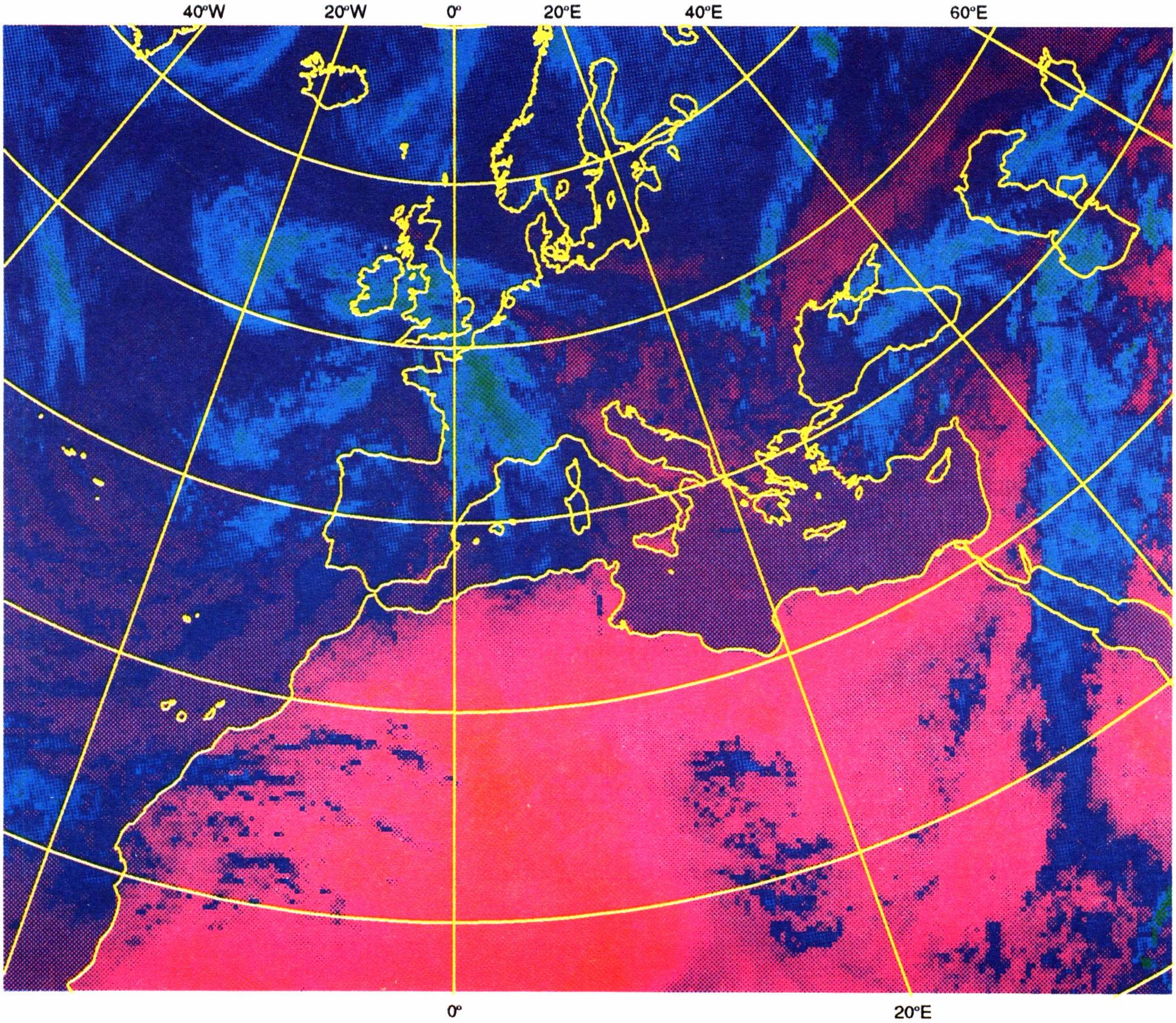
ECMWF Newsletter

Number 62 - June 1993



Shinfield Park, Reading, Berkshire RG2 9AX, England. Telephone: U.K. (0734) 499000,
International (+44 734) 499000, Telex: 847908 ECMWF G, Fax: (0734) 869450

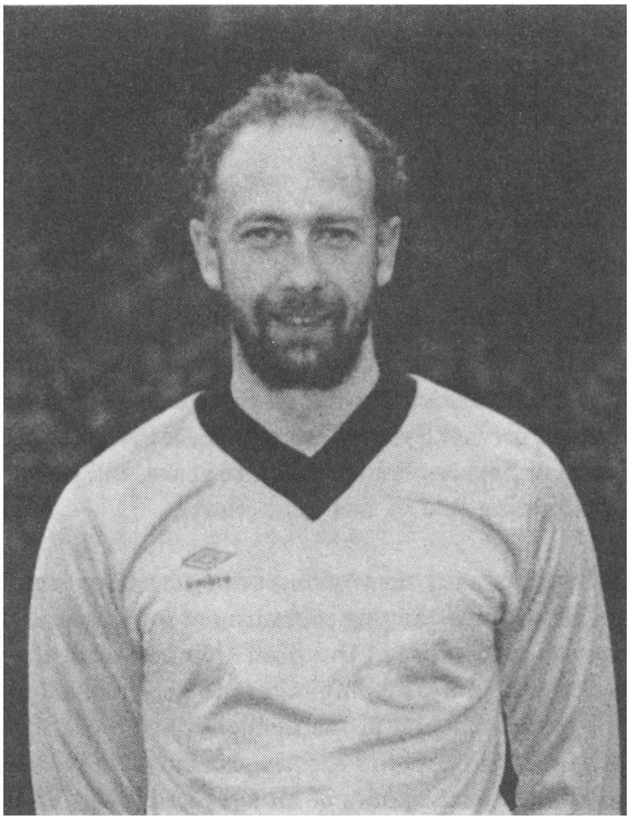
European Centre for Medium-Range Weather Forecasts
Europäisches Zentrum für mittelfristige Wettervorhersage
Centre européen pour les prévisions météorologiques à moyen terme

IN THIS ISSUE

COVER: Example of satellite image plotting in MAGICS. See article page 15.

This Newsletter is edited and produced by User Support.

The next issue will appear in September 1993.



Bill Heckley

It is with great sadness that we record the death of Bill Heckley from a brain haemorrhage sustained in the early hours of 17 June 1993 at the age of 36.

Although Bill was born in Kent in 1956, he considered himself a 'Northener', having spent his formative years and schooling in Lancashire. He studied physics at Imperial College, London and was awarded his BSc in 1977. From there he moved to the Meteorology Department at Reading University and, under the direction of Brian Hoskins, studied the meteorology of mid-latitude weather systems, obtaining his PhD in 1980.

He joined ECMWF from Reading University in November 1980 and was appointed to the Diagnostics Section where he was given the task of investigating the difficulties of weather forecasting in the Tropics. Bill tackled the problems with his customary vigour and produced a series of theoretical and diagnostic papers which established his name in the field of forecasting for the tropics and led to rapid promotion. The importance of this work for tropical countries was recognized by WMO, and Bill was invited to give lecture courses on the Centre's work in

Bangladesh in 1985, Trieste in 1986, Addis Ababa in 1988 and India in 1990. These courses were attended by meteorologists from many countries and were very successful and well received.

In 1986, after spending an enjoyable 3 months in the USA at the invitation of Florida State University, Bill moved to the Data Assimilation Section in the Centre and became involved in the search for ways to make use of satellite data in analyses for tropical regions. In the late 80s, Bill began work on the Integrated Forecast System - a major task involving collaboration between the Centre and the French Weather Service. For the last five years, Bill played an important scientific role and an important coordinating role in this joint effort.

Bill was known to his colleagues as a man of remarkable energy and enthusiasm. This was evident to all in the Centre in his wide-ranging commitment to the social side of the Centre as well as his scientific work. He was a member of the Staff Committee in the early 80s and again in 1986/7 and was noted for his zest in persuading others to get involved in sporting activities. He was one of the ECMWF's successful squad in the London Marathon in 1983 and inaugurated the 'Fun Run' in 1985 which has featured in the sporting/social calendar every Spring and Autumn since, with an amazing variety of participants of all ages and levels of fitness. He was active in all aspects of running the squash club: finding it a new home, encouraging new members, and organising tournaments to involve as many people as possible and widen its appeal. This was Bill's approach to life; to actively enjoy his pursuits and to bring others to enjoy them as well.

Outside the Centre too, he had interests which he pursued with equal energy and which he shared with others: he was a qualified Scuba diving instructor in the Bracknell Scuba Club and had trained to assistant instructor level in kayaking.

Above all, however, Bill was a dedicated husband and father. He is survived by his wife Jean and two young children, Gawain aged 10 and Elaine aged 9. We extend to them our sincerest sympathy.

- ECMWF Director and Staff

* * * * * * * * * * * *

The latest features of ECMWF's MAGICS software are described in this issue, with an illustration of their capabilities given on the cover, and in the text. In addition, for computer users, there is an article on the RAID disks newly acquired for the C90 and we continue our series of reprints, from the NCAR SCD Computing News, of articles on Fortran 90 by Jeanne Adams.

This edition also contains the announcement of the annual ECMWF Seminar, this year on "Developments in the Use of Satellite Data in Numerical Weather Prediction", and of the biennial Workshop, on Meteorological Operational Systems, which will review the present state and consider trends for the future.

Meteorologists will find a description of the main features of the Centre's variational analysis scheme, and a presentation of some of the first results of this scheme.

With sadness we also pay tribute to Dr. W. Heckley, one of the authors of the last mentioned article and a member of the Data Assimilation Section in ECMWF's Research Department, who died suddenly very recently.

* * * * * * * * * * * * *

CHANGES TO THE OPERATIONAL FORECASTING SYSTEM

Recent changes

No changes which would have had a significant impact on the performance of the ECMWF analysis and forecast system have been introduced during the last three months.

Planned changes

An improved representation of surface and planetary boundary layer processes will be implemented.

Enhancements to the quality control of aircraft data will be introduced.

