Hydraulics Research
Wallingford

AN ASSESSMENT OF THE USES OF THE ICL DAP FOR ESTUARY MODELLING

A J Cooper PhD

Report No SR 30
January 1985

Registered Office: Hydraulics Research Limited,
Wallingford, Oxfordshire OX10 8BA.
Telephone: 0491 35381. Telex: 848552

This report describes work carried out under Contract No DGR/465/35, funded by the Department of Transport from April 1982 to March 1984 and thereafter by the Department of the Environment. Any opinions expressed in this report are not necessarily those of the funding Departments. The DoE (ESPU) nominated officer was Mr A J M Harrison. The work was carried out by Dr A J Cooper in the Tidal Engineering Department of Hydraulics Research, Wallingford, under the management of Mr M F C Thorn. It is published with the permission of the Department of the Environment.

SUMMARY

In the past the applications of numerical modelling techniques to predict the flow of water and the movements of sediment and pollution in estuaries and coastal waters have often been limited by the excessive cost of using a grid size fine enough to resolve the shoreline, changes in bed levels and flow patterns. The development of more powerful computers has opened the way to obtain greater resolution in computer modelling than hitherto and, in particular, the ICL Distributed Array Processor (DAP) at Hydraulics Research Ltd is ideally suited to such detailed tidal computations.

The DAP is a unit of store with 4096 processors which all carry out the same instruction simultaneously. A special version of fortran called DAP fortran is used to facilitate the use of arrays with operations on all of the elements simultaneously. The DAP is connected to a host machine (an ICL 2972) through which data must be transferred.

DAP models have been successfully used for a number of studies of water flow in estuaries. These include patched and unpatched depth integrated flow models and three dimensional models. Transport models for heat and mud have been developed but have not yet been applied. In these applications it has been found very important to map the model onto the DAP grid with care to make the maximum use of the possibility of parallel processing. If this is done it has been found possible to run a flow model considerably faster (up to about a hundred times faster) than on a conventional serial computer.

CONTENTS

1 INTRODUCTION

Hydraulics Research Limited has recently acquired an ICL Distributed Array Processor (DAP). This is a parallel processing computer in which up to 4096 arithmetical operations can be performed simultaneously. The processors are arranged in a 64 x 64 array, each one being connected to its North, South, East and West neighbours. Fig 1 is a schematic illustration of the DAP. Each processor has 4K bits of memory associated with it. All 4096 processors may be instructed to perform simultaneously the same operations on the different data under them.

Such a computer is ideally suited to two-dimensional calculations of water flow and the transport of mud and pollution in estuaries, such as HR has previously performed on serial computers. Three-dimensional modelling on the DAP can be achieved by performing calculations at all the points within each layer simultaneously but processing the layers serially.

Hydraulics Research has undertaken research into effective ways of using the DAP and the resulting programs have been used in repayment studies. It has been found possible with the DAP to use more grid cells or a finer resolution than would be financially acceptable on a serial computer. Some programs have not been adapted for DAP use but the necessary techniques have been worked out as described below.

It has been found convenient to produce DAP programs which are consistent with the existing HR suite of two-dimensional estuary programs. This should allow, for example, the results of a flow model computed using the DAP to be input to a transport model run on a serial machine and for DAP model results to be processed with existing programs to produce plots of velocity vectors, isotherms, float tracks etc.

2 USING THE DAP

As the DAP has no input/output facilities it can only be used when attached to a host machine. At HR this is an ICL 2972 computer. A DAP program comprises a host section written in ordinary fortran and a DAP section written in DAP fortran, which is a language designed to handle parallel processing. The host section reads in data and then calls a subroutine in the DAP section and when a return statement is reached control returns to the host program to allow output of the results. In calling a DAP subroutine from the host no arguments are allowed so all data are passed to the DAP in common blocks. In calling a DAP subroutine from another DAP subroutine arguments are permitted as usual. DAP subroutines called from the host are called ENTRY subroutines in DAP fortran. Before using data in the DAP it must be converted to DAP storage mode by calling special subroutines. DAP fortran is fully described in Ref (1).

As well as the ordinary (scalar) variables of standard fortran, DAP fortran allows vectors and matrices which have 64 and 64 x 64 elements respectively and may be integer, real or logical. A logical matrix is also called a logical mask, it occupies only one bit plane of the DAP and logical manipulation is performed very rapidly.

In DAP fortran simple arithmetic operations can be written such as,

```
A = B+C
```

which, if A, B and C are real matrices, causes all 4096 elements of B to be added to the corresponding elements of C simultaneously and the result stored in matrix A. Thus in a 2D model there is a large reduction in the number of DO loops. This makes some programs simpler to understand and more concise than programs written in standard fortran. Logical masks are important largely because of a technique called left hand side indexing. For example if LAND is a logical mask,

```
A (.NOT.LAND) = B
```

will set the values of matrix A to those of matrix B only in sea areas leaving LAND areas unchanged. Such a technique is necessary if all of the assignments are carried out simultaneously, as it is impossible to check each point in turn for whether it is a sea or land point.

The host section is compiled as a normal fortran module but the DAP section has a three stage compilation process. After syntax checking the first stage compiled code goes into an AMF library. The second stage is to a CIF library and the last stage is a collect stage when external subroutines are collected and an OMF file produced. The second stage of this process can be very time consuming.

3 NUMERICAL MODELS OF ESTUARIES

The types of numerical models of estuaries used are summarised in Fig 2. The first type is a cross-section averaged model. The equations are reduced to one space dimension (along the estuary axis) although this may be complicated by having more than one branch (eg in deltas). This simple kind of model cannot represent horizontal circulations (eg different flood and ebb channels) or vertical circulations (which may be due to wind or to salinity variation). Horizontal circulations can be represented in the second type of model which is depth integrated. This is a two dimensional x-y-t model. Vertical circulations can be modelled by the third type which is width integrated. This is a two dimensional x-z-t model. Finally the 3D model can represent, in principle all of the processes taking place in an estuary.

