Precision

Most of the results presented in this paper were obtained on the FR HP computer system, using the NAG routine E04DGF routine. On the HP system, only DOUBLE PRECISION versions of NAG routines are available, while the recursive filter was only available as a REAL version, so some interface copying was necessary, and the resulting effective precision was probably only 32bit. Some tests have been performed on the HDS mainframe, using a single precision version of the NAG routine, with acceptable results.

There is a significant loss of precision in the inverse filter at high resolution. This manifests itself near the poles, where points come very close together, and in high resolution limited area experiments. Tests that the inverse filter routine is a true inverse can fail in these cases. For the global case the method avoids problems due to loss of precision near the pole by transforming to a control variable on a quasi-uniform grid, having the same latitude rows, but fewer points per row near the pole. In the limited area case, problems are avoided as long as the analysis grid is not very much finer than the correlation scale used in the filter.

There were some problems, probably due to loss of precision, in non-standard configurations (e.g. without preconditioning, or without the quasi-uniform global grid). For the default method it seems that 32bit precision is sufficient.

3.3 Descent Algorithms

The NAG routine E04DGF routine uses a pre-conditioned, limited memory quasi-Newton conjugate gradient method. This gave acceptable results, but was inconvenient since source code was unavailable. Its inbuilt gradient testing facility was found to be unreliable, giving false alarm failures - we ended up coding our own (TESTOBJFUN).

After doing many of the runs we obtained the source code for routines M1QN3 and N1QN3 from the MODULOPT library from Jean Charles Gilbert at INRIA (Gilbert and Lemaréchal 1989). (This is the code used by ECMWF in their variational analysis project). In order to compare with the NAG routine, we implemented the double precision N1QN3. When properly set-up, specifying a sensible initial step-length, the INRIA routine converged more rapidly than the NAG routine, as illustrated in figure 15. It is also more flexible in its storage usage, and in preconditioning options. Moreover we have the source code available, allowing greater portability, and easy implementation in fortran90.

4. CONCLUSIONS

The work described here has shown that variational analysis for a grid-point model, using a filter to define the background term's constraint on smoothness, is a viable approach. The recursive filter used for the experiments is designed to match a "SOAR" covariance function. The filter gives results that differ from those implied by an optimal interpolation using the SOAR function, in the shape of the influence of a single observation, which is rather square, and in the weight given to an observation at the pole, but the differences are probably less than our uncertainties in the true covariance function. Practical methods for dealing with the boundaries of a limited area grid can be defined.

The "multigrid" approach, doing some early iterations at a lower resolution, works. This technique can potentially make savings in the computational cost of the variable transforms and background penalty term calculations, and also (in 4DVAR) in running the perturbation forecast and adjoint model. However it does not make a saving in the cost of the observational penalty term, whose cost is grid-independent. Nor does it make a saving in the number of iterations needed, since the method is already well conditioned. (This is in contrast to other applications of the multi-grid approach, which converge for small scales faster than large scales). It is not possible from this 2DVAR system to make firm recommendations for 3DVAR or 4DVAR, since the cost ratio of grid dependent and grid intependent calculations will be different, as will the conditioning.

For implementation at normal practical resolutions 32bit precision is sufficient for most calculations; we can code the VAR system using the default REAL arithmetic. However tests at very high resolution should not be expected to always work. This should not be a practical limitation - we would avoid an analysis grid that is much finer than the correlation scale by using the multigrid approach.

We should use the INRIA descent algorithm M1QN3.

5. FURTHER WORK

Some experiments demonstrating the advantage of preconditioning by using a transformed control variable have been perfomed (Lorenc and Griffith 1994).

The 2DVAR system has been used to test the use of non-Gaussian penalty functions to do implicit observational quality control (Ingleby 1994).

The recursive filter needs to be compare with the spectral filtering approach, which is implicit in variational analysis schemes at ECMWF and NMC which use spectral models. The spectral approach offers more flexibility in spectral response, and (using spherical harmonics) a homogeneous response on the sphere, avoiding the polar "fixes" of the recursive filter. However it is likely to be more expensive. A double sine-transform filter has been coded for the limited area mode of 2DVAR, but not yet debugged.

The recursive filter also allows variable filter scales, e.g. allowing a different scale to be used for the tropics. This option has not yet been tested.

The NMC SSI variational analysis system (Parrish and Derber 1992) allows for correlated observational errors in satellite data by putting the on a regular grid and filtering them. Lorenc (1992) tries some similar ideas. On the other hand the ECMWF system only allows for correlations within one processing batch of observations, doing an explicit matrix inversion. The 2DVAR system could be used as a testbed to decide which approach to use in VAR.

References

Gilbert, J.Ch., and C.Lemaréchal 1989
"Some numerical experiments with variable-storage quasi-Newton algorithms", *Mathematical Programming*, **45**, 407-435.

Ingleby, N B 1994
"Treatment of gross errors within the 2-dimensional variational analysis testbed" Forecasting Research Working Paper No. 166, June 1994.

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

Lorenc, Andrew and Anne Griffith, 1994
"Experiments with the preconditioning in the 2DVAR model." FR Tech Report 109

Numerical Recipes; Press, Flannery, Teukolsky, Vetterling; Cambridge University Press (1989)

Parrish,David F., and Derber, John C. 1992
"The National Meteorological Center's Spectral Statistical Interpolation analysis system" *Mon.Wea.Rev.* **120,** 1747-1763