- Bernard Strauss

* * * * * * * * * * * * *

THE VARIATIONAL ANALYSIS SCHEME - MAIN FEATURES AND SOME EARLY RESULTS

Introduction

A variational analysis scheme has been developed at ECMWF. It is currently being tested and compared with the operational scheme. The variational scheme basically solves the same analysis problem as the operational "Optimum Interpolation" (OI) scheme, but the variational formulation is more general than OI - it can handle non-linear operators, whereas OI is entirely linear. It has been shown theoretically that OI and the variational scheme are equivalent in the linear case. We can therefore expect very similar results for conventional data (observations of height, thickness, wind, temperature and relative humidity) with the two schemes.

For TOVS and scatterometer data, however, it will be possible to use the observed quantities directly instead of retrieved quantities. Another advantage of the variational scheme is that the level of gravity wave activity can be controlled within the analysis, so that a separate initialisation step will not be needed. The scheme therefore combines retrieval, analysis and initialisation in one step, so that a more nearly optimal combination of the information in the various observation types and in the background (a six-hour forecast) can be achieved.

The new analysis scheme is being run in experimental mode and compared with the operational scheme. Data assimilation experiments have so far been run at spectral resolution T63, for periods of up to fourteen days. Ten day forecasts (at T63) are being run from the two sets of analyses and the verification scores compared - the quality of the forecasts is the ultimate indicator of the performance of the data assimilation system. Some early results are presented here after a summary of the main characteristics of the variational scheme focusing on the observation operators. More details on the variational analysis can be found in papers by Heckley et al., Vasiljevic et al., Andersson et al., Thépaut et al., Rabier et al. and Courtier et al. in the Proceedings of the ECMWF Workshop on "Variational Data Assimilation with Emphasis on Three-Dimensional Aspects", 9-12 November 1992 (in press).

Main characteristics

The main advantage with the variational scheme is that it extends to the non-linear case. For instance TOVS radiance measurements can be used directly rather than using "retrieved" profiles of temperature and humidity. For data from the scatterometer instrument of the ERS-1 satellite, there is a choice of using either observed radar backscatter directly, or retrieved 10-metre winds. Near-surface observations, for instance SYNOP 2m-temperature and humidity and 10m-wind, can be analysed through the model's boundary layer parametrisation rather than through simplified (linear) relations. It is expected that this will lead to a better use of the data.

The operational OI scheme sub-divides the globe into so-called "analysis boxes" with dimensions of around 2000 km. OI solves the linear equations exactly within each box, but since partly varying sets of data are used for the analysis of each box some discontinuity between neighbouring boxes is unavoidable. The variational scheme on the other hand is applied globally. All observations are used simultaneously, so data selection is not an issue. This will produce less small-scale 'noise' in the analyses.

The variational formulation solves the non-linear analysis problem globally but not *exactly*. An approximate solution is found through a number of iterations of a minimisation algorithm. The scheme minimises a "cost-function" which measures the global misfit of the model to the observations and to the background. For each iteration the minimisation requires the calculation of the cost-function and its gradient with respect to the model variables. The cost-function is just one number, but the gradient has the same dimension as the model - in the order of 10^6. The most efficient way to calculate the gradient is through the application of the "adjoint" technique. The cost of calculating the gradient with the adjoint technique is typically two to four times the cost of calculating the cost-function.

The variational scheme can be applied in three or four dimensions. In three dimensions (3D-Var), data from a period of six hours, centred at the analysis time, are gathered together to form an analysis (as in current operations). In four dimensions (4D-Var), the distribution of the observations in time can be properly taken into account. The observation operator then includes an integration of the forecast model to the time of the observations. The computation of the gradient requires the integration of the adjoint of the model backwards in time. The four-dimensional scheme has so far been run with the adiabatic version of the forecast model, over periods of 24 hours. The dynamics of the model introduce a powerful additional constraint to the analysis: the solution is an analysis which produces a forecast which is close to all observations within the 24-hour assimilation period. The additional cost of the repeated integration of the model and its adjoint is however large, but it is already feasible to carry out experimental four-dimensional assimilations at spectral resolution T63. See the ECMWF Workshop proceedings where some 4D-Var results were presented.

General features of the variational scheme within "IFS"

The development of the IFS/Arpège "Integrated Forecasting System" began at the end of 1987, the work being carried out in collaboration with Météo-France, Toulouse. IFS was designed from the beginning to provide all the modules needed to perform three-dimensional and four-dimensional variational data assimilation, i.e.:

- A forecast model and its adjoint (for four-dimensional data assimilation);
- The "observation operators" (and their adjoints) needed to compare the model with observations and to compute an observation cost function (J_o: metric of model minus observation distance) and its gradient;
- The "background operators" (and their adjoints) needed to compare the model with a background (usually a 6-hour forecast) and to compute a background cost function (J_b: metric of model minus background distance) and its gradient;
- Mass/wind balance operators, i.e. normal mode initialisation (NMI) and also an optional gravity wave cost function (J_c) which measures the energy in the gravity waves of the analysis and can be used to control the amount of gravity waves in the analysis;
- A minimisation algorithm. Standard packages developed by INRIA (Institut National de Recherche en Informatique et en Automatique), Le Chesney, France, are used.

The code also contains a tool to test that the coded adjoint is the correct adjoint.

Background constraint

OI produces multi-variate analysis increments through imposition of geostrophic coupling in the forecast error statistics. This is only effective away from the equator. 3D-Var tackles the problem through allowing different constraints to be applied independently to Rossby, gravity and "uni-variate" components. By penalizing the gravity part sufficiently one may ensure that the increments are sufficiently balanced - without recourse to further, explicit, mass/wind balance operators, referred to above. The 3D-Var increments are truly multi-variate at all latitudes, even at the equator.

Observation types

By conventional observations we mean ground based observations whose observation errors are assumed not to be correlated in the horizontal. In addition, the space based satellite wind measurements (SATOB) are included in this group. The following is a list of observations currently used in the variational analysis:

"Conventional data"

SYNOP: Surface pressure (p_s), 10m wind components (u_{10m}, v_{10m}), 2m temperature (T_{2m}) and 2m relative humidity (RH_{2m})

AIREP: Upper air wind components (u,v) and temperature (T)

SATOB: Upper air wind components (u,v)

DRIBU: Surface pressure (p_s), 10m wind components (u_{10m},v_{10m}) and 2m temperature (T_{2m})

TEMP: 10m wind components (u_{10m}, v_{10m}), 2m temperature (T_{2m}), 2m relative humidity (RH_{2m}), geopotential heights (Z), upper air wind components (u, v), upper air relative humidity (RH) and upper air temperature (T)

PILOT: Upper air wind components (u, v)

"Non-conventional data"

TOVS: Cloud-cleared radiances (R), or retrieved layer-mean temperature (T_v) and humidity (PWC).

SCAT: Radar back scatter (σ^0), or retrieved 10m winds (u_{10m}, v_{10m}).

Most of the observed quantities listed above are used also by the current ECMWF OI system with the exception of some near surface observations, AIREP temperature, TOVS radiances and SCAT radar back scatter. Specifically in the OI, 10m wind and 2m relative humidity observations are used only over sea, whereas 2m temperature is not used at all (10m wind is also used in tropical land areas with low altitudes).

Evaluation of the cost function and its gradient

The J_o term is a quadratic form comprising the covariance matrix of the observation errors and the observation departures from the model (observation minus model). Calculation of the observation departures requires the specification of "observation operators" (H) which calculate model equivalents of the observed quantities at the observation points.

Since, with rare exceptions, no model variable is directly observed, H plays the role of a "post-processing" operator, providing the model equivalents of the observed parameters. The operator H is the product of several distinct operators. Fig. 1 shows schematically the various steps involved in computing the cost function and its gradient for TOVS radiance data. It starts with a change of variable from the control variable to the model spectral variable, followed by an integration of the spectral forecast model to the time of the observations. This forecast step is suppressed in the 3D case. Thereafter follow the inverse spectral transforms which provide the model values on the Gaussian grid at all model levels. The model grid-point data are then interpolated in the horizontal with a 12-point bi-cubic interpolator and in the vertical to the levels of the observations. The horizontal interpolation is common for all observation types. The vertical interpolation differs for different parameters. For TOVS the model data are interpolated to 40 pressure levels from which radiances are computed with a radiative transfer model, which integrates the radiative transfer equation in the vertical.

The evaluation of the cost function is straightforward (lower part of Fig. 1), given the observed departures and the matrix of observation errors.

The minimisation algorithm requires the gradient of the cost function with respect to the control variable. This computation uses the adjoint relationship, similar to the chain rule for differentiation. The gradient is obtained by the application of the adjoint of each of the operators of Fig. 1 - in the reverse order.

The gradient of J_o is added to the gradient of J_b (and J_c) and passed to the minimisation, which finds a new model state closer to the minimum of the cost-function. The whole procedure is repeated until convergence is reached, or until the maximum number of iterations has been reached. We currently stop 3D-Var after 50 iterations.

