UK METEOROLOGICAL OFFICE'S PLANS
FOR USING MULTIPROCESSOR SYSTEMS

P W White and R L Wiley
UK Meteorological Office
Bracknell, Berks, U.K.

1. COMPUTING REQUIREMENTS OF THE METEOROLOGICAL OFFICE

1.1 Demand for powerful computers

Numerical modelling of the atmosphere is now an important tool in many areas of meteorology such as:

- forecasting the weather on time scales of hours, days or weeks
- investigations of the general circulation for research into matters such as climatic change
- gaining a deeper understanding of physical processes

Leading organisations in the development and application of relevant models have always been able to use the most powerful machines that technology could offer at any time and see ways of making good use of even greater processing power. It is quite clear that this position is unchanged and that there is a growing demand for much more powerful computers than existing machines such as the CYBER 205 and CRAY X-MP. Past experience suggests that useful improvements in numerical models, for example to improve the forecast quality for a target time 48 hours after observations to the same level as previously achieved 24 hours after observation time, require an increase in effective computing power of well over an order of magnitude.

1.2 The next major change

Financial constraints are such that the Meteorological Office can only replace its major computer system at intervals of ten years with only minor upgrades between replacements. A CYBER 205 was installed in 1981 and will probably be replaced in the early 1990s. At that time something like thirty times the power of the CYBER 205 will be required to handle the range of computing tasks encountered by the Meteorological Office. Peak processing rates may not give a good indication of whether this can be achieved because of the difficulties of mapping numerical models of the atmosphere on to the architecture of machines that may be available in the early 1990s. More main memory will be required to allow for larger models and to allow for a number of processes to run concurrently without paging. About 500 Mwords will be needed.

1.3 Problems in achieving the necessary performance

During the 1970s the most powerful machines that were commercially succesful worked in a generally serial manner and something close to peak performance could be obtained in a properly configured system. This situation changed with the introduction of vector-oriented machines in the 1980s because only part of any model could be fully vectorised. The questions now are:

- how will technology provide substantial increases in power?
- what resulting problems will face those implementing numerical models?

2. THE IMPACT OF TECHNOLOGY

2.1 Fast processors

Technological improvements will certainly allow the speed of existing designs to be improved. However, the obvious possibilities, such as reducing gate switching time and circuit lengths, do not offer very much hope of meeting the required targets by the early 1990s, at least in commercially available machines. In practice an improvement in speed by a factor of three is all that can be expected from technology alone. How then is a further factor of ten to be achieved?

2.2 Architecture

The most promising prospect is to use a number of processors simultaneously, that is multiprocessing. There are already examples of multiprocessors in general purpose and powerful scientific computers, e.g. IBM 308X and Cray X-MP. These, however, are only available with up to four processors and are most easily used when each processor handles an independent stream of tasks. In cases where all the power has to be applied to a single problem it is the user's responsibility to split the process into sub-processes. The user must also ensure that the sub-processes are properly synchronised and that boundary problems between the sub-processes are handled. Given that eight or sixteen processors are likely to be required to provide the necessary power, these problems could taken on daunting proportions.

2.3 Alternatives

It has been proposed that machines should be built to match meteorological models rather than vice versa. An example is the Distributed Array Processor from ICL in which it is proposed that there should be one processor per grid point. This architecture is very satisfactory for purely dynamical models but it is difficult to achieve good performance overall and the structure of the model is constrained for the life of the machine. There are also suggestions that arrays of Transputers might be developed into general purpose super computers. These ideas are at an early stage of development and may not be relevant on the timescale of 1990. The multiple super computer appears more likely to meet the needs of meteorology in the 1990s.

2.4 Technological progress

Over the past 25 years there has generally been a factor of thirty improvement in the processing capability of the most powerful scientific computers each decade. The cost of the most powerful computers at any time has been more or less constant. If this trend continues a machine of 30 to 100 GFLOPS should be available in the early 1990s for around $20M.