FIGURES

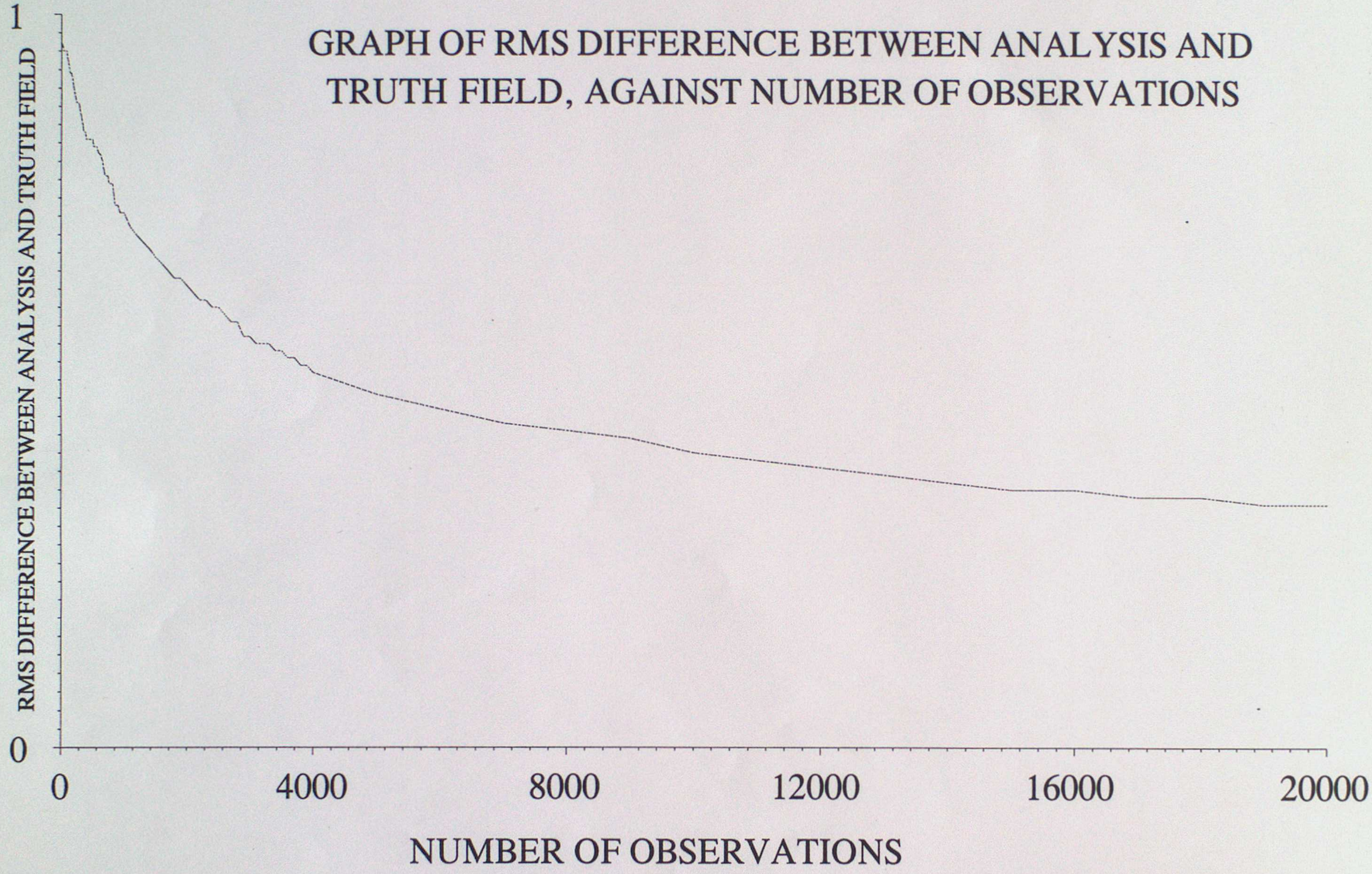

GRAPH OF RMS DIFFERENCE BETWEEN ANALYSIS AND
TRUTH FIELD, AGAINST NUMBER OF OBSERVATIONS
RMS DIFFERENCE BETWEEN ANALYSIS AND TRUTH FIELD
1
0
0
4000
8000
12000
16000
20000
NUMBER OF OBSERVATIONS