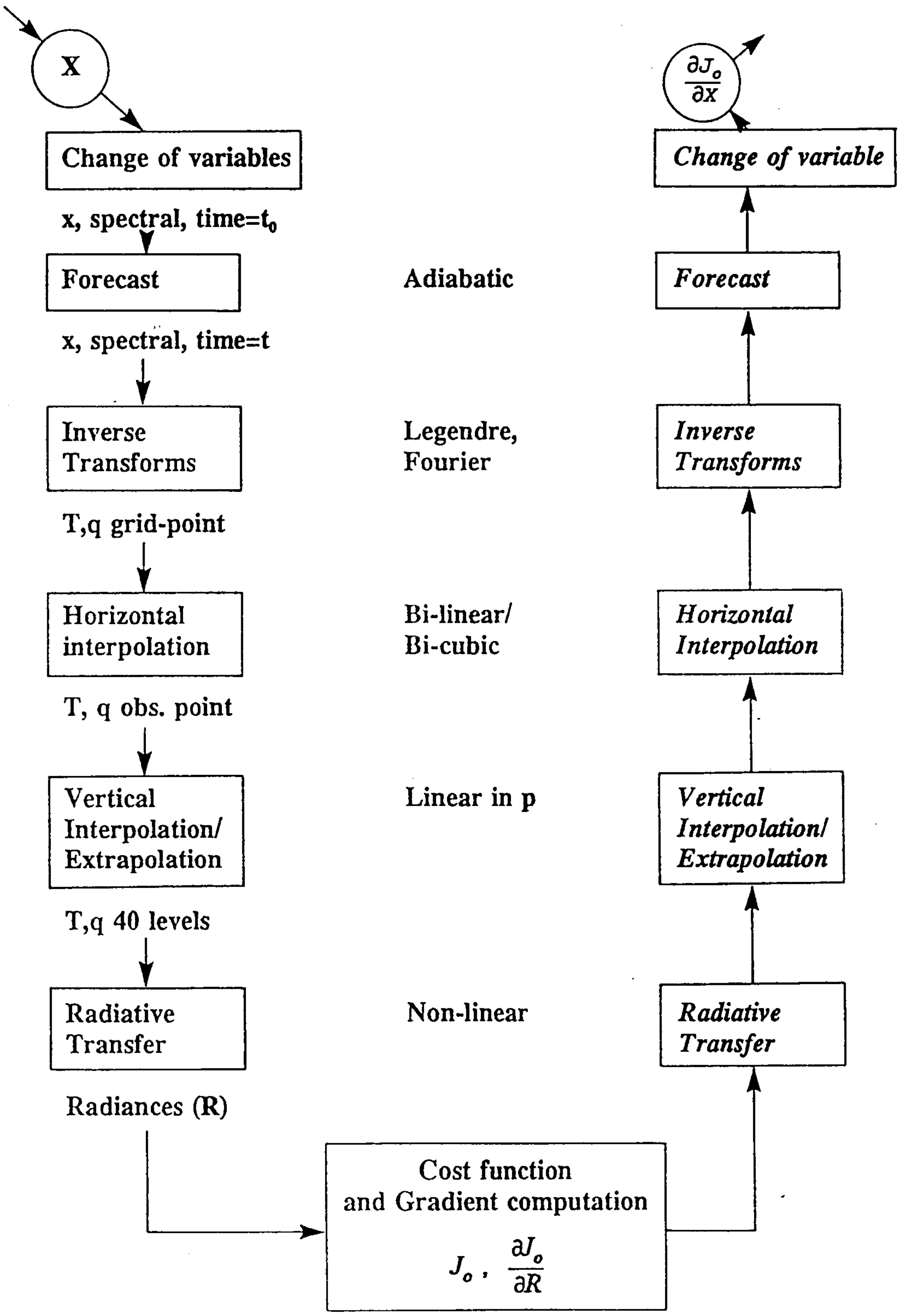
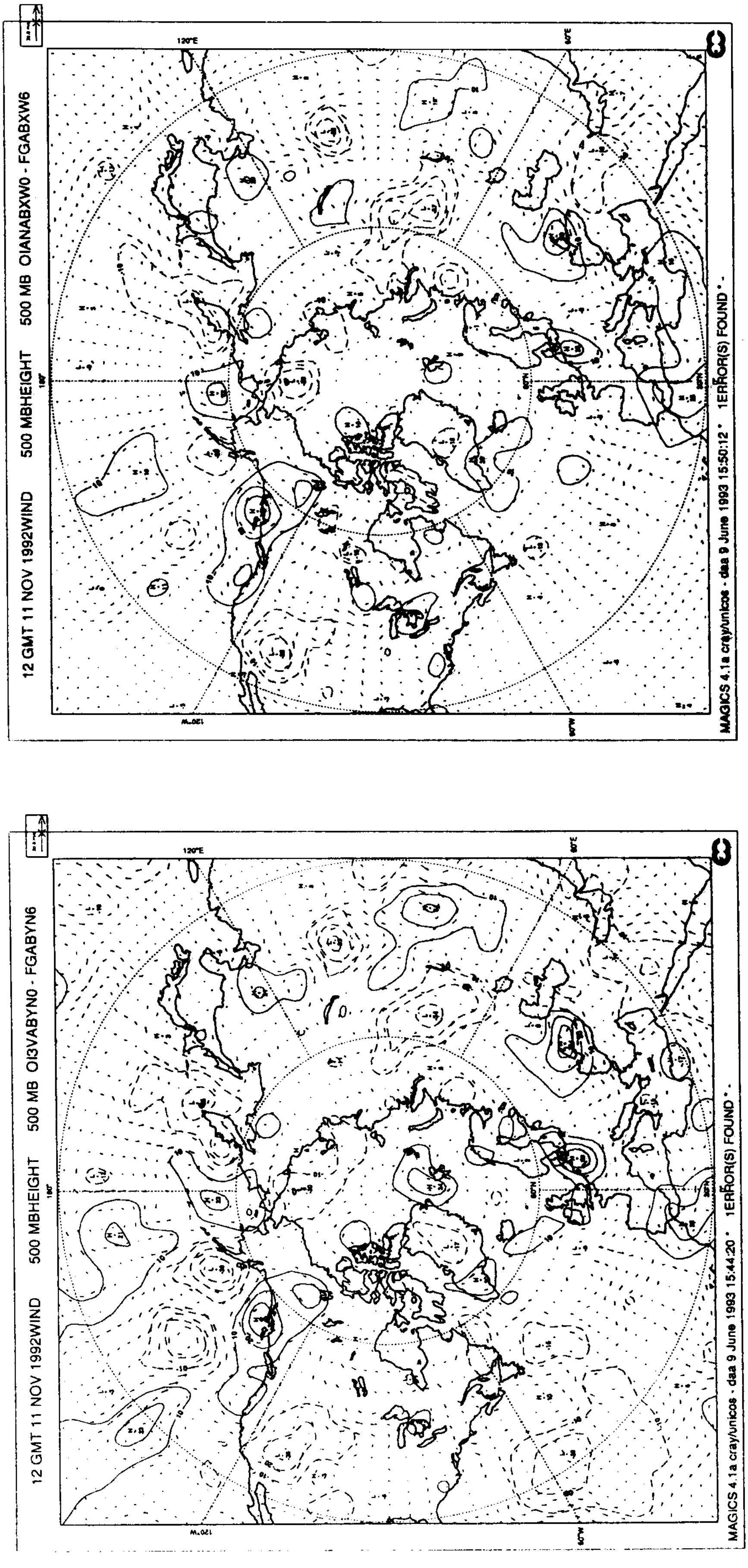


Fig. 1: Schematic of the calculation of the observation cost-function (J_o) for TOVS radiances - direct (on the left) and adjoint (on the right).



b)

a)

Fig. 2: Analysis increments (analysis minus background) of geopotential height and wind at 500 hPa, 92.11.11-12 UTC, with a contour interval of 10 metres, negative dashed. a) is 3D-Var and b) is OI.

Results

Fig. 2 shows an example of a 3D-Var T63 analysis (2(a)) using all data as described above, except scatterometer data. The figure can be compared with the corresponding OI analysis at the same resolution (2(b)). The 3D-Var analysis has used TOVS radiances directly, whereas the OI analysis has assimilated TOVS data in the form of retrieved profiles of temperature and humidity. The figures show analysis increments (analysis minus background) of 500 hPa height (contoured) and wind (arrows). We see that the two analyses are very similar, with the 3D-Var increments generally somewhat larger at this level.

The assimilation experiments were continued for five days (1992.11.11-12 UTC to 1992.11.16-12 UTC) and T63 10-day forecasts were run from the 12 UTC analyses of each day from the 12th to the 16th. The average forecast scores in the form of anomaly correlations (left) and root mean square error (right) are shown in Fig.3, for the Northern Hemisphere (top panel) and for the Southern Hemisphere (bottom). The figures show the scores of the OI experiment (dotted line) and of two 3D-Var experiments. The difference between the two 3D-Var experiments is in the specification of forecast error statistics, specifically the vertical correlation of forecast errors, which determines how the information of the observations is spread in the vertical. In the first experiment (BXQ, dashed line) the specification is similar to OI, i.e. well suited for the assimilation of conventional data. In the second experiment (BYN, full line) the specification is similar to what is used in the ECMWF retrieval scheme for TOVS data, i.e. well suited for the assimilation of TOVS radiance data. We can see that the performance of BXQ is virtually equivalent to the OI experiment in the Northern Hemisphere, but much worse in the Southern Hemisphere. The forecasts from BYN on the other hand are worse than OI in the Northern Hemisphere but better than OI in the Southern Hemisphere.

The explanation is that TOVS data dominate the analysis in the Southern Hemisphere and conventional data dominate in the Northern Hemisphere. The results show that the forecasts from the variational analysis are comparable in quality to those from the OI scheme provided the appropriate forecast error statistics are used. The forecast error statistics used in OI and in the BXQ 3D-Var experiment are good in the Northern Hemisphere but less appropriate for the assimilation of TOVS radiance data. The solution is a "non-separable" specification of forecast error statistics. Work in this area is continuing.