3. MAKING EFFECTIVE USE OF MULTIPROCESSORS

3.1 The workload profile

Through the 1980s and 1990s the Meteorological Office will be running models concerned with short period forecasting, daily forecasting and global circulation. There will be associated streams of analysis, assimilation and preparation of output. Some scope therefore exists to allocate independent processes to individual processors and achieve increased throughput. However, this will not be enough to achieve the overall peformance target and major models will have to make use of multiple processors.

3.2 The requirement

The hardware and software in a multiprocessor system should be capable of taking a program in a high level language, such as extended FORTRAN, and generating a set of linked sub-processes allowing for the following points:

- a variable number of processors (to allow for failures and upgrades)
- execution independent of particular processors
- synchronisation of sub-processes
- communication between sub-processes
- preallocation or dynamic allocation of sub-processes to processors.
- conflicts between sub-processes accessing shared memory.
- interactive optimisation for maximum performance

It may be reasonable to expect suitable features to exist in hardware available at the end of this decade but it is very doubtful whether software to deal adequately with the above problems will be developed within ten years. The commercial success of large multiprocessor systems in the supercomputer arena may well depend on the availability of systems that permit the user to achieve a high utilisation factor, over a number of processors, from a straightforward program.

3.3 Necessary hardware features

Assuming that a lot of the housekeeping necessary to apply multiple processors to a single task will be the responsibility of the user, there is a need for appropriate facilities in the hardware. In addition, the configuration must be balanced to permit maximum utilisation of the processors. It is likely that the following features will be needed:

- a mixture of very fast memory dedicated to processors and also fast shared memory.
- a mechanism for passing data directly between dedicated memory associated with any processors
- global synchronisation flags
- means to use processors interchangeably
- a control processor to optimise allocation of processes whose execution time depends on data.

3.4 Using multiprocessors

The Meteorological Office's plans for making use of multiprocessors are at an early stage of development as the requirement to do so, although inevitable, is not immediate. Preliminary thoughts on developing independent, relatively small processes and splitting larger ones into sub-processes are presented in succeeding paragraphs.

4. PRESENT AND FUTURE MODELLING ACTIVITIES

4.1 Large scale forecasting models

The current forecasting models consist of a global model with a resolution of $1^1/_2$° x $1^7/_8$° x 15 levels and a regional model covering the North Atlantic and most of Europe with a grid length of $^3/_4$° x $^{15}/_{16}$° x 15 levels. Both are grid point models and use a split explicit integration scheme with 4th order Lax Wendroff advection. A time step of 15 minutes is used in the global model and $7^1/_2$ minutes in the regional model. On the Cyber 205 the global model takes 4 minutes for each forecast day and the regional model 6 minutes. Both models obtain their initial data from a repeated insertion data assimilation scheme.

The forecast suite of programs has to be run to a strict schedule and, as this is governed largely by customer requirements outside the Meteorological Office, it is not likely to change greatly in the future. Consequently improvements in forecast models must be accommodated within the same time slots.

There is evidence that greater detail and more accurate short range predictions could be obtained from models having finer resolutions than are used at present, however it must be borne in mind that halving the grid length and doubling the number of levels will result in 16 times more computation. In order to provide more detailed initial data for such models it will be necessary to rely increasingly on satellite observations and on data obtained from other automatic observing systems. As more of these become available the calculations in the data assimilation scheme will increase substantially. More elaborate numerical techniques will probably be required to enable accurate calculations in the neighbourhood of internal discontinuities, such as fronts, to be made in the finer scale models.

4.2 Local short range forecasting

A non-hydrostatic meso-scale model is currently being tested for short range local weather forecasting. The area of integration covers the British Isles and the model has a 15 km grid length, 16 levels and a time step of 1 minute. It uses a semi-implicit leap frog finite difference scheme. If the model is introduced operationally it is thought that 12 or 18 hour forecasts might be produced every 3 hours, with detailed meso-scale analyses made every hour.