The x-y-t model is currently the most frequently used as the velocity profile through the depth often has a simple form. This model solves the "shallow water" or long wave equations. It is often possible to model the water movements independently of the transport of heat or salt in such a model as vertical circulations are not included. Given the pattern of water movement it is then possible to compute the movement of heat, salt or mud with a larger timestep. In this way it is possible to simulate the movement of many pollutants or one can suppose the flow to be periodic and simulate periods of months or years.

4 WHICH PROGRAMS SHOULD BE IMPLEMENTED ON THE DAP

The main advantage of the DAP is that it is possible to execute certain programs much faster than on a serial machine as many operations can be performed at the same time. However not every program can be made to take advantage of this feature. The times taken on the DAP for simple arithmetic operation on scalar quantities tend to be longer than on serial machines as simple processors are used, so it is necessary to find which programs can take advantage of this opportunity.

If a program can take advantage of the DAP's parallel architecture then it may be substantially cheaper than on a conventional computer. It also makes possible runs that would take too long on a serial machine to be feasible (perhaps 20 hours). This consideration is very important for three-dimensional models.

Another economic aspect is the need to use as many processors as possible. So a model with only 400 cells takes as long to run as one with 4000 and the greatest advantage of the DAP is if one can use all 4096 processing elements.

The principal programs of interest are the flow and transport programs in two and three dimensions. The other programs in the HR suite of estuary programs take much less processor time and little would be gained by using the DAP.

5 EXPLICIT OR IMPLICIT NUMERICAL SCHEMES

Existing HR transport programs for mud or heat use explicit finite differences. The flow programs use in some cases explicit and in other cases implicit differences. The advantage of an implicit scheme is that a longer timestep is possible before the calculation becomes unstable. However there is more calculation to do at each timestep in an implicit scheme so a timestep perhaps twice the explicit value is necessary to make the implicit scheme cheaper. Sometimes a short timestep is needed for reasons of accuracy (for example in modelling harbours) so the explicit scheme may be preferable.

For the DAP an explicit scheme has the advantage of being simple to program and taking very little time per timestep as the calculation proceeds simultaneously at each point of a two-dimensional grid.

Implicit techniques on the DAP are, however, quite feasible. The implicit scheme currently used by HR on a serial computer is a "fractional step" scheme in which there are two steps. Firstly in the E-W direction an implicit calculation is performed along each row inverting a tridiagonal matrix with explicit differencing in the N-S direction. Then implicit calculations are performed along the N-S columns using explicit differences in the E-W direction. On the DAP there is a program available (F04_TRIDS_64 SQ, see Ref 2) to solve up to 64 tridiagonal matrix invertions simultaneously. Therefore a high degree of parallelism could be incorporated into an implicit model.

Such a model has not been written for the DAP due to lack of time but it should be straightforward to do. However, the computing time per timestep would be much greater than that in the explicit program and a very long timestep would be needed to make such a program financially advantageous.

Another possible advantage of the implicit scheme will appear below. The explicit flow model requires diffusion to stabilise it as a completely stable explicit representation of the nonlinear advection terms has not so far been found.

6 THE DAP PROGRAMS

The various programs discussed here share certain features. In order to fit into the already existing HR program system the input and output are, so far as possible, the same as in a serial program. The database file structure for the HR modelling system is described in Ref 3. The most noticeable feature is that the arrays are stored in compressed form using an integer array of pointers called KARRAY. This may greatly reduce the storage requirement of a program and as all quantities are held in single dimensioned arrays it also keeps down CPU time. KARRAY simply numbers the (potentially) active cells in the model starting top left and moving to the right along each row. All land cells are numbered 1.

The structure of the host program is essentially the same as a serial flow program up to the point that the actual flow calculation begins.

Most of the input/output is the same as in a conventional program (however boundary conditions are treated differently, see Section 9). Before entering the DAP program section a subroutine EXPAND is called for each variable held in compressed form. This simply produces a 64 square fortran array which is passed to the DAP program via a common block. When the calculation has been done it is necessary to call a COMPRESS subroutine before the results can be made use of. This process is in addition to the fortran to matrix mode conversion described briefly above in Section 2. Any variables, such as geometry data, which are not needed again do not need to be compressed but just remain in the DAP store. The EXPAND and COMPRESS subroutines become complicated in the cases of patched models or models with grids of dimension greater than 64. In this case the expanded form becomes an array of matrices.

Because host to DAP conversion is clearly expensive it is best not to undertake it more often than is essential. For this reason the DAP program has been set up to continue timestepping until the first dump either to disc or to a line printer file is required. After leaving the DAP at the end of the run the endfile is written in the same way as in a serial flow program.

The DAP programs so far written are all based on this way of working. The individual programs will now be considered in more detail.

6.1 Explicit Two Dimensional Flow Model

The two-dimensional (x-y-t) explicit flow model has been developed from an HR benchmark model translated by ICL into DAP fortran. The model uses a space and time staggered grid exactly as on a serial computer. The finite difference scheme is also the same except that as there is no direction of sweep the partially implicit "angled derivative" scheme for computing the nonlinear advection terms, uu_x, vu_y etc cannot be used. This scheme has been replaced by backwards time level representation which would give exactly the same result for a steady flow. Unfortunately this scheme is potentially unstable and to prevent the model blowing up a certain amount of diffusion must be included in the model. The amount needed to stabilise the scheme is approximately

$$\frac{1}{2}\Delta t(u^2 + v^2)$$

which in most cases is less than the physical diffusion (of the order of velocity x depth) so the program can be used for practical applications.

In this model it is straightforward to include the effect of a steady windstress on the water surface.

Several studies have been carried out using the x-y-t flow model. In some of these cases either the domain has been larger than 64 square or a region with different grid size has been patched in dynamically (or in one case both). These extensions of the basic scheme are described below in Sections 7 and 8.