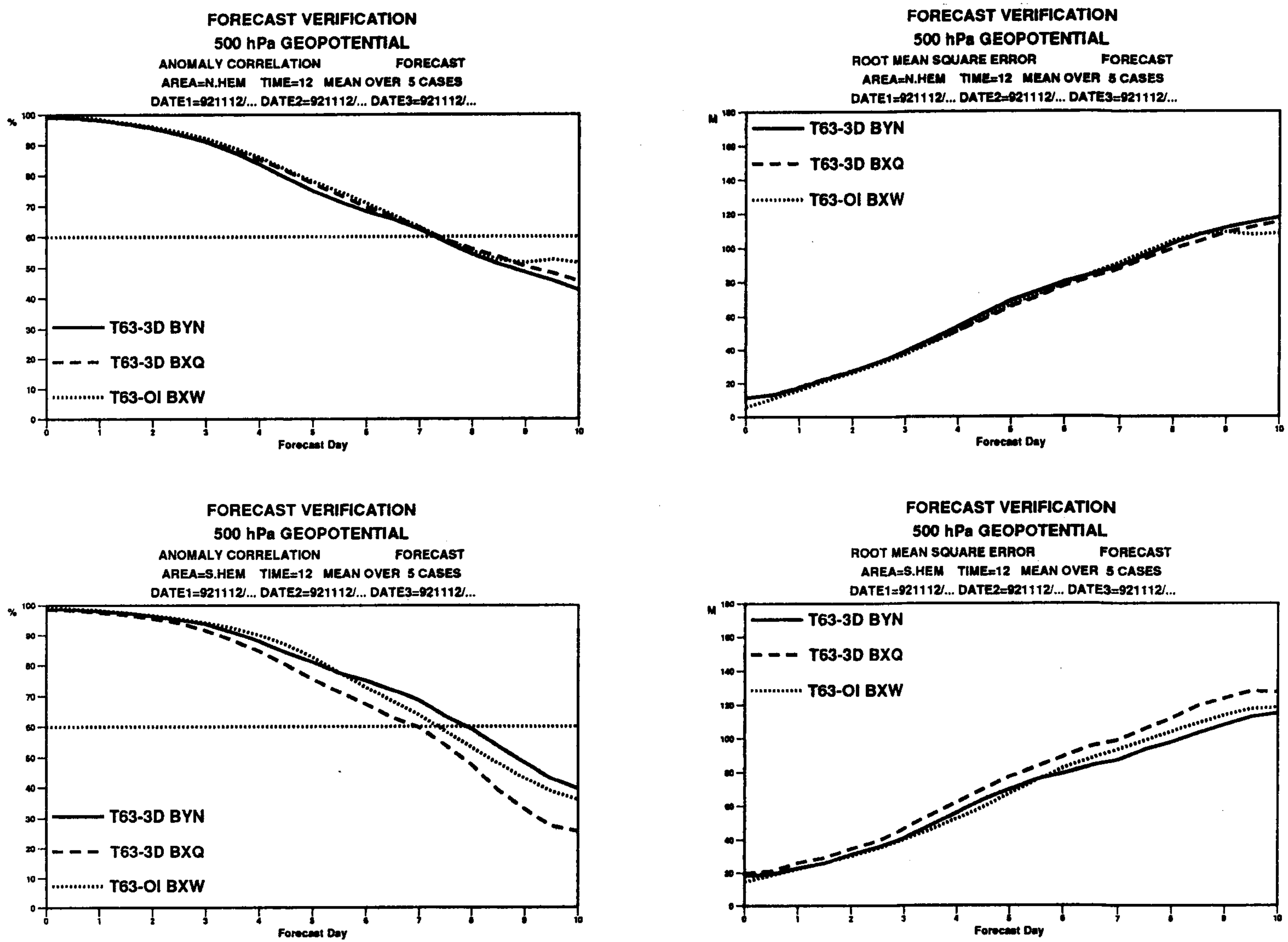


Fig. 3: **500 hPa height, average forecast verification of five 10-day forecasts at spectral resolution T63 from three different data assimilation experiments: BXW (dotted line) is OI, BXQ (dashed) is 3D-Var using forecast error statistics similar to OI and BYN (full line) is 3D-Var using forecast error similar to the ECMWF retrieval schemes for TOVS radiance data. Anomaly correlation on the left and root mean square error on the right. Northern Hemisphere, top panels, and Southern Hemisphere, bottom panels.**

<u>Summary</u>

We can conclude that the three dimensional variational data assimilation scheme produces analyses which compare well with the operational OI analyses. We have shown that 3D-Var in its current state is able to produce forecasts with a quality which is comparable to the operational scheme. Work on the new scheme is continuing. The new scheme offers much more flexibility, making it possible to introduce additional data, and in the future to develop and include additional physical constraints on the analysis.

- E. Andersson, C. Cardinali, P. Courtier, W. Heckley, F. Rabier, J-N. Thépaut, P. Undén and D. Vasiljevic

* * * * * * * * * * * * *

MAGICS - THE ECMWF GRAPHICS PACKAGE

Introduction

The Meteorological Applications Graphics Integrated Colour System (MAGICS), developed at the European Centre for Medium-Range Weather Forecasts (ECMWF), is a software system that permits the plotting of contours, satellite images, wind fields, observations, symbols, streamlines, isotachs, axes, graphs, text and legends.

Development started in 1984 with the first release for Centre users in 1985. It has since been used by several Member States. There have been many developments since the last Newsletter article on MAGICS and the purpose of this paper is to summarise the present status.

MAGICS was designed to conform to and use new meteorological and graphics standards, e.g. GRIB, BUFR,GKS. It makes use of modern contouring methods (CONICON[1]) and enables users to take advantage of colour graphics and device independence.

Data fields to be plotted may be presented in various formats, i.e. GRIB code data, BUFR code data or in matrices.

The format of the graphical output from MAGICS depends on the GKS workstations (drivers and metafiles) supported by the GKS implementation. MAGICS interfaces to a GKS level 2B implementation.

MAGICS features

MAGICS contains the following features:

- flexible user interface with a comprehensive list of simple English language parameters;
- extensive use of colour;
- Matrix, GRIB and BUFR code data input;
- selection of geographical area and direct projection of data;
- polar stereographic, cylindrical, mercator and spaceview projections;
- high quality contouring based on CONICON and lower quality linear contouring;

[1] Developed by the University of Bath, UK

- satellite image plotting in all four projections;
- shading between contour lines and also in graph and coastline plotting;
- wind fields directly projected, as arrows, WMO flags, streamlines or isotachs;
- full observation plotting;
- symbol plotting;
- axis and graph plotting;
- advanced legend plotting;
- device independence (metafiles);
- storage of program context (specification groups).

MAGICS subroutines and parameters

MAGICS consists of a small set of FORTRAN callable subroutines which are divided into five different categories, one for each of initialization, parameter setting, enquiry, action and pseudo action. Action routines are the only ones that produce plotted output and the parameters associated with each action routine should be set, if required, before the action routine is called.

Parameters in MAGICS are the attributes to be assigned to the various items that make up plotted output, such as line-style, colour and size of plot. They control all plotted output.

The parameters may be set by calling MAGICS parameter setting routines, where the use of a keyword (a MAGICS parameter) as an argument defines the action to be taken.

A typical MAGICS program consists of a series of parameter setting, action and pseudo action routines. Following is a simple example of a MAGICS program.

```
CALL POPEN                                    \* OPEN THE MAGICS PACKAGE *\
CALL PSETC('CONTOUR_LINE_COLOUR','RED')       \* PARAMETER SETTING *\
CALL PSET2R('INPUT_FIELD',FARRAY,320,160)     \* PARAMETER SETTING *\
CALL PCOAST                                   \* PLOT GLOBAL CYLINDRICAL MAP *\
CALL PCONT                                    \* PLOT CONTOURS *\
CALL PCLOSE                                   \* CLOSE MAGICS *\
```

Plot layout

To make the positioning of users' plots simple, MAGICS has introduced the concepts of subpage, page and super page.

A super page corresponds to the plotting area to be used on a plotting device. With the plot layout facilities, pages can be positioned either automatically or by the user within the super page. Similarly, one or more subpages can be plotted inside a page.

Mapping

Mapping is the placing or projecting of coastlines, grids and data onto the user's subpage. Four types of projections are catered for in MAGICS, cylindrical, polar stereographic, mercator and spaceview (satellite). It is not necessary for users to extract the required area before passing data as MAGICS can perform this function and the conversion to the various projections, if required. It is possible to shade both the land and sea areas.

Data input

Data to be plotted can be presented in a number of ways, depending on its type: contour fields, wind fields, observations, satellite images, axis and graph data etc.

A contour field can be presented as either a two-dimensional array of grid point values or in GRIB code format. GRIB code can be presented as either an external file or as a one-dimensional data array. Matrix data can be regularly spaced or on a gaussian grid.

A wind field can be presented as either a pair of two-dimensional arrays of grid point values or in GRIB code format. The pair of two-dimensional arrays can contain either U or V velocity components, or speed and direction components, of the wind. GRIB code can be presented as either an external file or as a pair of one-dimensional data arrays.

Observations must be presented as an external file. Data for observation plotting must be in WMO BUFR code.

Satellite images may be presented as a two-dimensional array of points or in GRIB code.

Axis and graph data must be presented in one dimensional arrays.

GRIB data

Facilities exist in MAGICS for automatic processing and plotting of GRIB coded data. In particular MAGICS can decode the GRIB data[2], convert from spectral to grid point, scale the fields and set up the MAGICS data input parameters. The pseudo action routine PGRIB handles all these functions. MAGICS can plot data in either GRIB edition 0 or 1, various gaussian grids and bit-map fields. MAGICS can cope with the extended GRIB cope for satellite image data and GRIB data on shifted/rotated grids can also be plotted. A special feature allows the plotting of wave direction and heights as wind arrows.

[2] By interfacing to the ECMWF library EMOSLIB

Contouring

MAGICS contouring is based on CONICON, which is designed to draw the contours of continuously differentiable fields.

For the user interface, emphasis has been placed on simplicity, making it easy for the user to define the contour levels required and the attributes to draw them with. Users have full control over plotting of labels, highs and lows as well as the thickness and colour of contour lines.

There is a faster but coarser method of contouring available in MAGICS, based on a linear interpolation. MAGICS shading facilities enable the user to shade between contour levels. Another feature of MAGICS contouring allows users to plot the grid point values on their exact location, either on their own or overlaid on contour lines.

Satellite image plotting

MAGICS provides facilities for plotting satellite images (see front cover) with optional overlaid fields, observations and coastlines. Each pixel in an image will be plotted in the tile format, i.e. each pixel is represented by a small rectangle filled with a colour (note that grey scale definitions are special cases of colour definitions). Input image data must be in the spaceview projection. If necessary, MAGICS will automatically reproject the image from spaceview to the required projection.

Image data can be passed to MAGICS in the GRIB representation or can be passed as a matrix.

The colour distribution of the plotted image is controlled by the use of a colour table. The user may specify his own colour table or can use the MAGICS default one.