In order to represent some of the finer scale, but important, detail of the topography of the British Isles a 5 km grid length should be used rather than a 15 km one. Additional levels are also necessary to enable the boundary layer processes to be resolved more accurately. The size of the area of integration needs to be increased so that the lateral boundaries are further away from the region of interest. Such improvements would require a computer 30-50 times faster than the Cyber 205.

4.3 Climate studies

The 11-layer atmospheric general circulation model designed for climate studies normally uses a $2^1/_2°$ x $3^3/_4°$ global grid, though resolutions of 2° x 3° and 5° x $7^1/_2°$ are also used for some experiments. The model uses a leap frog explicit finite difference scheme and takes about 15 CPU hours on the Cyber 205 for a one year run. Several multi-annual cycle runs have been made for periods of up to 8 years. Work is currently in progress to couple the atmospheric general circulation model to a global ocean/sea-ice model. The ocean model will initially have the same resolution as the atmospheric model but an appreciably finer resolution is planned for the future.

The Meteorological Office is playing a full part in the World Climate Research Programme and it can be anticipated that research in this aspect of meteorology will expand over the next 20 years or so. Models of various degrees of complexity will be applied to the three streams of climate research (stream 1 - 1 month to a season, stream 2 - 1 to 5 years, stream 3 - decades to centuries). Finer mesh models with more accurate methods of representing physical processes will be needed as well as simulations for longer periods. More tests will be conducted on the likely effects of man's activities on both the global climate and on regional climates.

4.4 Research models

The Meteorological Office has a number of research projects that involve running large computer models in conjunction with field experiments or laboratory studies. The research is aimed at obtaining a better understanding of certain meteorological phenomena such as cumulonimbus convection, frontal dynamics, boundary layer turbulence and diffusion, airflow over orography, fog and stratocumulus, atmospheric chemistry, rotating fluid flow etc. Greater understanding of these phenomena can be expected to lead to improvements in the way they are represented in large scale numerical models of the atmosphere. It is anticipated that work in this area will need to be maintained in support of the more complex global and regional models being developed.

5. SOME SUGGESTIONS FOR IMPLEMENTING PARALLEL COMPUTATION

5.1 Vectorization

An important factor in the design of efficient code for the Cyber 205 is the organisation of the computation in terms of long vectors. To assist this, the data need to be arranged either as illustrated in Fig 1(b) or as in Fig 1(c) rather than as in Fig 1(a) which gives short vector lengths. The present climate model and mesoscale model use the vertical slice arrangement (Fig 1(b)) while the forecast model uses horizontal fields (Fig 1(c)) (which permits the longest vector lengths). We assume that the start-up time for vector operations on future generations of computers will be shorter than at present (so that there will be less advantage in using horizontal fields rather than vertical slices) but that short vectors will still need to be avoided.

5.2 Task sectioning

A simple way of splitting a forecast model up into parallel computation streams is to divide the region of integration up into sections as illustrated in Fig 2. Each processor deals with all the computation for a different sub-region. If the architecture of the computer is such that each processor has a large local memory, the bulk of the data could remain in place with new values overwriting old ones. Some overlaps will be necesary to allow the calculation to proceed at the edges of the sections and these would need to be copied at the start of each time step from the results in neighbouring sections.

A disadvantage of the technique is that the amount of computation may vary from one processor to another because different amounts of sub-grid scale physics will be involved in each section of the integration domain (for example, there is likely to be more convection in the tropics than at higher latitudes, radiation calculations are likely to be less extensive near the winter pole, there will be a greater preponderance of the relatively complex calculation of land surface processes in the Northern hemisphere). Some of the processors will then complete the calculations allocated to them earlier than others. Synchronisation checks will ensure that the processors remain in step as a whole, but there will be periods when some of them are idle.

