The Current State of Meteorological Modelling

by

D M Burridge

ECMWF

1. INTRODUCTION

Numerical weather prediction involves the solution of large systems of coupled partial differential equations which describe atmospheric motions. A brief description of the physical and mathematical basis of numerical models of the atmosphere is given in the paper by Bengtsson in this volume. For our purposes the governing equations may be written formally as

$$\frac{\partial X}{\partial t} = \underline{D}(X) + \underline{P}(X) \qquad (1)$$

where X is any model variable .... wind, temperature, humidity etc. ... The problem is 3-dimensional, with the dependent variable X being a function of spatial co-ordinates x, y and z and a function of time t. The term $\underline{D}$ represents the "dynamics" - advection, pressure forces etc - and the term $\underline{P}$, normally referred to as the "physics", is a source/sink term which for large-scale models of the atmosphere describes physical processes such as the evaporation and condensation of water, solar heating, infra-red cooling, and frictional drag at the earth's surface. For fine-scale models of thunderstorms a description of cloud physics processes is required; this adds new variables to the problem - cloud water drop-size spectra and for each drop-size an electrical charge spectrum. With present computing power, severe compromises must still be made for such models. In contrast to the "dynamics" embodied in the term $\underline{D}$, the "physical" processes, $\underline{P}$, are intermittent and are often represented by on/off processes. The

programming or implementation for these processes largely effects the efficiency of vectorization on computers such as the CRAY-1 and the CYBER-205, with different strategies being required for different computers. For large-scale models such problems have been overcome by carrying out redundant calculations. The effect on parallel processing is discussed in the paper by Dent in this volume.

In the rest of this paper I shall describe the basic numerical techniques in current use - Section 2 - and in Section 3 future requirements for global weather forecasting models.

2. NUMERICAL SCHEMES AND THEIR IMPLEMENTATION

Three different techniques are in general used for numerical models of the atmosphere - finite differences, finite element and spectral. With each of these techniques a 3-dimensional grid of points is required. At the time of writing typical horizontal grid spacings for global models are 200 km on 15 levels (in the vertical) ranging from the earth's surface to heights of 25 km- these resolutions requiring approximately 200*100*15 = 300,000 grid points. For the vertical representation in numerical models of the atmosphere the finite difference technique predominates though there have been some successful implementations of the finite element technique. The next three sub-sections 2.1 to 2.3, describe each of the techniques and their implementation. The fourth sub-section, 2.4, discusses the implementation of time-stepping algorithms.

2.1 Finite differences

In the early 1970s, almost all models of the atmosphere were based on finite difference techniques, and, of the three main approaches, finite difference methods are perhaps the most readily understood and the most easy to implement. Model variables are defined at grid points and nowhere else; the derivatives in $\underline{D}(X)$ being approximated by differences. The simplest difference technique require 9 points (i, j; i:1-200, j:1-100) in the horizontal at each of 3 levels (k; k:1-15) in the vertical. The "physics" requires, for an efficient implementation, the presence of all dependent variables for a particular vertical column. (15 levels of information for a 15-level model). The memory sizes of the present generation of supercomputers will, normally, only allow a fraction of the model's grid point description be retained in memory at any one time. The normal approach is to retain in memory vertical slices of the data for a

few values of j - for our model problem data arrays for memory resident variables would be DIMENSIONED (200,15); typical vector lengths for vector calculations being 200. The full description is retained on a secondary memory - normally a high speed disc system driven by asynchronous i/o drivers. However whereas such an approach is very efficient on CRAY computers, indeed there is little overhead if the secondary storage is a CRAY-SSD system, the optimal approach for the CYBER-205 is to design a "memory-resident" model such as that developed by the British Meteorological Office; this permits the use of very long vectors. Obviously a "memory-resident" model limits flexibility particularly with regard to resolution increases.