Figure 1

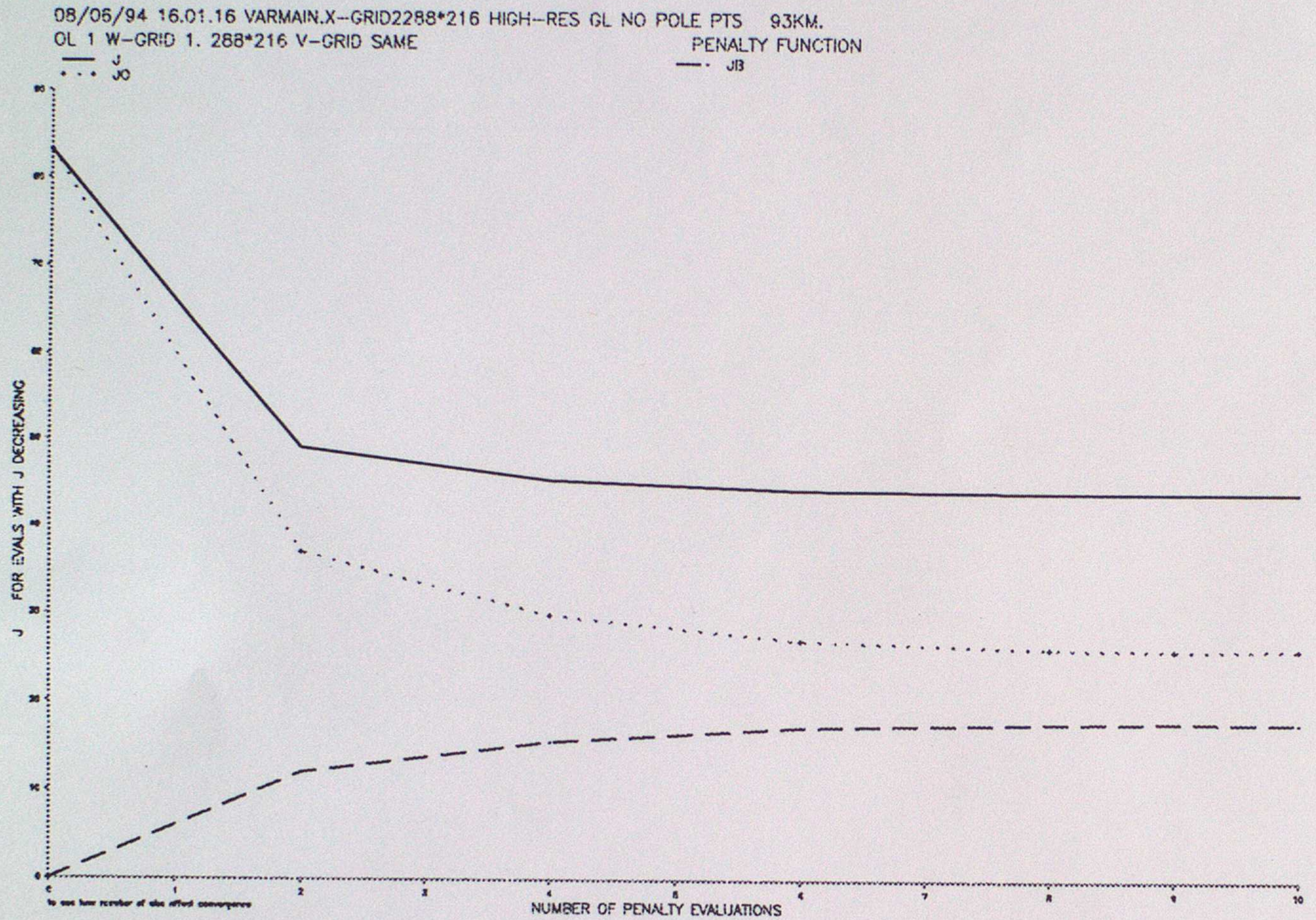


Figure 2 penalty function against inner iterations

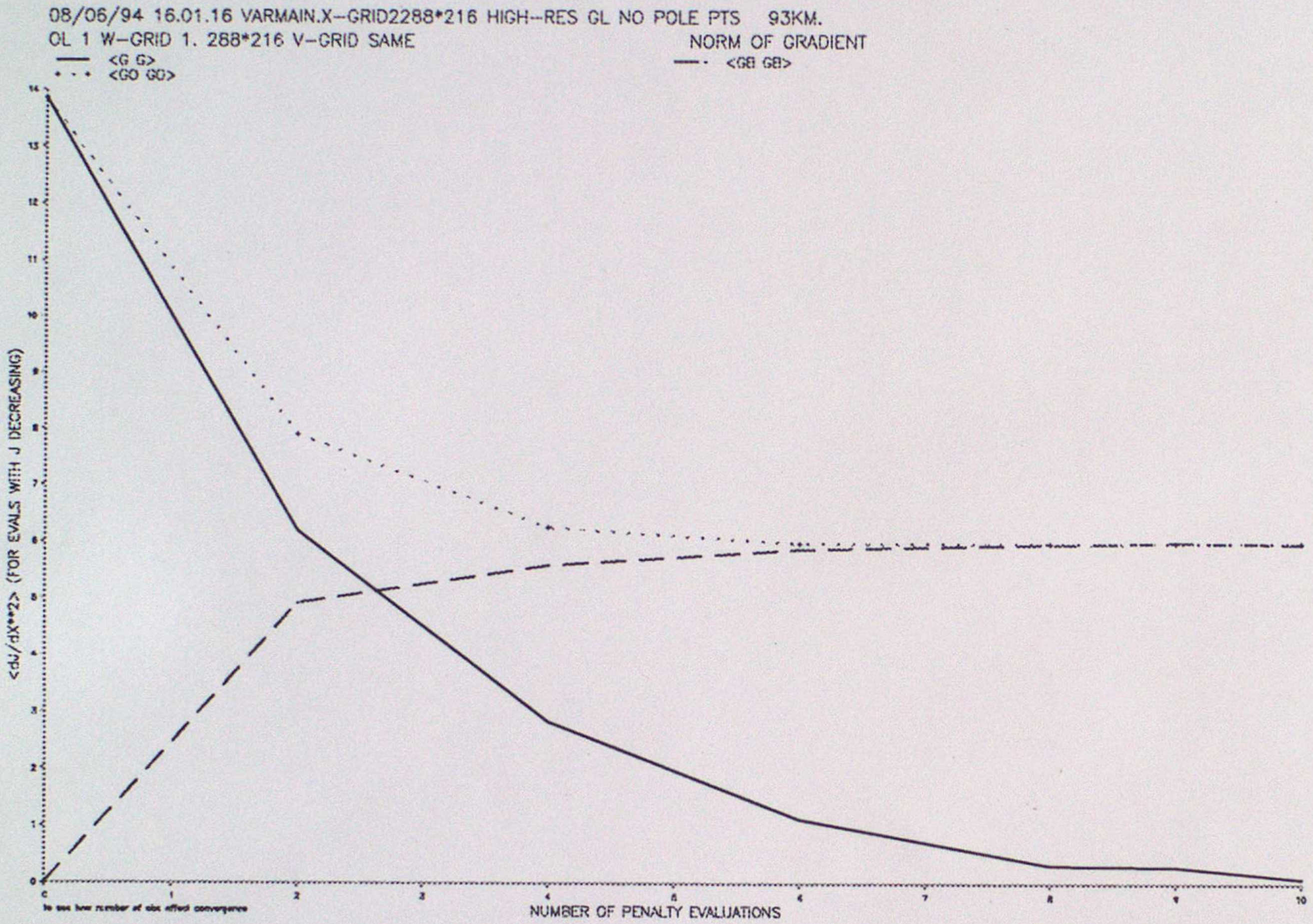
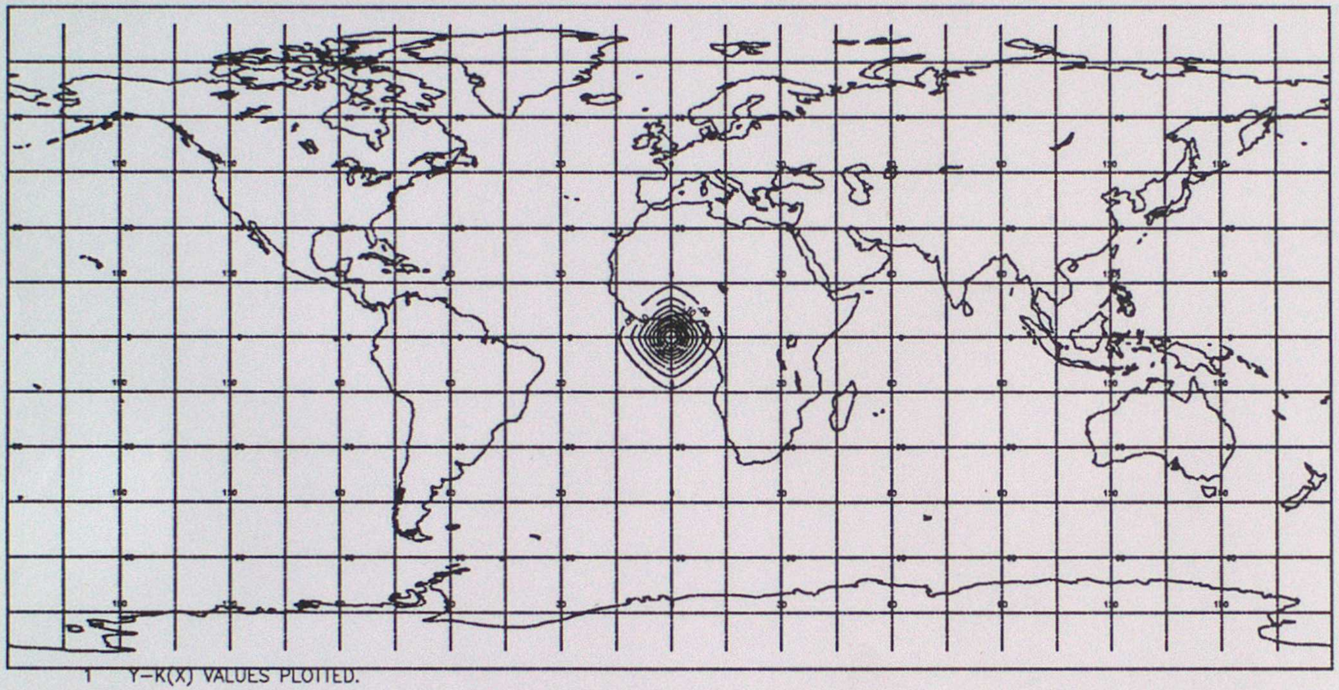


Figure 3 gradient of penalty function against inner iterations

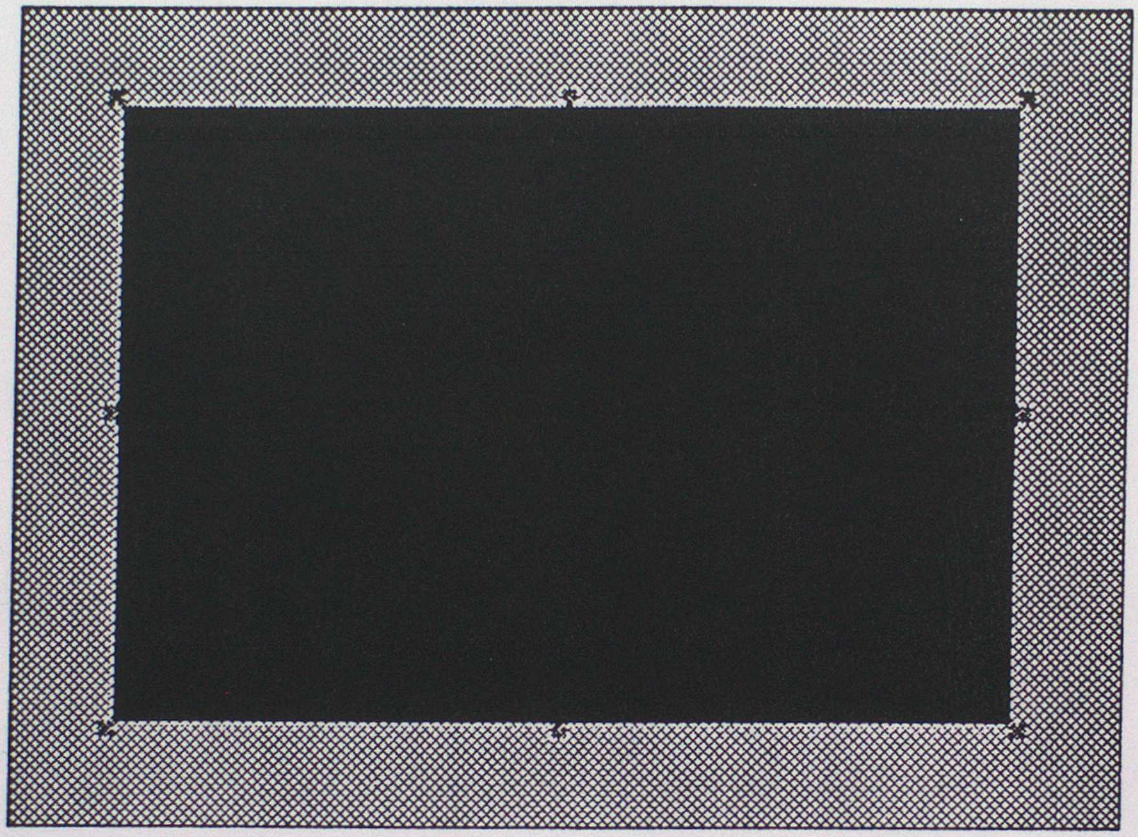
```
20/07/94 16.40.16 VARMAIN.X-GRID1288*217 HIGH-RES GL PTS AT POLE   93KM.
OL 1 W-GRID 1. 288*217 V-GRID SAME
OBJFUN    0  1     .12498     .12214     .49997     .49997     .00000
                   XFINAL  MN=    .002 RMS=    .022 MAX=    .506 MIN=    .000
```