5.3 Task pipelining

A suggested way of avoiding the problem outlined in the previous section is illustrated in Fig 3. We imagine that the calculation for the step $t = t_o + \Delta t$ is started in processor 1. As soon as it has completed the calculation for a sufficiently large number of vertical slices, processor 2 starts the computation for time step $t = t_o + 2\Delta t$. Processor 3 then does time step $t = t_o + 3\Delta t$, processor 4 does time step $t = t_o + 4\Delta t$ and eventually processor 1 does time step $t = t_o +$

5Δt and so on. With this way of organising the calculation all processors do computations for the entire global atmosphere, but for different time steps. Each processor will therefore be evenly balanced with its neighbours. In principle the system can run without synchronisation steps, provided that checks are made to ensure that results required by each processor have been completed by its predecessor.

Only explicit (or split-explicit) finite difference schemes can be used with the method, since both semi-implicit schemes and spectral models require the calculation for each time step to be complete before the next is started. The success of the technique depends on the speed that results from one processor can be made available for input to another processor.

The processors are acting in an analogous way to the elements of a vector pipeline with the computation being streamed through them. There will be a start-up time before all the processors are fully functioning, but subsequently one complete time step will be produced from N processors in 1/N times the time taken for one processor to do the calculation on its own.

5.4 Other considerations

Semi-implicit methods (and spectral models) cannot use the task pipelining approach described above because the calculations at each grid point depend on values at the same time level at all other grid points. A simple method of implementing a semi-implicit model on a parallel processor is illustrated in Fig 4. This reverts to the task sectioning approach discussed above (with the same disadvantages) for the explicit part of the calculation while the second order partial differential equation that arises for the semi-implicit scheme is solved by decoupling the vertical modes and solving for them in separate processors.

A more general approach to the problem of multi-tasking is to avoid trying to adapt the program structure to the specific architecture of the computer but instead to split the program into a number of self contained tasks which are then placed in a "task bin". Each processor then selects a task from the top of the bin and starts the calculation. When it has finished one task it selects another from the task bin. The inevitable task dependencies that arise (for example if task A must be complete before task B starts) could be achieved by a networking procedure though in many cases a less rigid program structure might work more effectively. One could instead introduce task attributes such as "task buoyancy" (more buoyant tasks would tend to rise to the top of the bin more rapidly than less buoyant tasks), "task stringing" (to cope with data dependencies that require one task to follow another) and "task stratification" (to enable groups of tasks to be logically separated from other groups of tasks). In the context of meteorological models, examples of separate tasks might be the calculation of the adiabatic terms in the dynamical equations for each vertical slice, the calculation of changes in humidity and temperature due to convection, the calculation of surface fluxes etc. Longer (coarse grain) tasks might be given greater buoyancy so that the shorter (fine grain) tasks can fill in the gaps that might otherwise give rise to processor idle time at the end of each task stratum. Task stratification could be useful, for example, in the semi implicit scheme to ensure that the explicit part of the calculation is complete before the Helmoltz equations are solved. An example of the use of task stringing might be in the solution of second order partial differential equations by sequential over relaxation where the result obtained for one grid point is used in the calculation of the next. The development of a general scheme of multi-tasking of this sort would probably be beyond the scope of an individual user but it could form the basis of a vendor supplied multi-tasking operating system.

6. CONCLUSIONS

The computer architecture most likely to satisfy the Meteorological Office's need for at least thirty times more computing power in the early 1990's is one with a relatively small number (8 or 16) state-of-the-art parallel processors and a very large ($\sim$500 Mwords) common memory. There does not seem to be any intrinsic difficulty in adapting atmosphere or ocean models for such a computer and the necessary program reorganisation appears to be particularly simple for explicit or split-explicit grid point models.

7. ACKNOWLEDGEMENT

The authors would like to thank Dr A Dickinson for providing many of the ideas in section 5.