2.2 Finite element schemes

The essence of finite element schemes is to expand the model's dependent variables in terms of a set of basis functions, each of which is a low-order polynomial of compact support (i.e. each basis function is non-zero only over a small sub-domain or element of the full region); a familiar basis set being the linear-spline or "hat-functions"

$$e_i(x) = \begin{cases} \dfrac{x-x_{i-1}}{x_i-x_{i-1}} & x_{i-1} \leqslant x \leqslant x_i \\ \dfrac{x_{i+1}-x}{x_{i+1}-x_i} & x_i \leqslant x \leqslant x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

Insertion of this representation into the model's equations leads to a set of equations of the form

$$\underline{m}\,\frac{\partial X}{\partial t} = \underline{d}(X) + p(X) \qquad (2)$$

where $\underline{m}$ and $\underline{d}$ are large sparse matrices and X is a vector of amplitudes. To all intents and purposes the algebraic form of the right hand-side of

this equation is similar to that obtained with finite difference treatments. The main difference, computationally, between finite element and finite difference is the requirement to invert the sparse matrix $\underline{m}$. Apart from the purely computational cost, this inversion normally presents considerable technical problems. This arises because, in general, the structure of $\underline{m}$ requires the whole of the right hand side of equation 2 in memory for the inversion. However, schemes can be designed in which $\underline{m}$ can be written as a product of matrices each of which is relatively simple to invert. For example with a bi-linear spline description for the horizontal representation, $\underline{m}$ can be written as the product $\underline{m}_x * \underline{m}_y$, where $\underline{m}_x$ and $\underline{m}_y$ correspond to the x and y discretizations respectively; both matrices being narrow banded. The matrix $\underline{m}_x$ is normally inverted by Fourier analysis and synthesis and the matrix $\underline{m}_y$ by a L/U decomposition involving a forward elimination step and back substitution step. In this case the inversion of $\underline{m}$ is similar to the problem of carrying out Fourier analysis and re-synthesis in two dimensions, for which there exists many efficient "out of memory" implementations.

2.3 Spectral methods

The last decade has seen the rapid development of spectral methods for the horizontal description in large-scale models of the atmosphere. As with finite element approach dependent variables are represented by expansions in terms of a set of basis functions - normally orthogonal polynomials - however the description is "global" rather than local. For global models, the basis functions are normally surface harmonics (Legendre functions for the North/South representation and Fourier series for the East/West representation). The most efficient implementation of the spectral technique requires two related representations - the spectral description

and a grid point description. Horizontal derivatives are first calculated exactly at each grid point using the spectral representation. The second stage is to calculate $\underline{D}(X)$, using the exact values for derivatives; and $\underline{P}(X)$ at grid points. A time-stepping procedure determines new values of the dependent variables at grid-point from which new values of the spectral representation are calculated. The implementation of a spectral model on the CRAY-XMP22 is discussed in the paper by Dent in this volume.

In contrast with finite difference techniques, spectral methods are very non-local; the complete spectral representation is needed to calculate values at a single grid point and each grid point value is required to calculate individual amplitudes in the spectral representation. With the present memory sizes on super computers this requires, for an efficient implementation, at least a two-pass structure for the calculation. For the ECMWF spectral model the complete spectral representation is retained in central memory and is used in the first pass to calculate primary variables and their derivatives at grid points; these grid point values are stored on an external i/o device in vertical slices. In the second pass these data are re-input into memory and a new 3 dimensional spectral description is computed - this includes all dynamical and physics calculations. The i/o requirements are similar to those for finite difference and finite element techniques. Indeed, there is a great deal of similarity between all the implementations. Essentially, they "look like" grid point techniques and the same basic data structures can be used.

The spectral model that ECMWF plans to introduce into operations in the spring of 1985 will require 16 mw of storage (storage medium CRAY-SSD) for primary variables and a little less than the 2 mw of the CRAY-XMP22's central memory.