The numerical scheme has been tested by running the same model using the DAP formulation, an explicit (serial) scheme and an implicit (serial) scheme. The model was a 1500m grid representation of the Severn Estuary. All of the models gave very similar results when a timestep less than the explicit limit (about 43 sec) was used but the implicit model gave rather different results if a Courant number of about 10 were used. At one station there was at one time an elevation difference of ½m and at another the velocity/time curve showed a delay of ½hr (Fig 3). Thus it seems that the DAP formulation is acceptable but care is needed when using the implicit scheme with a large timestep. The DAP scheme may run up to 110 times faster than the explicit scheme. For a large timestep the inplicit scheme may be cheaper than the explicit scheme but it also tends to become inaccurate.

6.2 Two Dimensional Transport Model

Transport models are used to calculate the movement of some material moving with the water (e.g. heat or sediment) based on a given flow field. The equations represent the conservation of this material but also include source and sink terms due to heat loss, sediment erosion and so forth. The most commonly used transport model is a two-dimensional x-y-t model which uses the results from a 2-D x-y-t flow model as described above.

The HR two dimensional transport programs for heat, mud or pollutant are based on the use of upstream differences. This is a very stable numerical scheme which has the disadvantage of introducing numerical diffusion of the order

$$\frac{1}{2}\Delta x\sqrt{(u^2 + v^2)}.$$

This term tends to smear variations and certain improvements to the program can be made to decrease this unwanted effect when the preservation of strong concentration gradients or peak concentrations is important.

The algorithm used for serial computers is very suitable for use on the DAP as the scheme is completely explicit. The upstream differencing scheme can be implemented using a logical mask. This allows all the calculations where u is positive to proceed in one step and all those where u is negative in another step. The program is rather quick and simple. A mud transport model has been implemented including a very simple representation of the processes of settling and erosion of mud beds.

To reduce diffusion one can use a "flux-corrected transport" algorithm (Ref 4). This is based on removing the numerical diffusion error in the upstream differencing scheme (thereby giving the centred scheme.) However, the centred scheme is notorious for giving rise to spurious spatial oscillations so the error flux is included only to the extent that wiggles do not arise. This has the advantage of retaining some numerical diffusion where it is needed but otherwise using second order accurate differences. The programming of this scheme on a serial computer was not completely explicit so the DAP version gives results which are not identical to it, however the result of a heat transport model with flux-corrected transport has been found to be acceptable.

Alan Owen (Ref 5) has pointed out how upstream differences can be computed very efficiently if the logical mask defining the upstream direction is obtained by equivalencing with the sign bit of the velocity matrix.

Currently the HR program just sets

```
MASK = U.GE.0.0
```

which is easier to understand and reasonably quick.

6.3 Three Dimensional Flow Model

The advantage of using the DAP is felt at its greatest with the 3D flow model. In this case it is possible to run simulations with the DAP which could not be considered on an ordinary serial machine as the computing time would be quite unreasonable. With the use of the DAP such models may take a period of hours to run; on the ICL 2972 the equivalent would be several days of continuous running.

The DAP version of the 3D flow model is developed from the serial form described in Ref 6. Certain changes have been made for use on the DAP. The horizontal terms are modelled as in the 2D flow model. The vertical terms are computed by implicit differencing, this involves an implicit sweep from the surface to the bed and back again (for velocities) and in the reverse direction for concentraions (heat, salt or mud). In this process several arrays of matrices are set up containing the quantities needed for the Gaussian elimination process.

This means that the program has a rather large storage requirement. The total DAP store under each processing element is 4K bits or about 120 real numbers (of about 6 significant figure accuracy). With this amount of storage the maximum practicable number of layers in the 3D model seems to be about 10. The 3-D model will be fully described in a subsequent report.

7 EXTENSION TO DOMAINS LARGER THAN 64 SQUARE

The DAP processors form a 64 square array. If a flow or transport model requires a larger mesh than this then it is quite possible to divide the region of interest into chunks each of size 64 square if an explicit numerical technique is used. The calculation proceeds on each chunk in turn and they are then matched up along the edges prior to the next timestep. Of course if two 64 square chunks are used then the execution time of the program will be doubled.

This technique has been used successfully on several projects. It has the advantages of a very low cost penalty (just matching the edges takes very little time) and the result is exactly the same as if a larger DAP array had been used.

An example of the use of this technique in practice is shown in Fig 4.

It can be seen here that the last two rows or columns in each chunk are identified with the first two in the next chunk. The effective size of each chunk is therefore 62 if matched at both ends and 63 if only matched at one end.

It may also be possible if a model has dimensions such as 100 x 30 cells to fit it onto one DAP in the same way by splitting down the middle into two 51 x 30 chunks. This, of course, does not incur a cost penalty.

This technique of dividing up a model with dimensions larger then 64 has only been used so far for depth integrated flow models and for a two layer flow model. However it should be equally valuable for transport and 3D models.

8 PATCHING

Flow and transport models become more accurate as the grid size is decreased i.e. the resolution increased. On the other hand it is usually advantageous to have the boundary conditions of a model as far from the region of interest as possible i.e. beyond the range of influence of the works being studied. A uniform grid over the whole area, fine enough to give the required resolution in the area of interest, would lead to unacceptable computing requirements in many cases and so models have been developed in which the grid size changes from one area to another.

HR has used such patched models frequently on serial machines (Ref 7). These models have two or more regions of different grid size differing in each case by a factor of three. Flow models with patching have been implemented on the DAP. The principle of joining the two grids, taking advantage of the DAP architecture, was invented by R Gostick of CCTA. The technique involves setting up a logical mask which is used to define a one to one map from the patch boundary cells of the coarse grid to the corresponding patch boundary cells of the fine grid (see Fig 5). The matching of the patch boundaries can then be carried out very rapidly using mainly logical manipulation which is extremely rapid on the DAP. An example of velocity vectors computed from a patched flow model is shown in Fig 6.