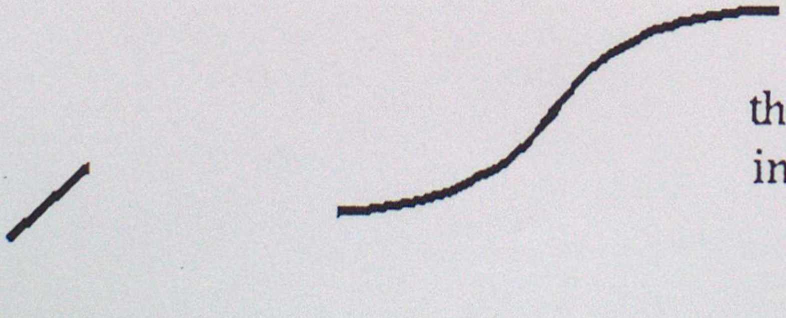
```
    1    Y-K(X) VALUES PLOTTED.
```

Figure 4 analysis of one observation at equator

Diagram showing reduced grid used in tangent linear to control variable transform



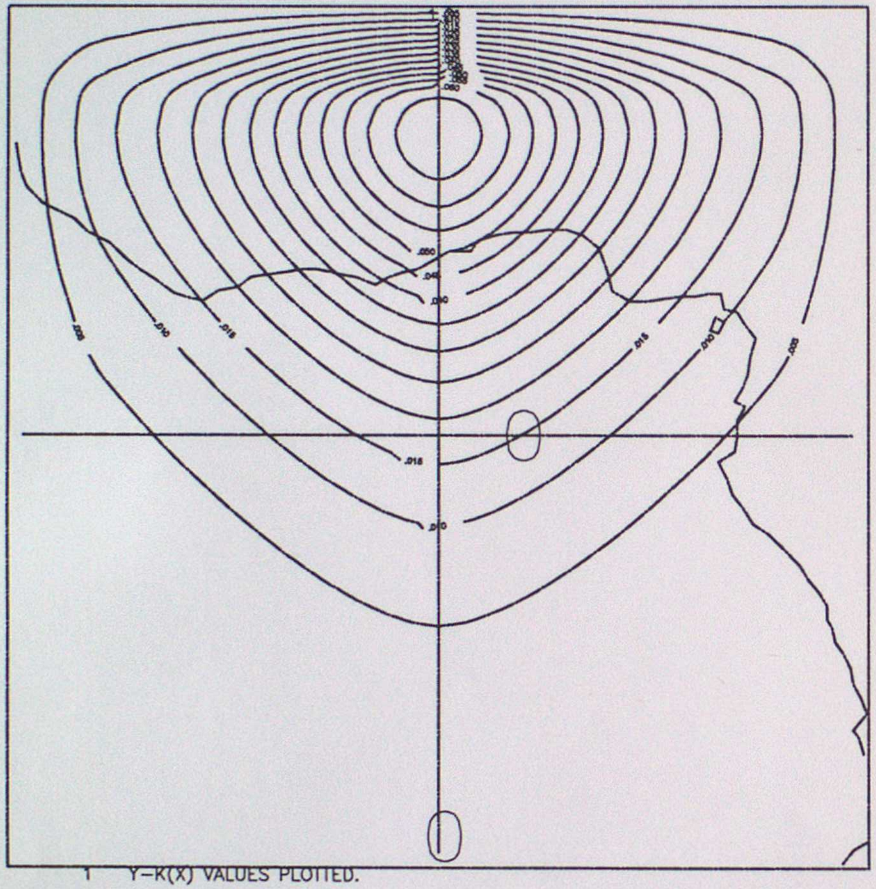
▨ these points are dropped ■ these points are kept



the two schemes for filling in the boundary

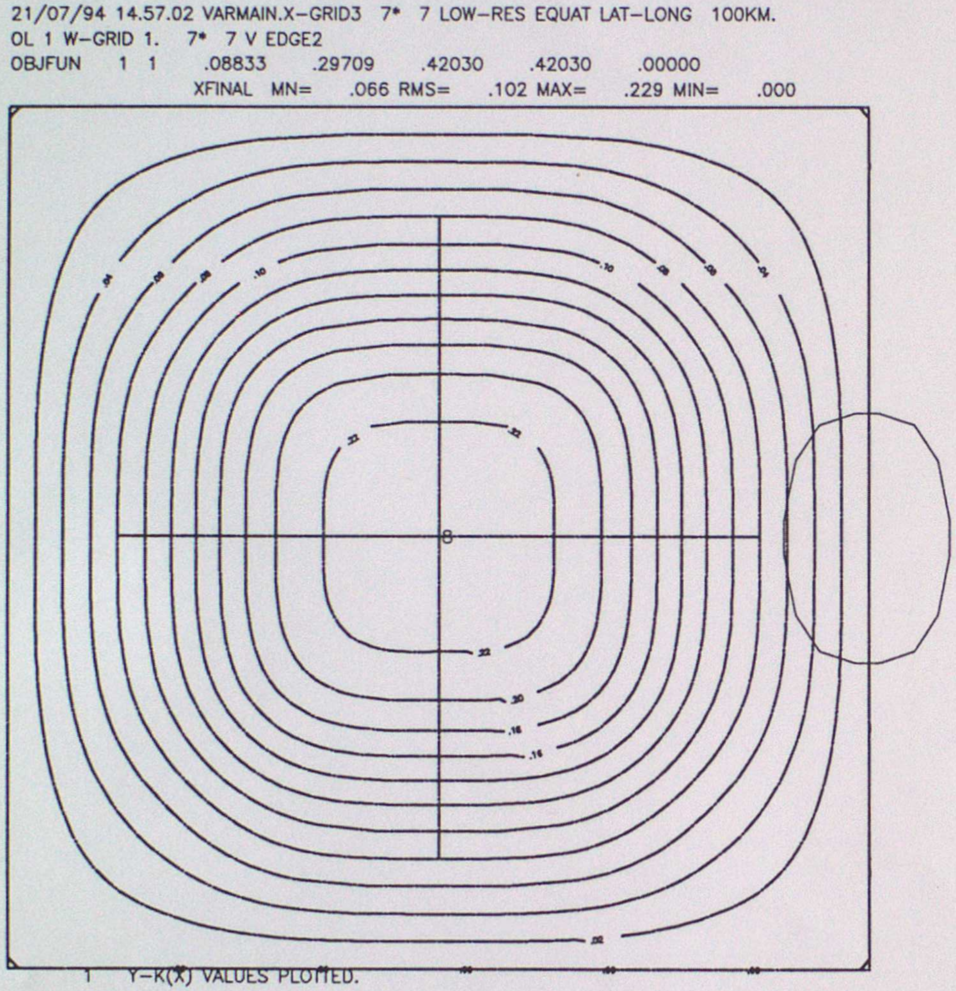
Figure 5

```
20/07/94 16.54.13 VARMAIN.X-GRID3 31* 31 MEDIUM-RES EQUAT LAT-LONG  100KM.
OL 1 W-GRID 1.  31* 31 V EDGE3
OBJFUN    1  1     .00314     .49370     .07923     .07923     .00000
                        XFINAL  MN=    .011 RMS=    .019 MAX=    .069 MIN=    .000
```