2.4 Time-stepping algorithms

The procedures outlined in the preceding three sub-sections are used to calculate the right hand-side of equation

$$\frac{\partial X}{\partial t} = \underline{D}(X) + \underline{P}(X)$$

The forecasting problem is, given initial values of X=X(t) at time t, to determine values of X at successive times using a time-stepping procedure. The simplest schemes are explicit, and a commonly used scheme is

$$\frac{X(t+\Delta t) - X(t-\Delta t)}{2\Delta t} = \underline{D}(X(t)) + \underline{P}(X(t-\Delta t)) \qquad (3)$$

This scheme requires complete model states at two time-levels (t and $t-\Delta t$) in order to calculate values at the new time-level $t+\Delta t$. Explicit schemes such as (3) are easy to implement and do not impose severe programming problems.

Despite their simplicity, explicit schemes can carry a serious disadvantage for meteorological models. This arises because the governing equations describe fast time-scale phenomena - gravity waves with horizontal phase speeds of 300 ms^{-1} - in addition to the slow time-scale weather events. The earliest numerical weather forecasting models were based on sets of equations which "filtered-out" fast time-scale phenomena. The presence of the fast time-scales require short time-steps - small Δt - with explicit schemes. Implicit schemes can be used, and can be designed to be unconditionally stable - allowing the use of very large time-steps; a typical example being

$$\frac{X(t+\Delta t) - X(t-\Delta t)}{2\Delta t} = \underline{D}\left\{\frac{(X(t+\Delta t) + X(t-\Delta t)}{2}\right\} + \underline{P}\left\{\frac{(X(t-\Delta t)) + X(t+\Delta t)}{2}\right\}$$

However, since $\underline{D}$ and $\underline{P}$ are a very complicated non-linear expression, the determination of $X(t+\Delta t)$ requires a prohibitively expensive inversion procedure, which for operational weather forecasting would be impractical.

The compromise is to treat only a linear part of $\underline{D}$ implicitly, that is, use a semi-implicit scheme. A typical semi-implicit scheme is

$$\frac{X(t+\Delta t) - X(t-\Delta t)}{2\Delta t} = \tfrac{1}{2}\{\underline{D}_o(X(t+\Delta t) + \underline{D}_o(X(t-\Delta t))\} + \underline{D}'(X(t)) + \underline{P}(X(t-\Delta t))$$

where $\underline{D}' = \underline{D}-\underline{D}_o$

Since $\underline{D}_o$ is a linear operator the inversion process is simple; usually $\underline{D}_o$ can be reduced by simple algebraic manipulation to a Laplacian.

The application of semi-implicit techniques to spectral models results in particularly efficient schemes, since the basis functions are normally chosen to be the eigenfunctions of the Laplacian and in this case the inversion process requires only a multiplication. Because they are not fully implicit, semi-implicit schemes are in general conditionally stable, and typically we require

$$\Delta t < \alpha(\frac{h}{U_{max}})$$

where h is a horizontal grid length, U_{max} the maximum horizontal wind and α a constant of order 1.

Recently, Lagrangian schemes have been developed in order to allow further increases in the time-step. The essential features of Lagrangian schemes are illustrated in Fig. 1 below (Fig. 1 is a space-time, x-t, diagram for a 1-dimensional problem.)