It should be noted that the method used for patching involves taking the patch boundary data from one grid and putting it into the row or column beyond the edge of the other grid. So an inner grid in a patched model needs room all around and should have no more than 62 active cells in a direction with a patch at either end, (in fact no more than 60 because the number must be divisible by 3) see Fig 7. Work has still to be done to develop a general patched transport model.

9 HANDLING BOUNDARY CONDITIONS

Various ways of handling the boundary conditions for a flow model are possible. It was felt important not to leave the DAP too frequently during the computation so all of the boundary conditions for a whole tide are usually read in at once at the start of the run and stored in an array of real matrices. The way of assigning the boundary values from such an array was invented by S MacQueen (private communication). The necessary values are brought to the start of the array of matrices by shifting along all the values to the left by NOPENT places, where NOPENT is the total number of boundary values. The values are then scattered in a boundary value matrix using the known positions of the boundary points. A special subroutine was written to do this. The elevation (z) and u and v velocity boundary values are treated indiscriminately so there is a limitation that a particular cell should not have boundary values on more than one of z, u and v. This has not yet occurred in any actual applications of the technique. The method has not yet been fully extended to multiple grid models, at present adjustments must be made to the individual model in the event that not all of the boundary points are in the coarsest grid.

Linear interpolation in time between the times at which boundary values are stored is used to obtain a result which is completely consistent with the method HR apply on serial computers.

If required it is also possible to calculate velocity boundary values on the basis that velocity is proportional to the square root of the depth. This has been used in several studies. It may be needed if the velocity at only one point is known or if the discharge is required to have a given value. The proportionality to the square root of the depths is part of the "normal" flow approximation of open channel flow. However if this technique is used it is vital to take care about the calculation of the depth. If local depth values are used at all of the boundary points a positive feedback mechanism operates. For example on inflow the greatest velocity will be where the greatest depth of water occurs. The inflow makes the depth of water greater thereby increasing the inflow etc. It is best to use the same elevation value at all of the boundary points (usually a good approximation) and add it to the depths of the bed below datum to obtain the values of total water depth used thereby eliminating the instability.

10 CONCLUSIONS

DAP models have been used successfully by HR for a number of studies. It has been found very important to map the model into the DAP carefully so as to make the most of the parallel processing capability. If this is done it is possible to run models conveniently which would otherwise take too long on a serial computer to be considered. However certain simplifications have been made compared to the standard serial programs in order to make use of parallel processing. It may be possible to develop better programs with more adequate formulations of nonlinear terms and grid patching than those presently in use.

As an example of the speed of the DAP a single timestep of a 2D flow (x-y-t) program with dimensions no larger than 64 square takes about 25 milliseconds. The equivalent time per cell if all 4096 cells available are being used is about 6 microseconds.

This compares with about 700 microseconds for an explicit timestep using the ICL 2972 computer.

11 REFERENCES

1. DAP: FORTRAN Language, Technical publication 6918, International Computers Ltd, Second Edition 1980.

2. DAP subroutine Library, DAP Support Unit, Queen Mary College, London.

3. HYDRAULICS RESEARCH STATION. HRS Tidal modelling system. Preliminary and tidal flow programs user manual. Report IT238, 1982.

4. Boris J P and Book D L Flux correct transport III, Minimal error FCT Algorithms J Comp Phys 20, pp397-431 (1976).

5. Owen A. A note on using finite differences of ICL Distributed Array Processor Applied Mathematical Methods, p394, 1982.

6. HYDRAULICS RESEARCH STATION LTD. Formulation and development of a 3D numerical model of estuaries, Report IT254, 1983.

7. HYDRAULICS RESEARCH STATION. HRS tidal modelling system. Tidal flow programs, detailed description, Report IT239, 1982.

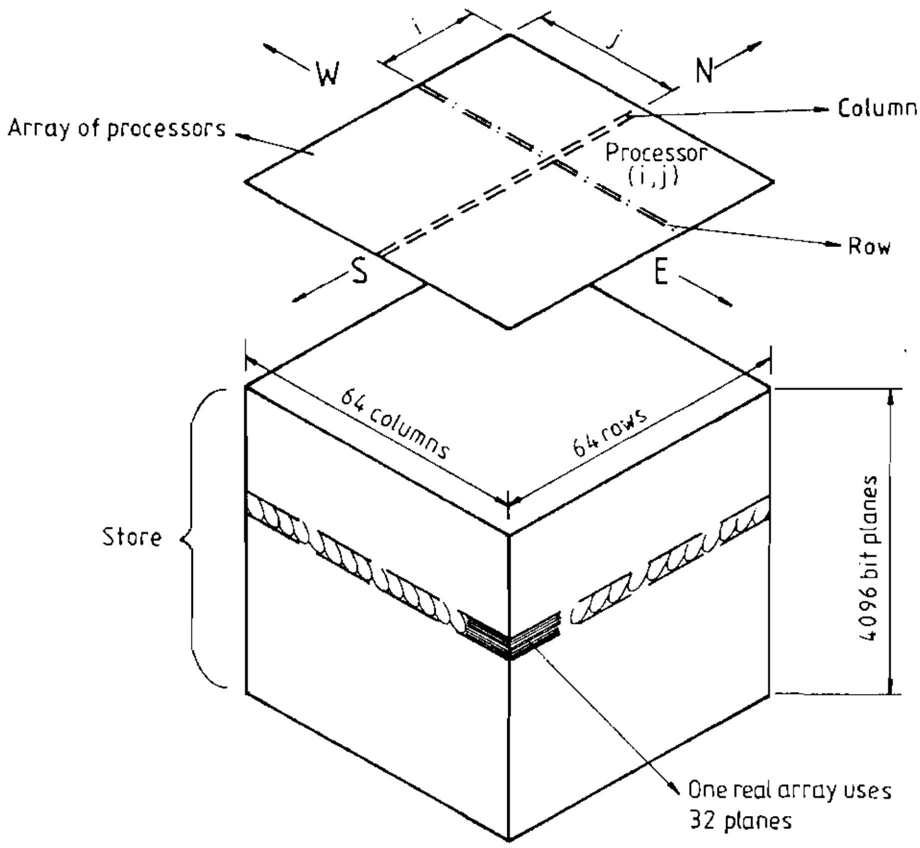


Fig 1 Schematic diagram of D.A.P.