1   Y-K(X) VALUES PLOTTED.

Figure 6 analysis of one observation at LAM boundary



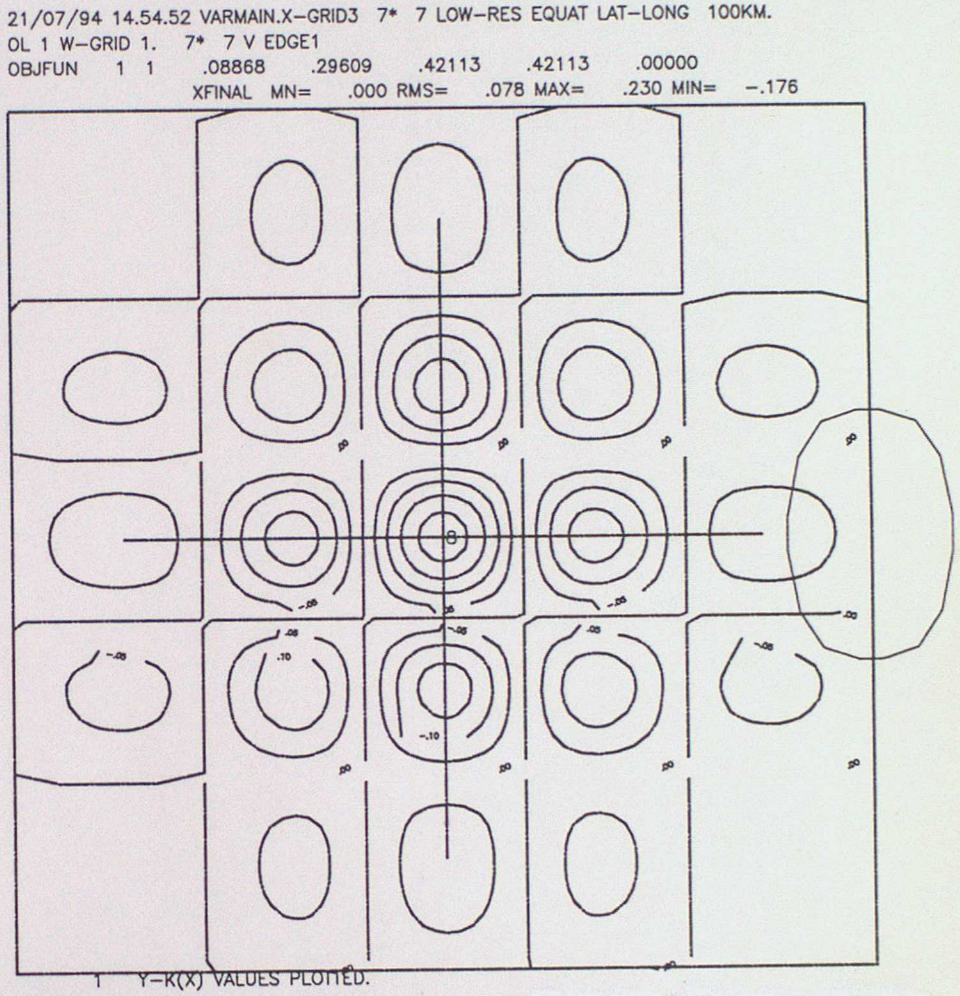
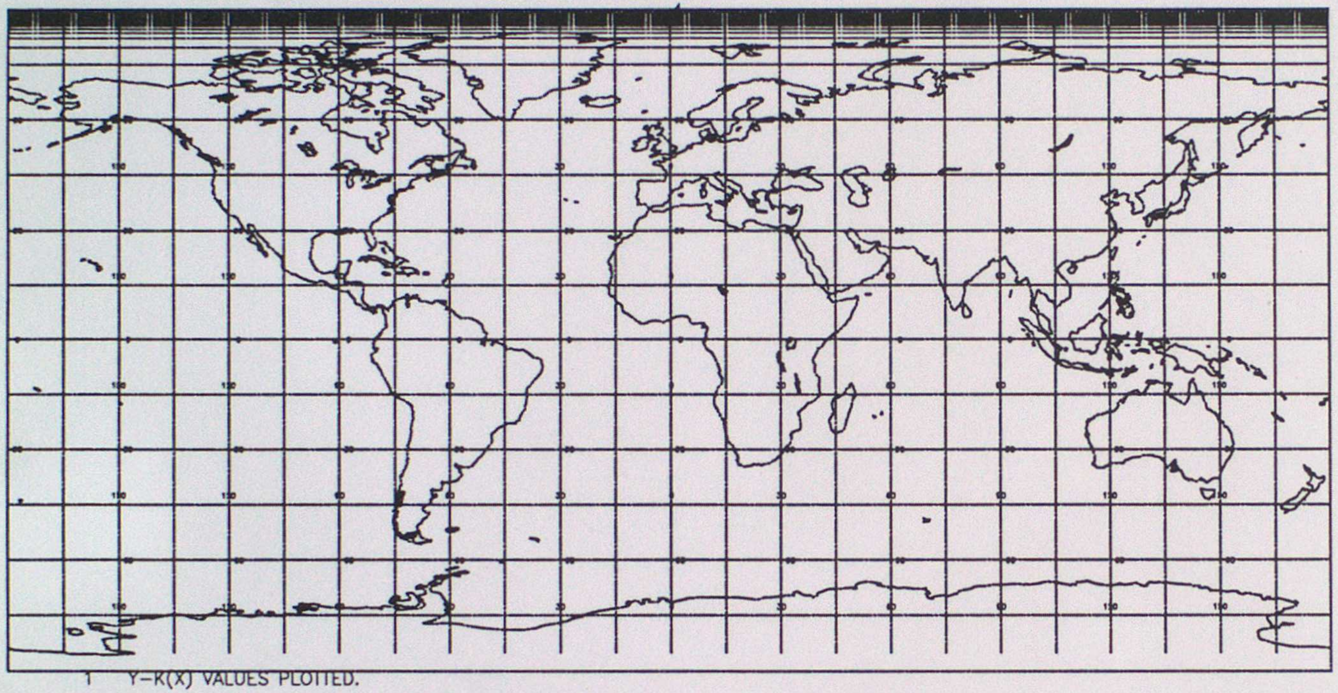