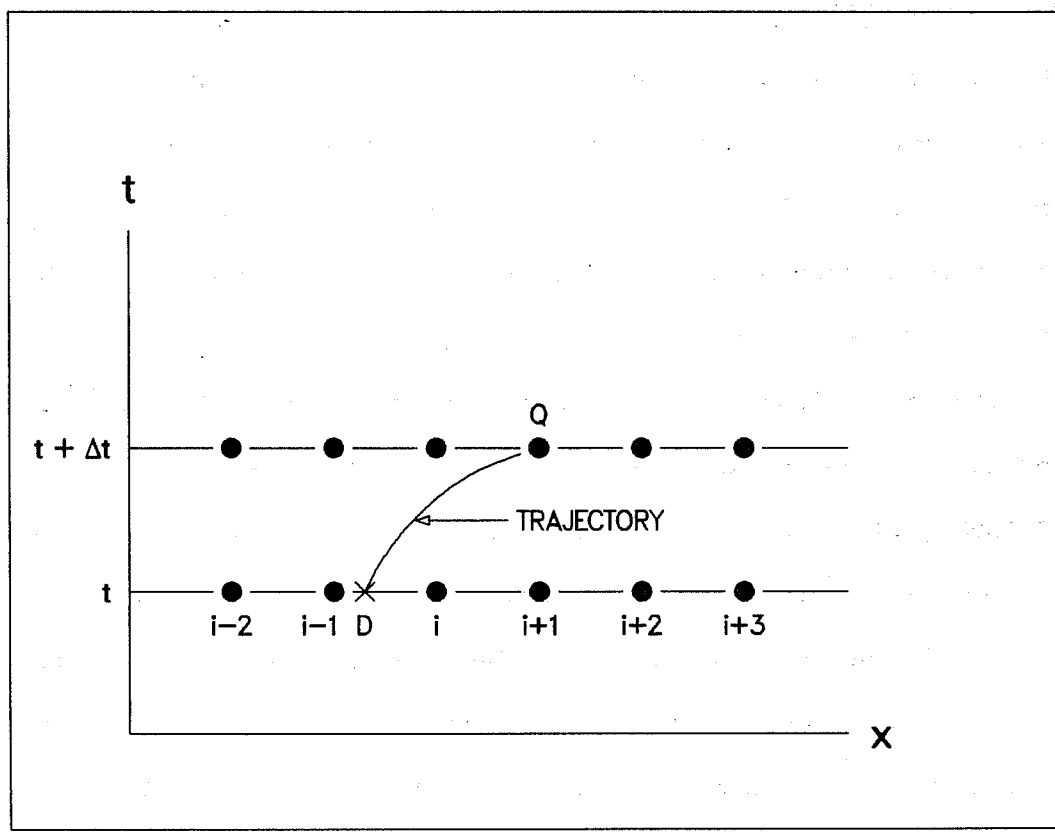

Fig. 1 Space-time x-t diagram for a Lagrangian scheme

In order to determine the values of the dependent variables at Q($x=x_i$, $t=t+\Delta t$) a "departure" point D is require - the curve DQ being a trajectory; values at Q are identical to those at D. Since D is not normally a grid-point the values at D have to be determined by interpolation from surrounding grid point values; the accuracy of this interpolation determining the accuracy of the scheme. In numerical weather prediction, Lagrangian techniques are used for the advection processes - the slow time-scale phenomena - and when combined with semi-implicit treatments of the fast time-scales allow the use of large time-steps.

3. FUTURE DEVELOPMENTS

The schemes outlined in Section 2 will undoubtedly continue to provide the basic numerical schemes of future forecasting models. However, the approach to modelling the dynamics may require a different use of the basic equations to provide a better description of dynamical processes and a more widespread use of implicit methods. By the end of the decade we shall see a dramatic increase in the resolution of operational global forecasting models. It can be expected that the operational use of a global model with 600*300 grid points on 25 levels will be justified by the research and development currently being carried out by many research groups. Such a model, if based on a spectral representation, will require about 100 mw of storage; this requirement increasing by about 20 mw for each additional dependent variable added to the model. For such a model to be operationally feasible for medium range forecasting - forecasting up to 10 days ahead - the minimum requirements is for a machine with fifty (50) times the throughput of a CRAY-1S. Undoubtedly this power will only be provided by significant increases in the number of processors and large increases in central memory.

Machines with large amounts of central are being planned - successors to the current CRAY and CYBER machines for example - and will, I am sure be built. For the programmer the two most important characteristics are the presence or absence of shared memory - that is shared between processors - and the multi-tasking support. About 30% of the code in the present generation of "out-of-memory" models is used to manage the i/o and the central memory for primary and temporary variables, and it is to be hoped that the next generation of large - memory supercomputers will have efficient mechanisms (hardware/software) to manage shared memory and/or a memory hierarchy.

Turning finally to multi-processing, I feel optimistic that parallel processing, particularly for weather forecasting models, overheads will reduce through algorithm re-design; the scheme described by Dent in this volume being a good first start with a large-scale code.