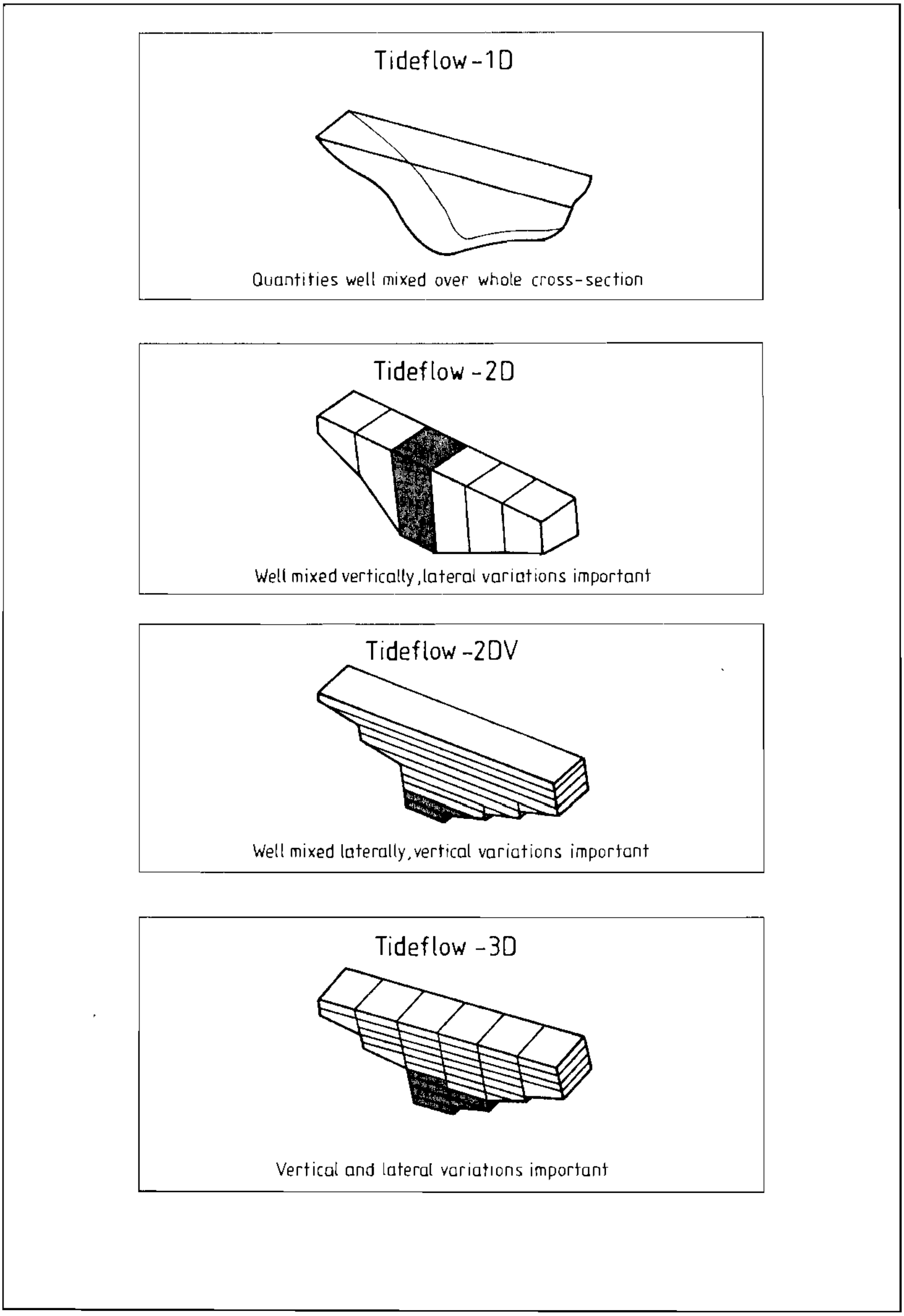
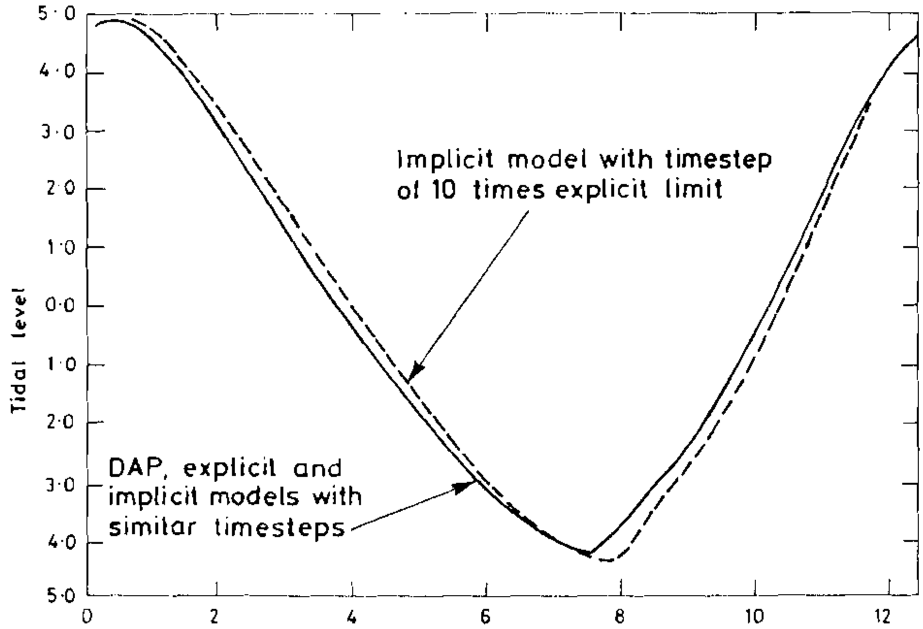
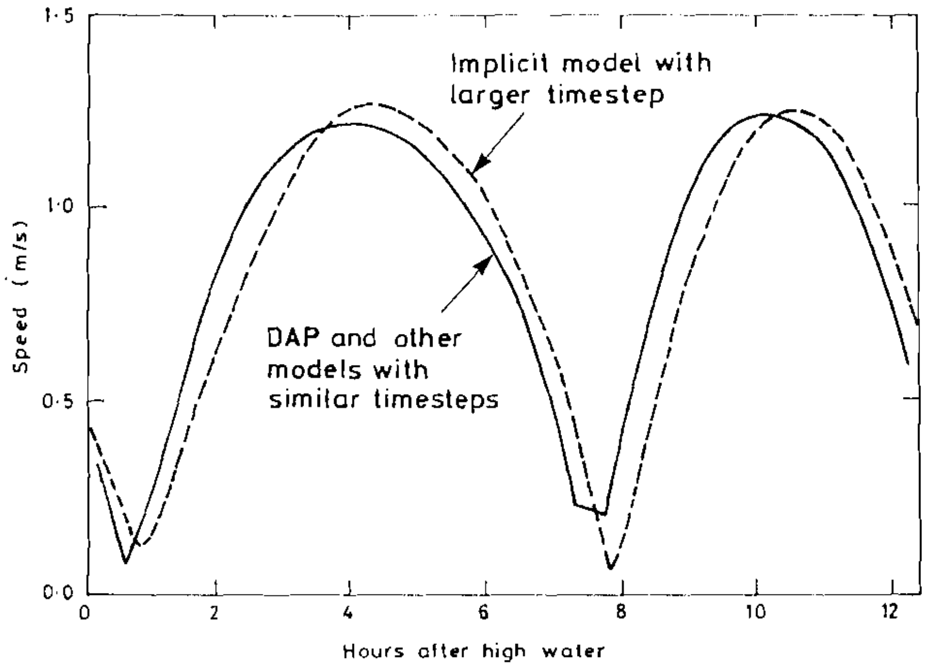