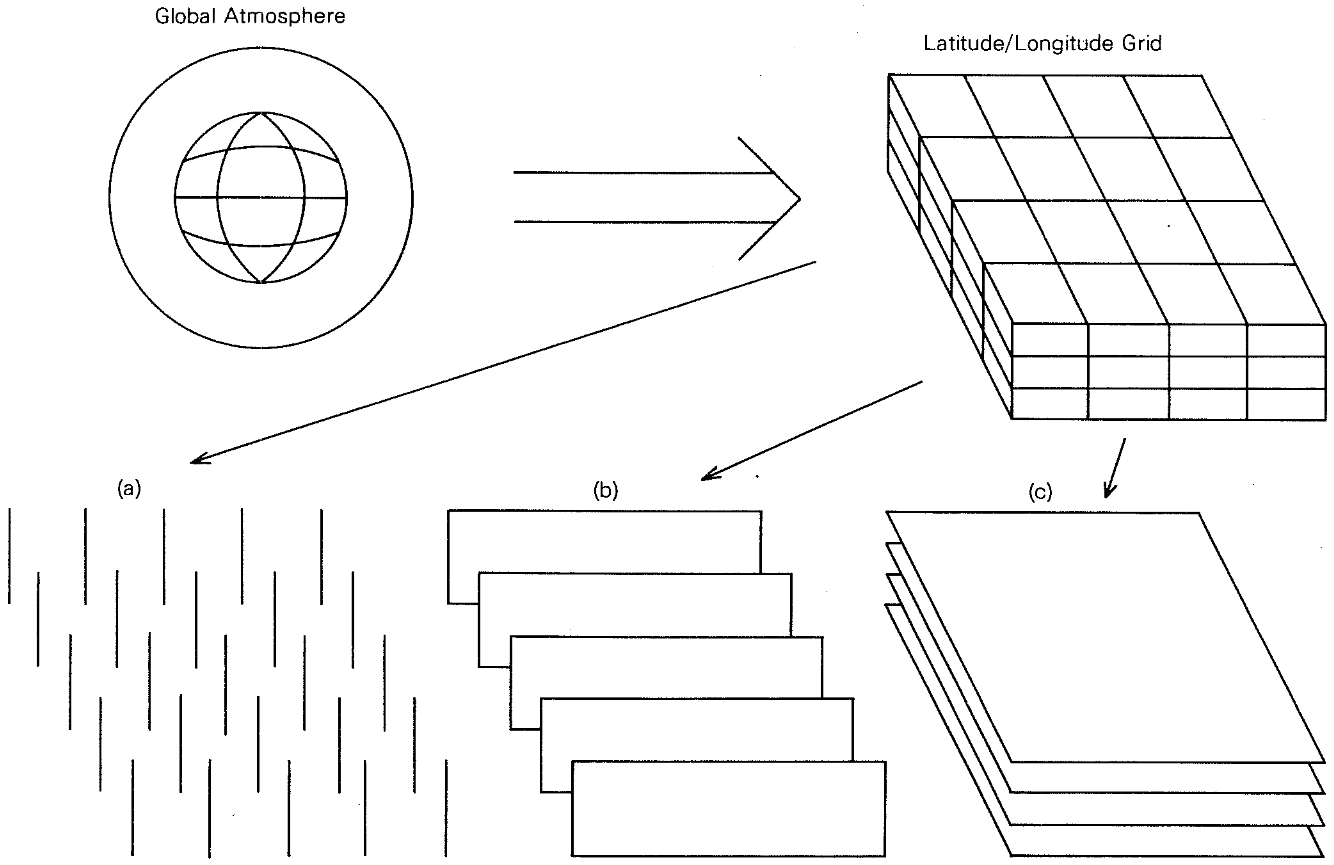

Figure 1 Different ways of organising data for vector computation
(a) vertical columns, (b) vertical slices, (c) horizontal fields