Wind field plotting

Wind fields may be presented to MAGICS as U and V velocity components, as speed and direction components or as GRIB code data. They may be plotted in one of four ways:

- wind arrows, where the wind is represented by a vector whose length is proportional to the speed of the wind;

- WMO standard wind flags, where a wind flag is represented by barbs and solid pennants;

- streamlines; a streamline is a line whose tangent at any point is parallel to the instantaneous velocity at that point;

- isotachs, which are contour lines of equal wind speed.

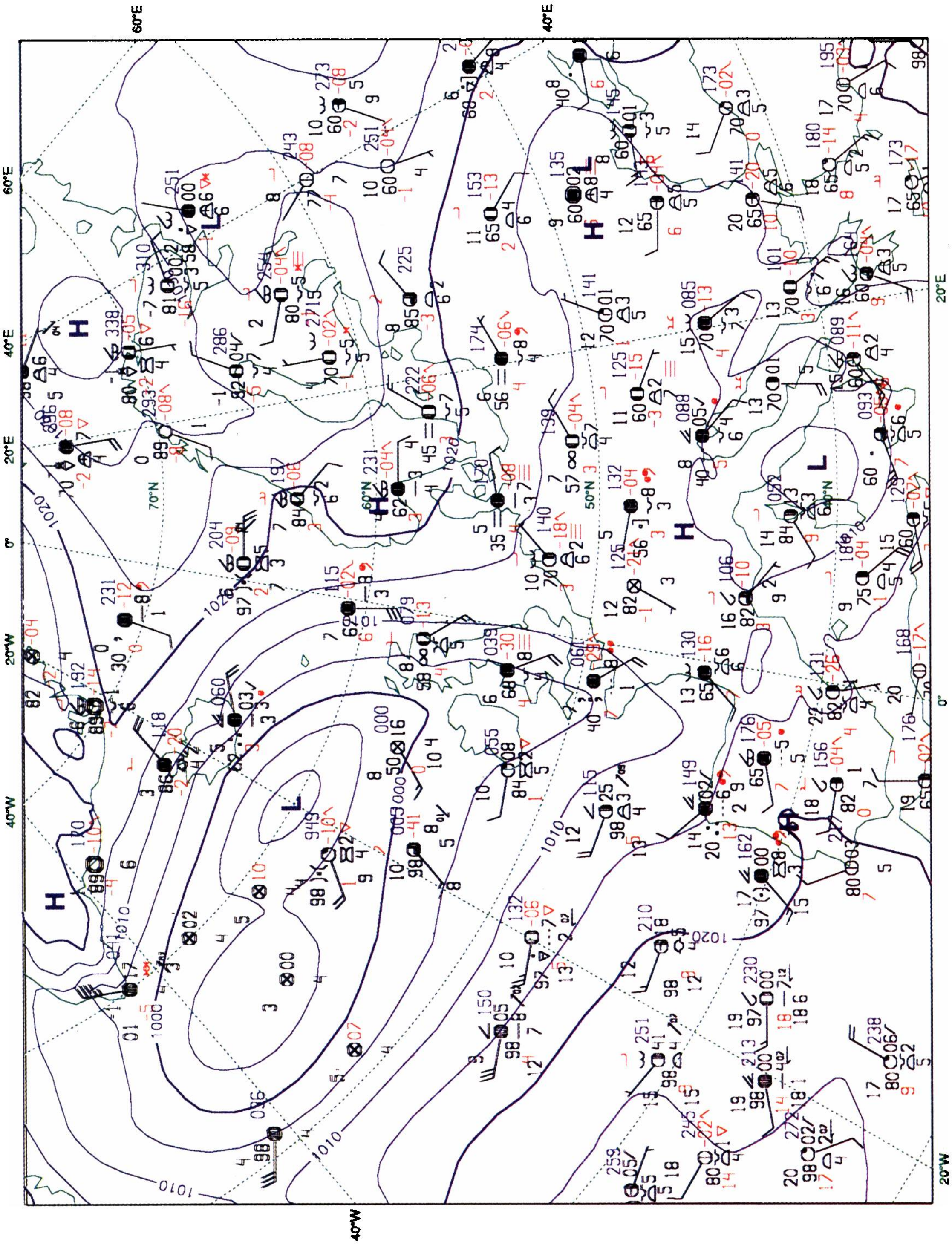


Fig. 1 Example of SYNOP observation plotting overlaid with contours of mean sea level pressure analysis (3 April 1993 12 UTC).

Wind related parameters, which are generated from differential properties of the wind field, may be plotted using normal contouring facilities, e.g. Divergence, Vorticity, Stream Functions and Velocity Potential. The use of colour enables the plotting of coloured wind arrows where the colour of the arrow can represent the relevant temperature, humidity, etc.

Observation plotting

Observation plotting (see Fig. 1) allows for the plotting of all observation types. WMO standard plotting formats are adhered to as closely as possible.

The user has full control over the type, number and size of the observations to be plotted. It is possible to plot only the positions of observations. The amount of meteorological information plotted may also be controlled by the user.

Axis plotting

MAGICS axis facilities allow users to plot vertical and horizontal axes. These facilities include axis labelling, axis title plotting and subdivision of axes with ticks. MAGICS axis parameters give the user control over all aspects of axis plotting, e.g. position, orientation, colour, thickness, etc. Axes may be regular, logarithmic or user-defined and may be subdivided with tick marks. Further subdivision between ticks, i.e. minor ticks, is also possible. Ticks may be labelled either on the tick mark or between ticks. Grid lines, may be plotted. MAGICS axis facilities can be used for plotting cross-sections or for graph plotting.

A special feature in Axis plotting is the automatic DATE/TIME axis facility. If the user specifies the start and end date, MAGICS will generate the axis with the correct hour, day, month and year.

Graph plotting

Graph plotting in MAGICS (see Fig. 2) is the plotting of line charts (curves), bar charts and area charts within a set of axes. An area chart is where the area between two curves is shaded. The user has full control over linestyle, colour and thickness of graphs and, if required, a legend describing the graph can be plotted. Bar charts and area charts can use all the MAGICS shading facilities.

Symbols can be plotted on the curves and a specified area around curves can be blanked. Curves can be drawn straight or rounded, where a smoothing algorithm is applied to the curve. There are options for dealing with missing data.

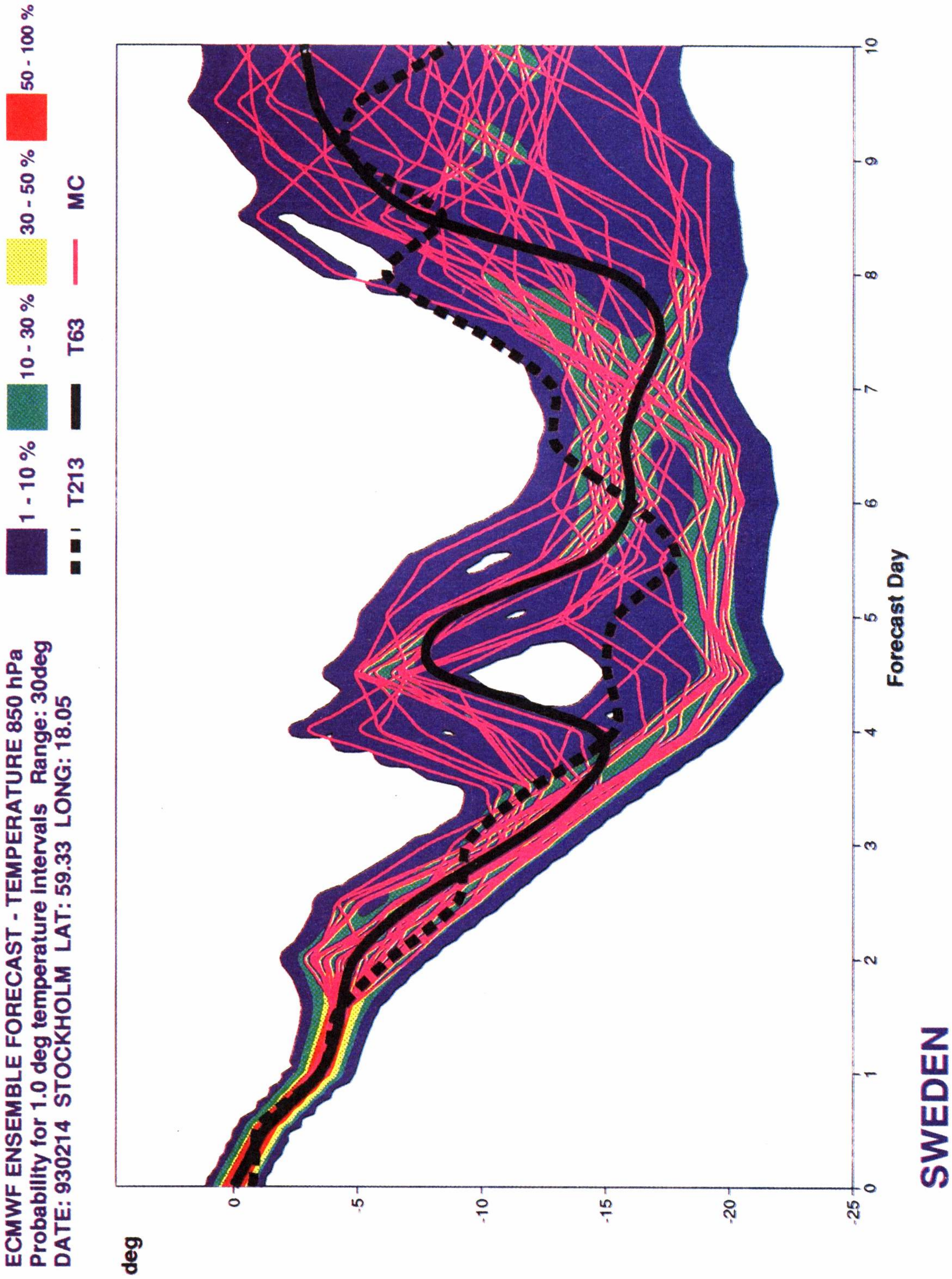


Fig. 2 Example of graph plotting, area shading and legends.