Figure 7 analysis of one observation using recursive filter

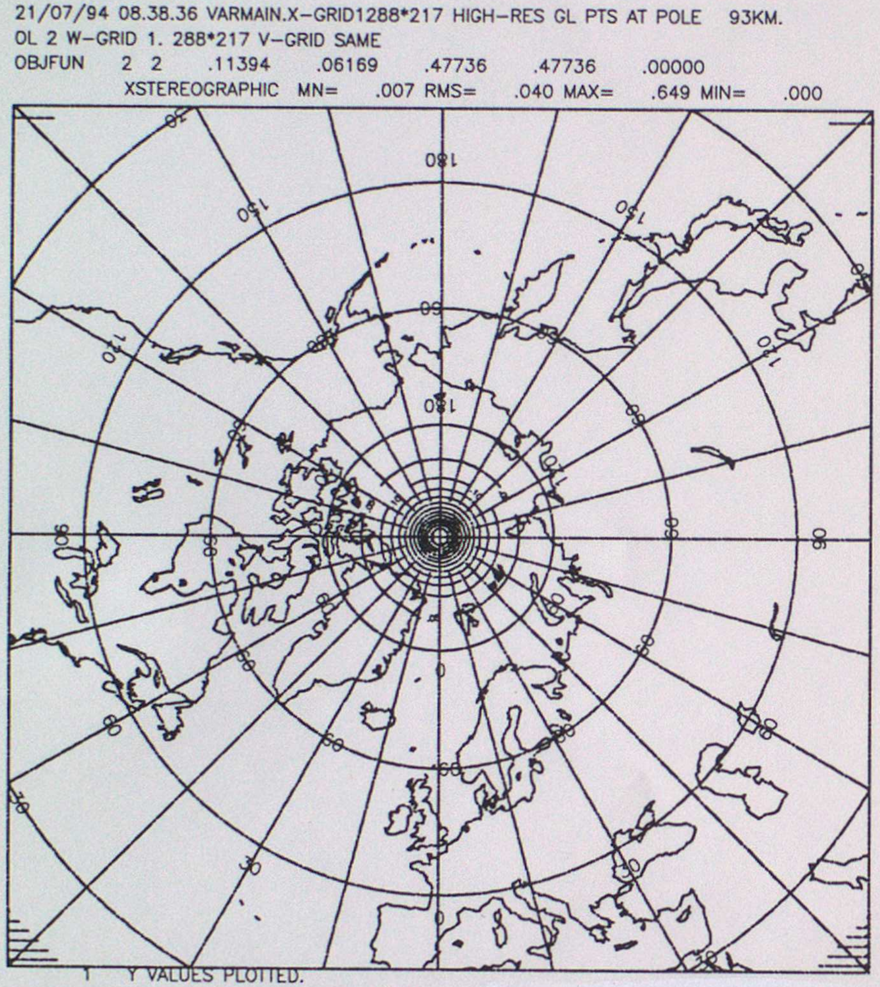
Figure 8 analysis of one observation using spectral filter

```
21/07/94 08.38.36 VARMAIN.X-GRID1288*217 HIGH-RES GL PTS AT POLE   93KM.
OL 2 W-GRID 1. 288*217 V-GRID SAME
OBJFUN    2  2     .11394     .06169     .47736     .47736     .00000
                       XFINAL  MN=     .002 RMS=     .019 MAX=     .649 MIN=     .000
```



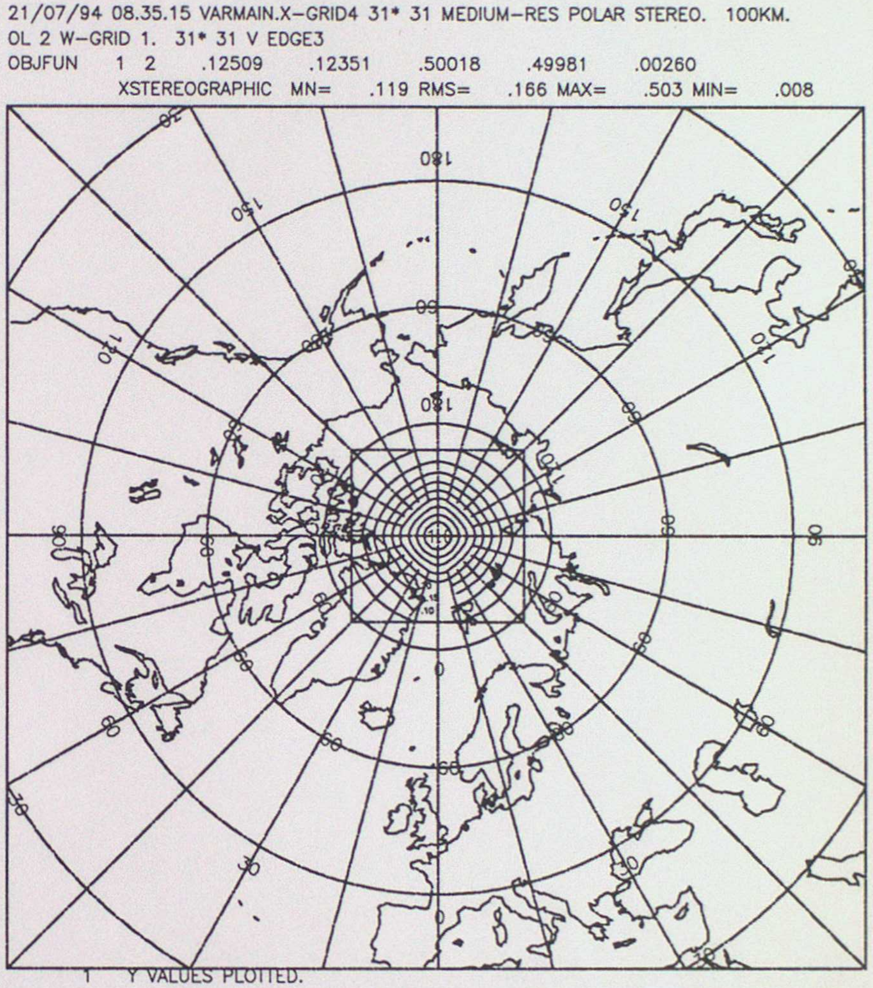
```
     1    Y-K(X) VALUES PLOTTED.
```

Figure 9 analysis of one observation at pole on global grid



```
21/07/94 08.38.36 VARMAIN.X-GRID1288*217 HIGH-RES GL PTS AT POLE   93KM.
OL 2 W-GRID 1. 288*217 V-GRID SAME
OBJFUN   2  2    .11394    .06169    .47736    .47736    .00000
           XSTEREOGRAPHIC  MN=    .007 RMS=    .040 MAX=    .649 MIN=    .000
```

Figure 10 analysis of one observation at pole interpolated to Polar Stereographic grid



```
21/07/94 08.35.15 VARMAIN.X-GRID4 31* 31 MEDIUM-RES POLAR STEREO.  100KM.
OL 2 W-GRID 1.  31* 31 V EDGE3
OBJFUN    1  2    .12509    .12351    .50018    .49981    .00260
            XSTEREOGRAPHIC  MN=    .119 RMS=    .166 MAX=    .503 MIN=    .008
```

Figure 11 analysis of one observation at pole on Polar Stereographic grid

```
20/07/94 17.14.35 VARMAIN.X-GRID2 96* 72 MEDIUM-RES GL NO POLE PTS  278KM.
OL 1 W-GRID 1.  96* 72 V-GRID SAME
OBJFUN    1  1     .12285     .15993     .49568     .49568     .00000
          XSTEREOGRAPHIC  MN=    .012 RMS=    .046 MAX=    .414 MIN=    .000
```