Fig 2 Types of numerical model of estuaries



(a) Tide levels



(b) Current

Fig 3 Comparison of different numerical schemes

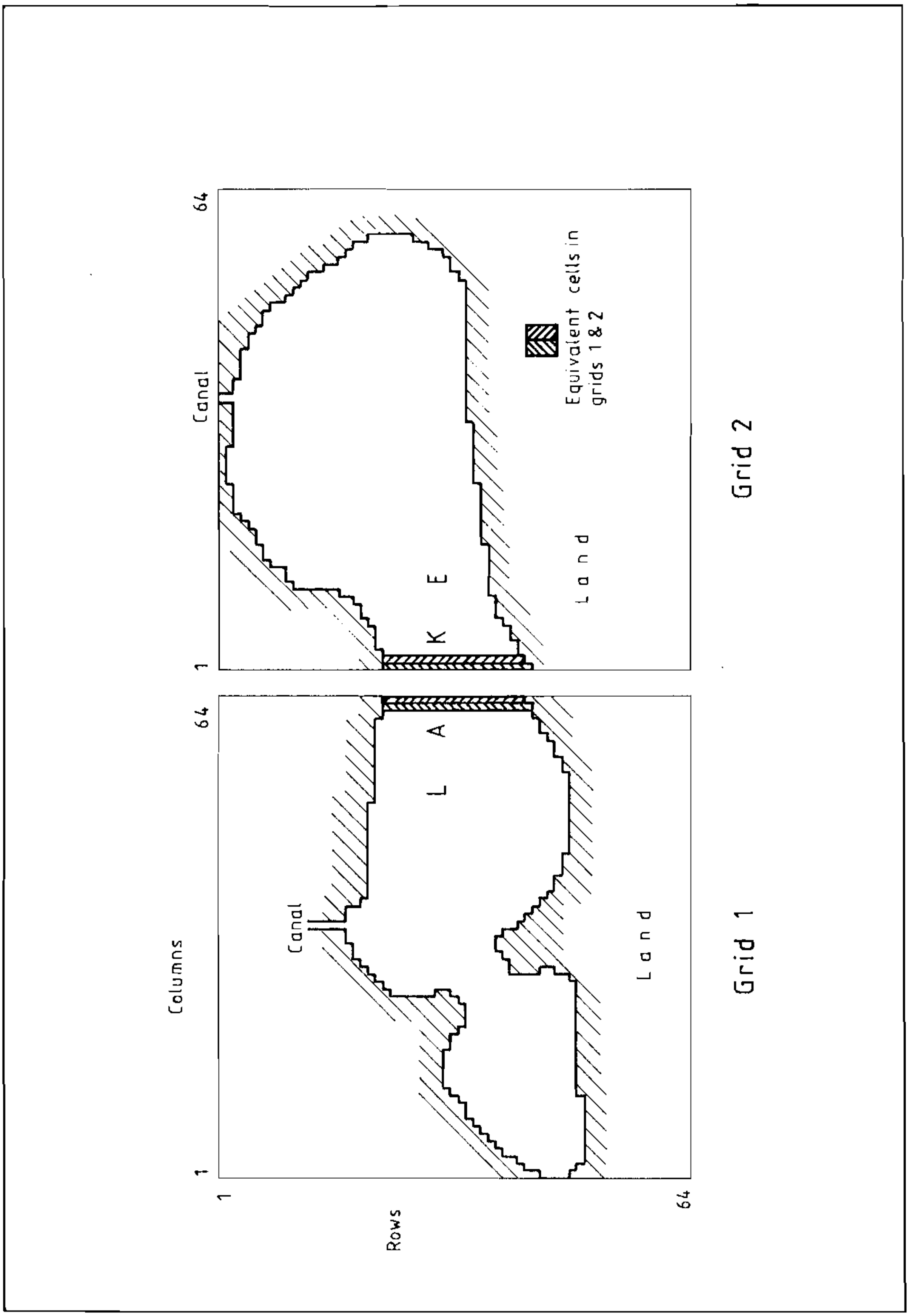


Fig 4 Mapping a model onto the D.A.P.