Symbol plotting

Symbol plotting in MAGICS is the plotting of different types of symbols at selected locations. A symbol in this context is a number, a text string, a WMO wind flag or a MAGICS marker. The position of a symbol may be defined by its geographical location (latitude/longitude), by its position in centimetres from the bottom left hand corner of the subpage or by its X/Y position on a graph. There are facilities in symbol plotting that allow the user to control the height and colour of each symbol and, if required, a legend describing the symbols may be plotted. Symbol plotting is achieved in MAGICS by calling action routine PSYMB which, like all MAGICS action routines, has no arguments.

Text facilities

MAGICS allows users to plot a block of text anywhere within the user's page. A text block consists of a number of text lines, up to a maximum of ten lines, and may be positioned automatically by MAGICS or specifically by the user.

Facilities exist for plotting integer, real and character values, the necessary type conversion being done by MAGICS. It is possible to plot the values of any MAGICS parameters in a text line.

Users have full control, via MAGICS parameters, over all aspects of text plotting, such as colour and style, as well as the height of text lines and the spacing between lines.

Instruction strings enable the plotting of complicated text strings and give even more user control of colour, height etc. Using instruction strings, the user has control over each individual text character and may also plot symbols, e.g. the degree sign.

Legend plotting

MAGICS legend facilities enable users to annotate their plots. Legends can be produced automatically for contouring, wind plotting, satellite imaging, graphs and symbols. For contouring, attributes like style, colour and thickness are associated with a text, describing the relevant contour interval used in the plot. Shading legend entries consist of a sample of each shading pattern used in the plot.

For wind plotting, the shape and length of arrows are associated with a text which can describe the wind speed. Also, wind arrow colours may be associated with a text to describe the significance of the colours. Wind flags, plotted each time in the legend with a full barb, are associated with a text which describes the wind strength.

The legend entries are plotted into an area known as the legend box, each legend entry consisting of three distinct parts: symbol, automatic text and user text. The user has full control over the positioning of the legend box, the number of legend entries and the way entries are plotted within the box.

MAGICS shading

MAGICS shading allows users to shade areas in contouring, graph plotting, coastline drawing etc. with varying intensities, patterns and colours. MAGICS parameters exist to control all the features and options in shading and their default values should ensure that, for most plots, a reasonable shaded plot can be achieved without having to set most of the shading parameters. Three different methods of shading are available; dot, hatch and solid.

Shading enables more information to be plotted on a map without causing confusion. It is useful for highlighting specific details, e.g. negative temperatures, rainfall above 20 millimetres etc. Shading can make a map more easily understood, particularly when used with MAGICS legend facilities.

Specification groups

A MAGICS specification group consists of a set of MAGICS parameter values, where each group may consist of one or more parameter values.

Users may request MAGICS to save, retrieve or delete specification groups in memory. It is also possible to save and retrieve specification groups in a user file. If required, the user may alter this file by using normal text editors. MAGICS can read specification groups from a user file, which has been created by MAGICS or by the user.

Specification groups are useful when writing MAGICS application programs that produce, for example, a predetermined sequence of plots. Users can pre-define and save specific groups of MAGICS parameters that are used for each picture in the sequence. They simplify programming and make programs more readable.

- Patrick O'Sullivan

* * * * * * * * * * * * *

RAID DISKS ON THE C90

Introduction

We have recently acquired four DA62 disks for the C90. These disks are of a type known as RAID. This is an acronym for Redundant Array of Inexpensive Disks. The DA62s conform to a format known as RAID-3. Each DA62 consists of five drives or spindles, of which four contain the data written by the program and the fifth contains parity, or error correction, information which allows the system to reconstitute the data, even if one of the other spindles is completely destroyed. It is to this that the "Redundant Array" part of the acronym applies. Unfortunately, the "Inexpensive" part of the acronym is more of a desire than a fact. Although the spindles themselves may be inexpensive, the extra circuitry, hardware and software required adds extra expense.

Besides the redundancy features of these disks another feature is that of speed. The data is written to all four (plus parity) spindles simultaneously. Each DA62 spindle is effectively a DD62 disk, hence each DA62 drive transfers data at 4 times the speed of a DD62. Since the DD62 transfer rate is 8 MBytes/sec, that of a DA62 is 32 MBytes/sec.

During provisional acceptance of these disks we performed some fairly severe tests. One test consisted of removing the cables from one of the spindles while running a program which was reading and writing a file on the disk. The system reported a recovered error on the disk but the program did not notice any problem at all.

Disk error recovery

It is interesting to note what happens in cases where an error is detected by the system when reading from a DA62 disk. Normally the reads (and writes) are done in 5-spindle mode. If the system gets a read error on one of the spindles, it then goes through its normal sequence of functions to try to reread the data from that spindle, just as it would do to try and correct a DD62 error. If it manages to read the data then the data is passed to the user program and the read is flagged as a recovered or corrected disk error. If, however, it fails to recover the data from this spindle, then it uses the error correction data from the fifth (or parity) spindle to reconstitute the data and the read is flagged as a recovered stream error. In both cases the user program receives the correct data with no indication that a problem existed. If the system notices a catastrophic failure of one of the spindles, as it did when we pulled out the cable, then it will automatically disable the failed spindle and put the DA62 into 4-spindle mode. In this case it will only read (and write) from the four good spindles and if the failed spindle is a data spindle, as opposed to the parity spindle, it will use the error correction data, stored on the parity spindle to reconstruct the data. All this will be transparent to the user job and the data will transfer at normal speed; the data reconstruction will not slow down the transfers.

Disk reconstruction

When a DA62 changes to 4-spindle mode (and this can be done by the operators and engineers as well as by the system) the normal course of action is to get the engineers to look at the spindle. This they can do with diagnostics that run while UNICOS and user jobs are still running. If the diagnostics indicate a failure of the spindle necessitating its replacement, then they can "hot-plug" a spare spindle. By this is meant that they can unplug the failed spindle and replace it with a spare, without having to stop UNICOS or user work. Once the spare is connected, they can "hot fix" the disk. A program is run which tells the operating system to put the drive in 5-spindle mode for writing, but leave it in 4-spindle mode for reading. The program will then go through each cylinder on the disk, read it, and write it back out. It will continue until all the cylinders on the disk have been read and rewritten. When this has been done it tells the system to put the disk into 5-spindle mode for both reading and writing. All the time that this is happening normal user jobs can be reading or writing files that reside on the disk. Obviously, the speed with which data is accessed during this phase will be affected, since a lot of head movement will be taking place on the disk. We have seen degradation of between 2 to 5 times using the disk test program. This reconstruction of the disk usually takes between 15 and 30 minutes to accomplish.

- Neil Storer

* * * * * * * * * * * * *

The following two articles are reprinted here by courtesy of SCD Computing News. The first appeared in the April 1992 issue, and the second in the May 1992 issue of that publication. Jeanne Adams chaired the International Programming Languages Committee of the ISO (International Standards Organisation), and chaired the ANSI Committee which developed Fortran 90. She also co-authored the book "The Fortran 90 Handbook: Complete ANSI/ISO Reference".

FORTRAN 90 ARRAY EXTENSIONS AVAILABLE ON THE CRAY COMPILER

Fortran 90 includes new array facilities, many of which are currently featured in CF77 5.0.

An array consists of a set of values, all of the same type. To declare that a variable is an array, you must state the dimensions in a DIMENSION or a type statement. The dimension declarator specifies the lower and upper bounds in each of seven possible dimensions. This is consistent with what has been possible in the Fortran 77 standard.

In Example 1, X is a one-dimensional array of 100 elements and Y is three-dimensional. The lower bounds of X and the lower bounds of Y along each dimension are assumed to be 1 by default. The A and B arrays are one-dimensional and are of the same size - ten elements each. The lower bounds of A and B are 1 and 0, respectively. ARRAY is four-dimensional, with 21 elements in the first dimension, 2 in the second dimension, and 3 in the last two dimensions.

Example 1. Stating the dimensions in an array

```
DIMENSION X(100), Y(10,11,2)
INTEGER A(1:10), B(0:9)
REAL ARRAY(-10:10,2,3,2:4)
```

Array terms

When we talk about arrays, we use some common terms:

size	Product of the number of elements along each dimension
rank	The number of dimensions
extent	The number of elements along a dimension
shape	A vector of extents in each dimension
conformable	Having the same shape
broadcast	Making a scalar the same shape as an array
attribute	A declared characteristic of a variable

The *size* of an array is the product of the number of elements along each dimension. In Example 1, ARRAY has a size of 378 (that is, 21*2*3*3). The *rank* of an array is simply the number of dimensions in the array (four for ARRAY in Example 1). Rank is always known at compile time; a scalar has rank 0.

The *shape* of an array is a vector of the *extents* in each dimension - for ARRAY in Example 1, (21,2,3,3).

Two arrays are *conformable* if they have the same shape. For instance, X(10,2) conforms with Y(0:9,2:3); both extents are 10 and 2. On the other hand, X(10,2) is not conformable with Z(2,10) because the extents in the corresponding dimensions are different. The vector of extents for X is (10,2); the vector of extents for Z is (2,10); thus X and Z are not conformable. Even though the size is the same, the shape is not.

A scalar in an assignment with an array is *broadcast* to an array so that the scalar has the same shape as the array and is conformable.

In Fortran 90, variables including arrays may be declared with certain characteristics, such as dimensionality or type: for example, INTEGER, REAL, or LOGICAL. These characteristics are called *attributes*. (Chapter 5 of the Fortran 90 standard contains a description of these additional attributes.)

Array references

Other terms that apply to arrays include:

array reference	Identifies a value or a set of values
array bounds	The lower and upper limits in each dimension
subscript	Identifies one element of an array
subscript triplet	Identifies a range of elements in a dimension

There are at least three kinds of array references:

whole array	A	(without subscripts)
array element	A(14)	(with subscript)
array section	A(12:50:10)	(a range of elements)

(You can also reference an array section by using a vector subscript; this method will be discussed in a future article.)

Whole-array reference

A whole-array reference, which appears without a subscript, identifies all the declared elements in the array.