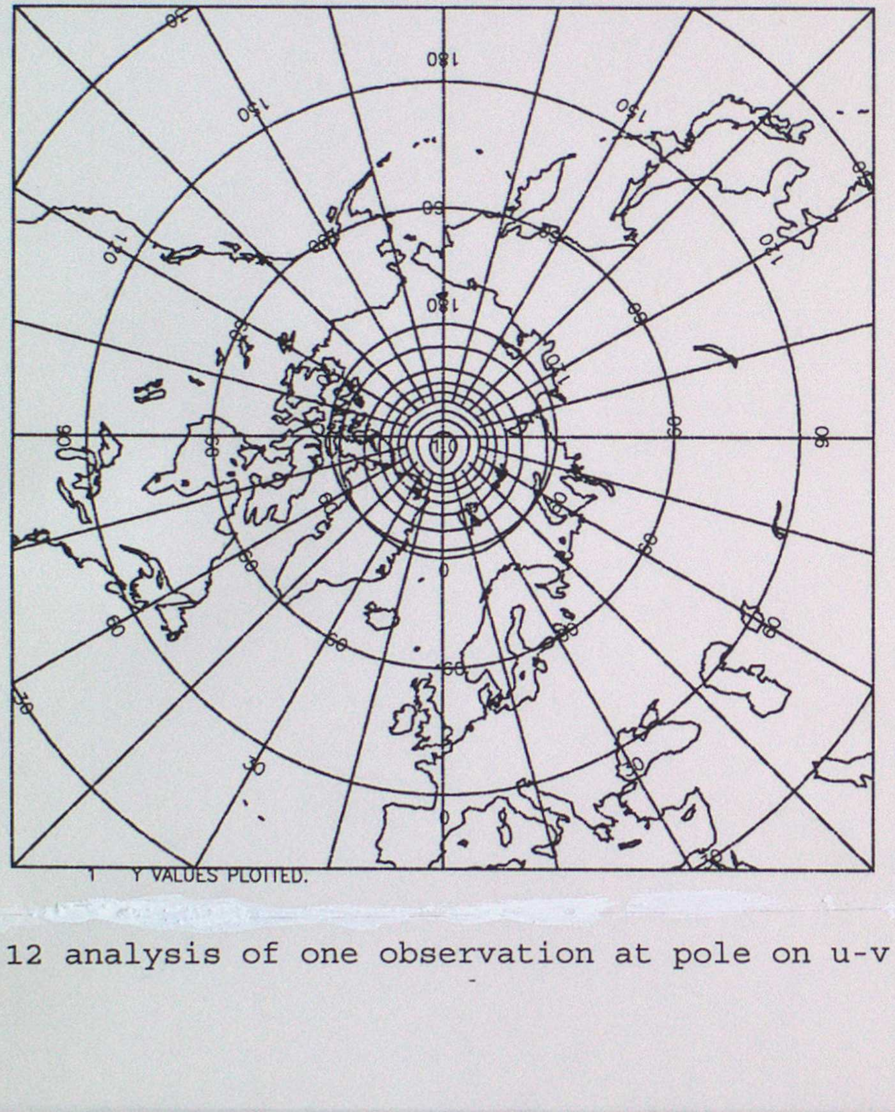


Figure 12 analysis of one observation at pole on u-v grid

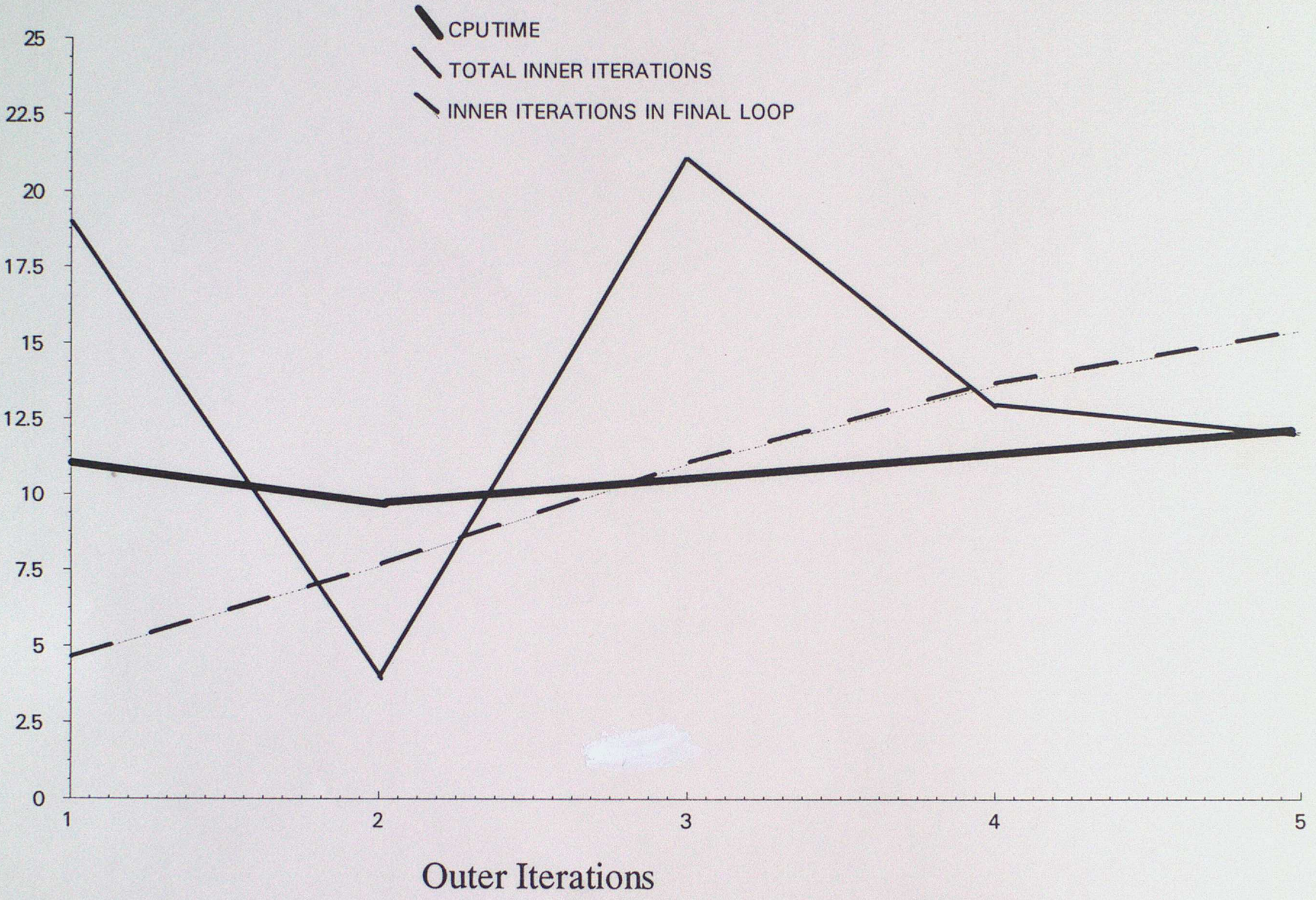


Figure 13 variation of; CPUTIME,total inner iteration, inner iterations in final loop, with outer iterations