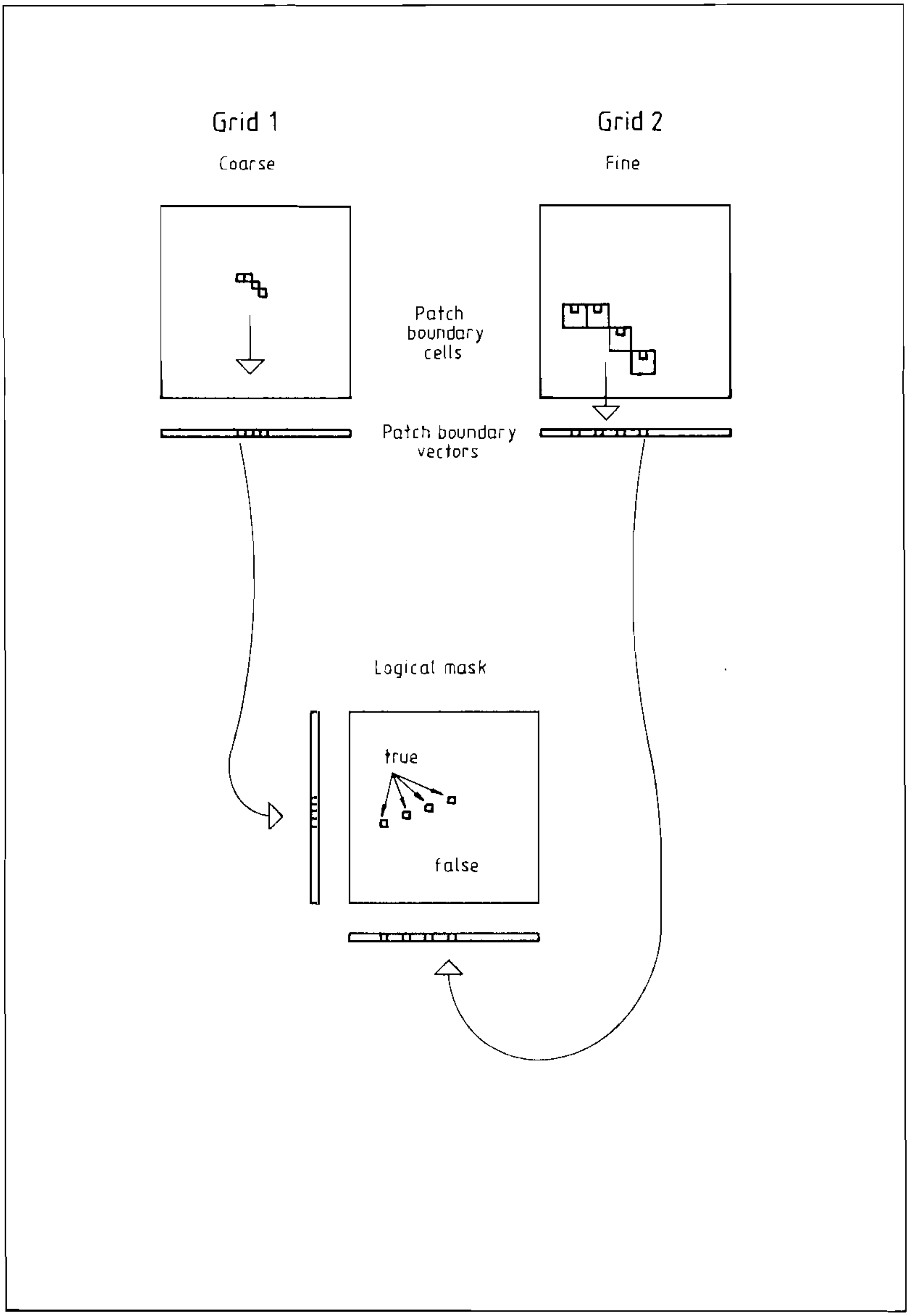


Fig 5 Patching on the D.A.P. (patching V velocities)