Array-element reference

When you reference a single element of an array using a subscript, the subscript is an integer expression that may contain functions or array elements. (Using array elements of functions in array bounds is not standard-conforming in FORTRAN 77, but is in Fortran 90 and is allowed in CF77 5.0 as well.) A subscript must evaluate to a scalar integer.

Examples of array elements are:

```
X(40)
Y(3,4,1)
B(0)
ARRAY (-5,2,2,4)
X(M(0))
X(ABS(M(3))).
```

Each of the previous array references indicate one array element. An array element is a scalar.

Array-section reference

An array section references a range of elements from the whole array; a subscript triplet specifies this range along a dimension. An array section is defined as a part of an array where a subscript triplet appears in at least one dimension.

The form of a subscript triplet is:

[lower] : [upper] [:stride]

The triplet consists of the lower subscript value (bound) of a range, followed by the upper subscript value (bound) of a range along a dimension, followed by the stride; each term is separated by a colon. The *stride* is the increment specified within the range; if it is omitted, the stride is 1. If the lower bound or the upper bound of a subscript triplet is omitted, the declared lower or upper bound is used. Note that while an array element is a scalar, an array section is an array.

In the case of A(12:50:10), the array section specified begins with A(12), followed by A(22), A(32), A(42). In this case, the section is a rank-1 array consisting of four elements of A.

In the case of B(10,5,2:8:3), B is three-dimensional. The triplet is in the third dimension referencing B(10,5,2), B(10,5,5), and B(10,5,8) as the range of elements. As mentioned previously, an array section itself is an array - in this case a rank-1 array.

Example 2 shows the use of array sections.

Array assignment

Assignment in Fortran is the evaluation of an expression on the right-hand side of an equal sign followed by placing the value in the variable or array on the left side of the equal sign. The expression on the right may contain a reference to an array element (or scalar), a section of an array, or a whole array. Note that assignment behaves as if the expression on the right is evaluated completely before assignment to the variable or array on the left is performed.

Array assignment is permitted when the expression on the right has the same shape as (that is, it is conformable with) the array on the left. Any portion of the right-hand side may be scalar, and the value of the expression is broadcast to the size of the array on the left.

Example 2. Use of array sections

```
PROGRAM ARRAYS
 IMPLICIT NONE
 INTEGER I, J
 REAL X(5,6), Y(5,6), Z(5,6)
 X = 0.0
 Y = 0.0
 Z = 0.0
 X(3,2:4:1)=1.0
 Y(4,2:6:2)=2.0
 Z(1:2,3:6)=3.0
 PRINT "(A)", "X(3,2:4:1) = 1.0"
 PRINT "(6F10.0/)", ((X(I,J), J=1,6),I=1,5)
 PRINT "(A)", "Y(4,2:6:2) = 2.0"
 PRINT "(6F10.0/)", ((Y(I,J), J=1,6),I=1,5)
 PRINT "(A)", "Z(1:2,3:6) = 3.0"
 PRINT "(6F10.0/)", ((Z(I,J), J=1,6),I=1,5)
END

X (3,2:4:1) = 1.0 yields:
0 0 0 0 0 0
0 0 0 0 0 0
0 1 1 1 0 0
0 0 0 0 0 0
0 0 0 0 0 0

Y (4,2:6:2) = 2.0 refers to the following section of Y:
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 2 0 2 0 2
0 0 0 0 0 0

Z (1:2,3:6) = 3.0 refers to the following section of Z:
0 0 3 3 3 3
0 0 3 3 3 3
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

In the program fragment in Example 3, since X and Y are conformable, the value of the Y array is assigned to X. If X and Y did not conform, the code would be nonstandard (illegal) and an error diagnostic should be produced. In the line X = 4.0 * COS (1.1), the right-hand side of the assignment is broadcast to be conformable with X.

Array assignment is done element by element, without a particular order specified in the standard. *The effect must be as though all the assignments were done simultaneously.* It is important to note that in some cases where there are dependencies in the evaluation, a DO loop and an array assignment may be different. This is true in both Fortran 90 and CF77 5.0. Another way of looking at this effect is that any values from the array on the left may be used in evaluating the expressing on the right, since the evaluation is completed before any assignments are made.

Example 3. A program fragment

```
      DIMENSION X(100), Y(100)
C     Whole array assignment, X and Y are conformable
      X = Y
C     is the same as:
C     do i = 1,100
C       x(i) = y(i)
C     end do
C     Whole array assignment to a scalar
      X = 4.0 * COS (1.1)
C     is the same as:
C     do i = 1,100
C       x(i) = 4.0 * cos (1.1)
C     end do
C     Whole array assignment of a mixed expression
      X = 4.0 * Y
C     is the same as:
C     do i = 1,100
C       x(i) = 4.0 * y(i)
C     end do
```

Syntax

Some of the formal syntax that could be used for Cray CF77 5.0 is:

assignment-statement	is	*variable = expr*
variable	is	*scalar*
	or	*array*
	or	*array element*
	or	*array section*
	or	...
section-subscript	is	*subscript*
	or	*subscript triplet*
	or	*vector subscript*
subscript-triplet	is	*[subscript]:[subscript] [: stride]*
vector-subscript	is	*rank-1-integer-expression*

- Jeanne Adams

* * * * * * * * * * * *

ARRAY SPECIFICATIONS IN FORTRAN 90 AND CF77 5.0

Many Fortran 90 array features are included in the Cray CF77 5.0 compiling system. A discussion of whole arrays and array sections appears in the previous article. This article discusses array specifications.

An array specification in Fortran establishes that a variable or function is an array and declares the lower and upper bounds, from which the array's extents, size, rank, and shape are determined. As noted below, there are four kinds of array specifications: explicit-shape, assumed-size, adjustable, and automatic. (Fortran 90 has defined additional facilities for dynamic arrays: assumed-shape and deferred-shape arrays. These latter two require dope vectors or descriptors. They are not available on CF77 5.0 and will not be discussed here.)

In an explicit-shape array, the extents are fixed at the beginning of a program. In assumed-size, adjustable, and automatic arrays the extents may vary and be dynamically assigned in a subprogram. Fortran 90 offers more flexibility than Cray Fortran for doing dynamic array assignments; FORTRAN 77 does not allow dynamic storage.

Types of array specifications

Explicit-shape arrays. An explicit-shape array is an array whose bounds are scalar constant expressions of type integer.

Syntax

The following syntax applies to explicit-shape and assumed-size arrays in Fortran 90:

explicit-shape-spec-list	is	*explicit-shape-spac [,explicit-shape-spec] ...*
explicit-shape-spec	is	*[lower bound :] upper bound*

where

lower-bound	is	*specification expression*
upper-bound	is	*specification expression*
assumed-size-spec	is	*[explicit-shape-spec-list,] [lower-bound:]**

Note: For adjustable and automatic arrays, the lower and upper bounds may be expressions containing variables or functions.

Assumed-size arrays. An assumed-size array is an array that is a dummy argument in a subprogram. (Note: Dummy arguments in a subroutine or function are placeholders for actual arguments when a procedure is called or referenced.) The size of the last axis is assumed from the actual argument. The syntax has no upper bound in the last dimension, using an * (asterisk) to mark that the size is assumed. If the array is a character array, the length of the character elements must be considered. Assumed-size arrays are a FORTRAN 77 feature carried into Fortran 90.

Adjustable arrays. An adjustable array is a dummy argument whose size is determined at execution time by the array declarator of the actual array used in each call. Adjustable arrays are a Fortran 90 feature and a Cray extension to FORTRAN 77.

Automatic arrays. An automatic array is a local array in the subprogram that uses dummy arguments or COMMON variables as bounds. Space is allocated on entrance to the subprogram and deallocated on exit. These arrays are not saved, nor are they initialised. Automatic arrays are a Fortran 90 feature and a Cray extension to FORTRAN 77.

Adjustable and automatic arrays appear to be the same; they look alike. However, the adjustable array and its bounds are dummy arguments in the subprogram, whereas the automatic array is a local array with variable bounds as arguments. The bounds expressions for both adjustable and automatic arrays may contain array elements, functions, and noninteger variables; this is a Fortran 90 feature (that is, these expressions are an extension of FORTRAN 77).

Using arrays

The specifications of an array may take different forms, depending on how the programmer plans to use the array. In Fortran 90 and Cray Fortran, explicit-shape arrays (those containing lower- and upper-bound declarations that are constant expressions) may appear in the main program and in any subprogram. The bounds of assumed-size arrays, automatic arrays, and adjustable arrays appear as dummy arguments in a subprogram or as variables in a COMMON block. Assumed-size arrays and adjustable arrays appear as dummy arguments, while automatic arrays are local variables.

A sample program

In the sample program in Example 1 (the results of which are shown in Example 2), the main program contains only explicitly declared arrays. In the subprogram, R and S are adjustable arrays. R has dummy arguments in both dimensions, while S has a fixed lower bound and a dummy argument for the upper bound. Q is an automatic array, local to the subroutine. Only the bounds are dummy arguments. Space for Q is allocated upon entry to the subroutine, and deallocated upon exit. K is a FORTRAN 77 assumed-size array with an * (asterisk) in the last dimension of the array specification.

Example 1. Sample program with array specifications

```
      PROGRAM TEST
       IMPLICIT NONE
       INTEGER I, J, N
       PARAMETER (N = 2)
C        Fortran 90 and Cray Fortran are not case sensitive
C        The following arrays are explicitly declared
       INTEGER K(4,N +1)
       REAL X(0:5), Y(0:5)
       REAL  A(0:5), B(0:5)
C
       y = 10.0
       x = y
       b = 5.0
       a = b
       K = 2
       I = 3
       J = 2
       CALL SUB (X, A, K, I, J)
       I = 4
       J = 3
       CALL SUB (B, Y, K, I, J)
      END