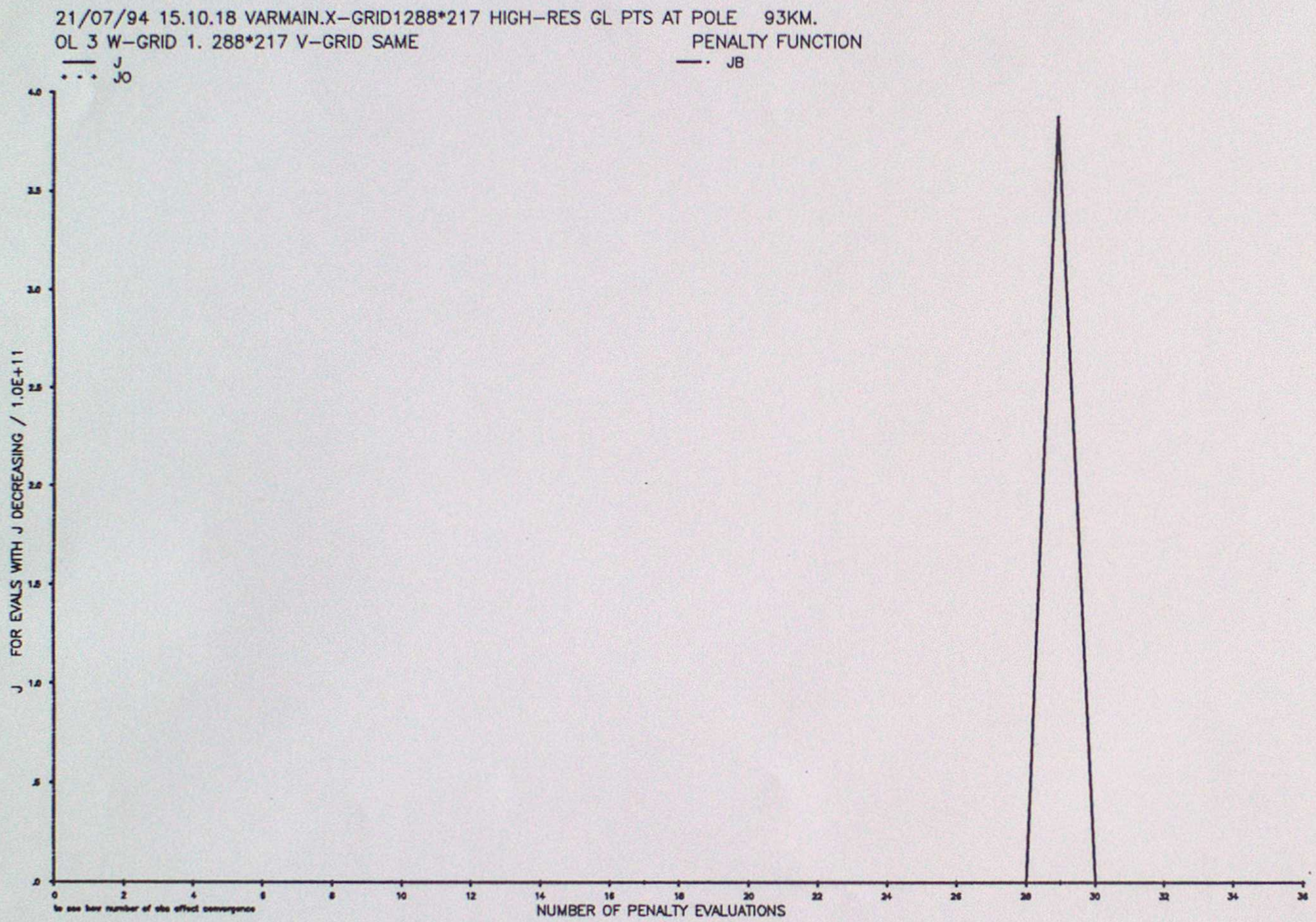


Figure 14 the spike in the penalty function that occurs for the final outer iteration

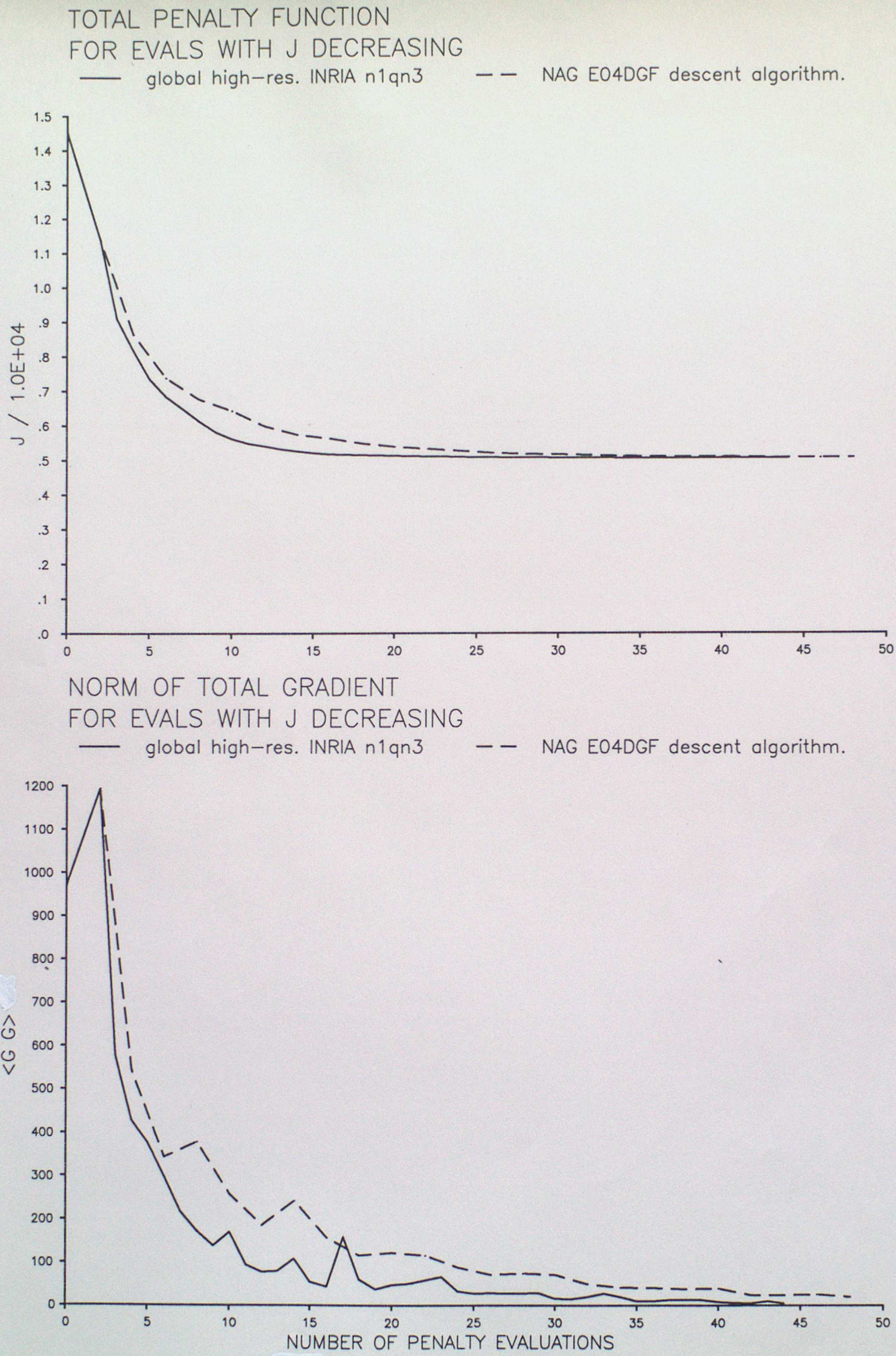


Figure 15 Penalty function, and norm of gradient, against number of penalty evaluations, for the NAG E04DGF and INRIA N1QN3 descent algorithms, for a 288*217 global grid with 10000 observations.