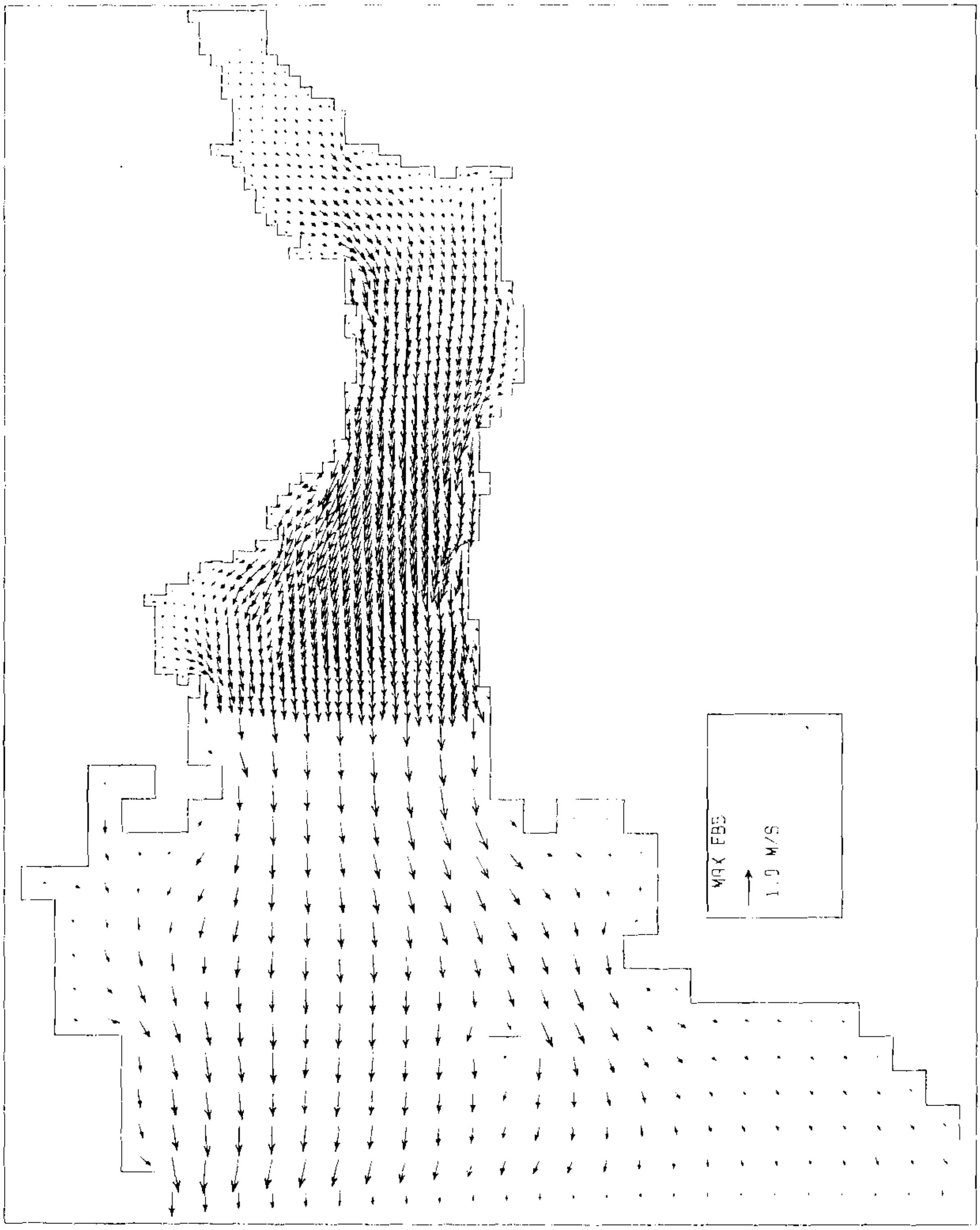


Fig 6 Velocity vectors computed for a patched model

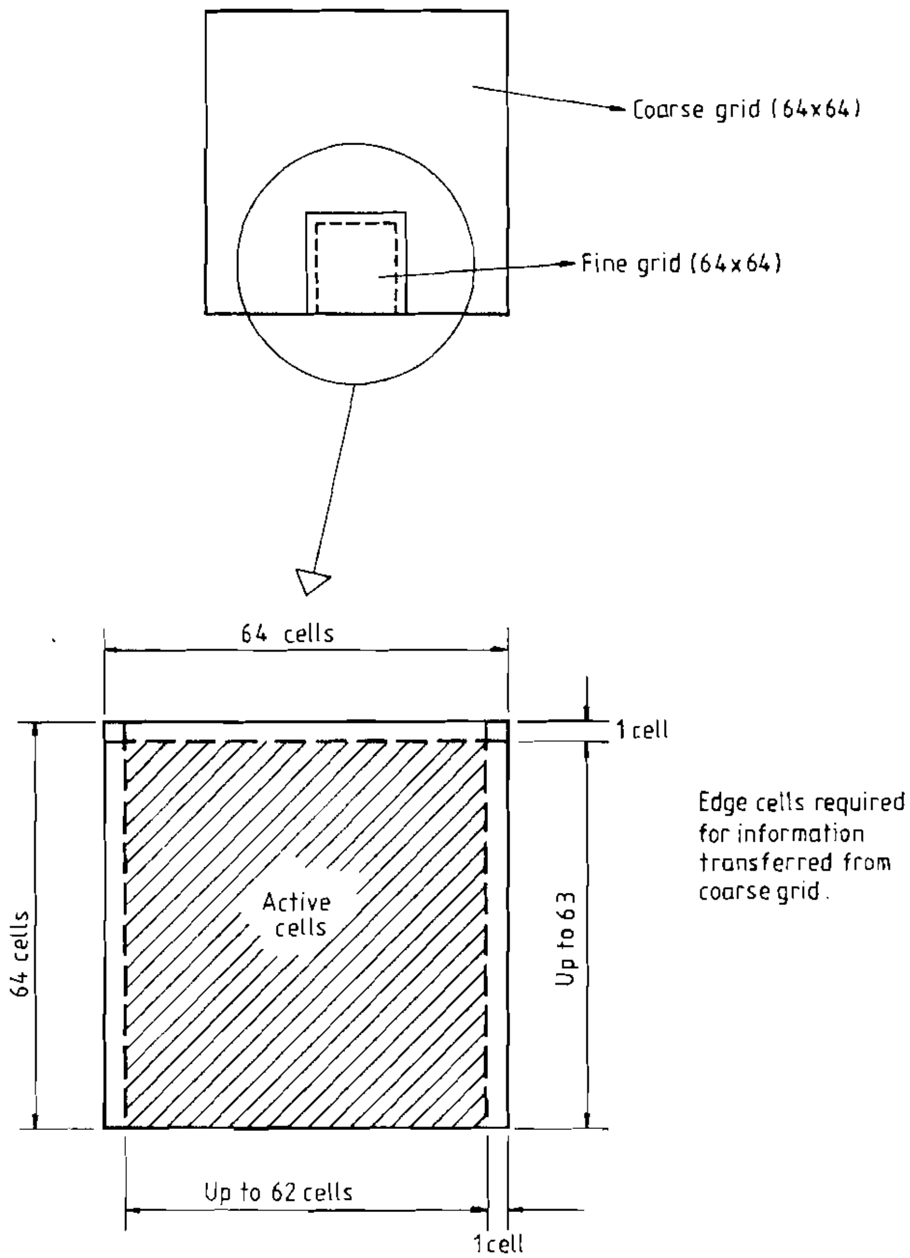


Fig 7 Inner grid in patched model