      SUBROUTINE SUB (R, S, K, I, J)
       IMPLICIT NONE
C        The following are adjustable arrays
C        R, S, I, J are dummy arguments
       REAL R (I, J), S(0:J)
C        The following is an automatic array
C        Q is local, J is a dummy argument
       REAL Q (J)
C        The following is an assumed-size array
C        K varies only in the last dimension
       INTEGER K(4,*), L, I, J, M
C        Initialize Q since automatic arrays are not initialized
       Q = 0.0
C
       PRINT 100, "R is", R
       print 100, "S is", S
       PRINT 100, "Q is", (Q(L), L = 1,J)
       PRINT 101, "K is", ((K(L,M), L = 1,4), M = 1,3)
  100  FORMAT  (A, (3F5.1/))
  101  FORMAT  (a, (4I5/))
       RETURN
      END
```

Example 2. Results of the sample program

```
R is
        10.0 10.0 10.0
        10.0 10.0 10.0
S is
         5.0  5.0  5.0
Q is
         0.0  0.0
K is
         2    2    2    2
         2    2    2    2
         2    2    2    2
R is
         5.0  5.0  5.0
         5.0  5.0  5.0
         5.0  5.0  5.0
         5.0  5.0  5.0
S is
        10.0 10.0 10.0
        10.0
Q is
         0.0  0.0  0.0
K is
         2    2    2    2
         2    2    2    2
         2    2    2    2
```

\- Jeanne Adams

* * * * * * * * * * * *

STILL VALID NEWS SHEETS

Below is a list of News Sheets that still contain some valid information which has not been incorporated into the Bulletin set (up to News Sheet 296). All other News Sheets are redundant and can be thrown away.

No.	Still Valid Article
204	VAX disk space control
205(8/7)	Mispositioned cursor under NOS/VE full screen editor
207	FORMAL changes under NOS/VE
224	Job information cards
235	VAX public directory - how to create
253	Copying/archiving NOS/VE catalogs to ECFILE Copying complete UNICOS directories to ECFILE
254	UNICOS carriage control
260	Changes to PUBLIC directories for VAX users
261	Meteogram system on UNICOS
265	Lost UNICOS outputs submitted via RJE or VAX Microfiche changes
266	Reminders on how to import/export magnetic tapes
268	Changes to WMO FM 92 GRIB

No.	Still Valid Article
270	Changes to the Meteogram system
271	New ECFILE features on UNICOS
280	UNICOS on-line documentation: docview
281	File transfer via FTP (possible problems)
283	New features for Member State batch users (RQS 1.1)
284	UNICOS 7 features & differences
286	Improving the performance of "model" jobs on the Cray Y/MP8 at ECMWF (pre-allocating disk space)
294	Changes to the Meteogram system
296	Introduction of the new (TCP/IP) ECFILE on the Y/MP16-C90

* * * * * * * * * * * * *

ECMWF ANNUAL SEMINAR

Every year ECMWF organises a seminar to discuss progress in a selected topic related to numerical weather forecasting. This year the seminar will be con-sponsored by EUMETSAT. The seminar forms part of our educational programme and is aimed at young, mainly post-graduate/post-doctorate scientists in the ECMWF Member States. The subject of the 1993 seminar will be:

Developments in the Use of Satellite Data in Numerical Weather Prediction

and will take place during the week 6 - 10 September 1993.

The seminar will discuss developments over the last five years in the use of satellite data for numerical weather prediction, data assimilation and model validation.

An account will be presented of recent developments in the assimilation of TOVS and other satellite data. The seminar will also discuss the impact of satellite data on the modelling of radiation and of the hydrological cycle, and will look forward to data expected from new operational satellite instruments.

Posters and registration material have been mailed to the national meteorological services and universities in the ECMWF Member States.

- Els Kooij-Connally

* * * * * * * * * * * * *

FOURTH WORKSHOP ON METEOROLOGICAL OPERATIONAL SYSTEMS

22 - 26 NOVEMBER 1993

Introduction

The planned biennial **Workshop on Meteorological Operational Systems**, to be held at ECMWF 22-26 November 1993, will be the fourth in the series.

With the advent of new supercomputers and workstations based on common platforms, such as UNIX, X-Windows and NFS, operational systems are developing fast allowing the exploration of new and ambitious ways of using medium-range forecast products through the introduction of experimental forecasting systems, enhanced data processing and interactive data visualisation.

The workshop will review the state of the art of meteorological operational systems and address future trends in the use of medium-range forecast products, data management and meteorological workstations.

Use and predictive skill of numerical forecasts in the medium range

ECMWF implemented an experimental ensemble forecasting system in December 1992. Products are made available to Member States through the dissemination system, in graphical form via telefax and through access to the online data bases on the Centre's computers. Other operational centres have experimented with different approaches to skill prediction.

Workshop participants are invited to present their views on, and expectations from, the experimental ensemble forecasts and the use of numerical products in the medium range. It is envisaged to review the usage and the development of ensemble forecast products during the presentations and in the working group sessions.

Operational data management systems

Meteorological applications have progressively been standardised on the use of UNIX systems. Operational systems will be reviewed highlighting the use of high level languages and design standards to facilitate the portability of software systems. Special attention will be given to the use of meteorological data bases providing a key towards efficient data access. The increasing rôle of commercial data bases in meteorological data management will be reviewed.

Meteorological workstation systems

The important anticipated benefit of the Meteorological Workstation (MWS) is the potential for integrating various data sources. The meteorologist has access to and sees a large and growing amount of data. The MWS should provide a way for the forecaster and the research scientist to assimilate large amounts of data very quickly. They not only want to see the raw data but also have a need to manipulate data. Performance is crucial both for interactive data access and visualisation. Animation has become an essential tool and 3D is beginning to find its niche.

Current and planned meteorological workstation systems will be presented and discussed during the workshop. Commercial companies will be invited to demonstrate their systems in support of meteorological applications.

- Horst Böttger

* * * * * * * * * * * *

ECMWF CALENDAR 1993

30 August	**ECMWF HOLIDAY**
6 - 10 September	**Seminar** - Developments in the use of satellite data in Numerical Weather Prediction
22 - 24 September	**Scientific Advisory Committee,** 21st session
4 - 6 October	**Technical Advisory Committee,** 18th session
11 - 13 October	**Finance Committee,** 51st session
14 - 15 October	**Policy Advisory Committee,** 1st session
15 - 17 November	**Workshop** - The role of the stratosphere in Numerical Weather Prediction
22 - 26 November	**Workshop** - Meteorological Operational Systems
1 - 2 December	**Council,** 39th session
24 - 28 December	**ECMWF HOLIDAY**

* * * * * * * * * * * * *

ECMWF PUBLICATIONS

Technical Report No. 70 — Scientific assessment of the prospects for seasonal forecasting: a European perspective

Proceedings of seminar on validation of models over Europe, 7-11 September 1992 (2 volumes)

Forecast and verification charts to 31 March 1993

* * * * * * * * * * * * *

INDEX OF STILL VALID NEWSLETTER ARTICLES

This is an index of the major articles published in the ECMWF Newsletter series. As one goes back in time, some points in these articles may have been superseded. When in doubt, contact the author or User Support.

USEFUL NAMES AND 'PHONE NUMBERS WITHIN ECMWF

			Room*	Ext.**
DIRECTOR	-	David Burridge	OB 202	2001
HEAD OF OPERATIONS DEPARTMENT	-	Michel Jarraud	OB 010A	2003
ADVISORY: Available 9-12, 14-17 Monday to Friday				2801
Other methods of quick contact:	-	Telex (847908 ECMWF G)		
	-	Telefax (+44 734 869450)		
	-	VMS MAIL addressed to ADVISORY		
	-	Internet mail addressed to Advisory@ecmwf.co.uk		
REGISTRATION				
Project Identifiers	-	Pam Prior	OB 225	2384
User Identifiers	-	Tape Librarian	CB Hall	2315
COMPUTER OPERATIONS				
Console	-	Shift Leaders	CB Hall	2803
Console fax number	-	+44 734 499 840		
Reception Counter	-	Tape Librarian	CB Hall	2315
Tape Requests	-	Tape Librarian	CB Hall	2315
Terminal Queries	-	Norman Wiggins	CB 026	2308
Telecoms Fault Reporting	-	Michael O'Brien	CB 028	2306
ECMWF LIBRARY & DOCUMENTATION - DISTRIBUTION	-	Els Kooij-Connally	Library	2751
LIBRARIES (ECLIB, NAG, etc.)	-	John Greenaway	OB 226	2385
METEOROLOGICAL DIVISION				
Division Head	-	Horst Böttger	OB 007	2060
Applications Section Head	-	John Hennessy (acting)	OB 014	2400
Operations Section Head	-	Bernard Strauss	OB 328	2420
Meteorological Analysts	-	Andreas Lanzinger	OB 314	2425
	-	Ray McGrath	OB 329	2424
	-	Anders Persson	OB 315	2421
Meteorological Operations Room	-		CB Hall	2426

		Room*	Ext.**	Bleeper
COMPUTER DIVISION				
Division Head	- Geerd-R. Hoffmann	OB 009A	2050	150
Systems Software Sect.Head	- Claus Hilberg	OB 104A	2350	115
User Support Section Head	- Andrew Lea	OB 227	2380	138
User Support Staff	- Antoinette Alias	OB 224	2382	154
	- John Greenaway	OB 226	2385	155
	- Norbert Kreitz	OB 207	2381	156
	- Dominique Lucas	OB 206	2386	139
	- Pam Prior	OB 225	2384	158
Computer Operations				
Section Head	- Peter Gray	CB 023	2300	114
Security, Internal Networks and				
Workstation Section Head	- Walter Zwieflhofer	OB 140	2352	145
GRAPHICS GROUP				
Group Leader	- Jens Daabeck	CB 133	2375	159
RESEARCH DEPARTMENT				
Head of Research Department	- Anthony Hollingsworth	OB 119A	2005	
Computer Co-ordinator	- David Dent	OB 123	2702	

* CB - Computer Block
OB - Office Block

** The ECMWF telephone number is READING (0734) 499000, international +44 734 499000, or direct dial to (0734) 499 + last three digits of individual extension number, e.g. the Director's direct number is (0734) 499001.

DEC MAIL: Contact scientific and technical staff via VMS MAIL, addressed to surname.

Internet: The ECMWF address on Internet is **ecmwf.co.uk**
Individual staff addresses are firstname.lastname, e.g. the Director's address is David.Burridge@ecmwf.co.uk