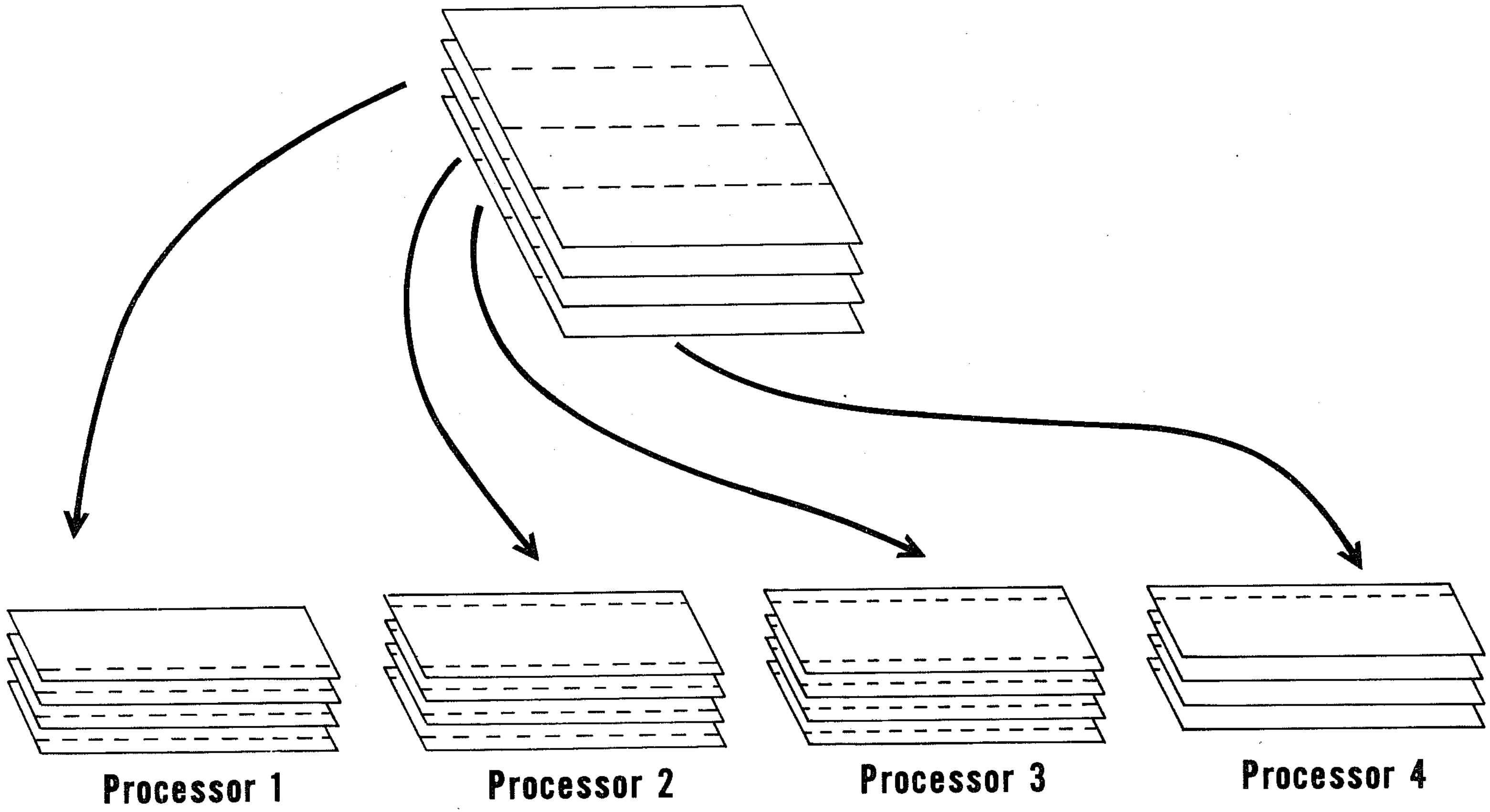

Figure 2 Task sectioning

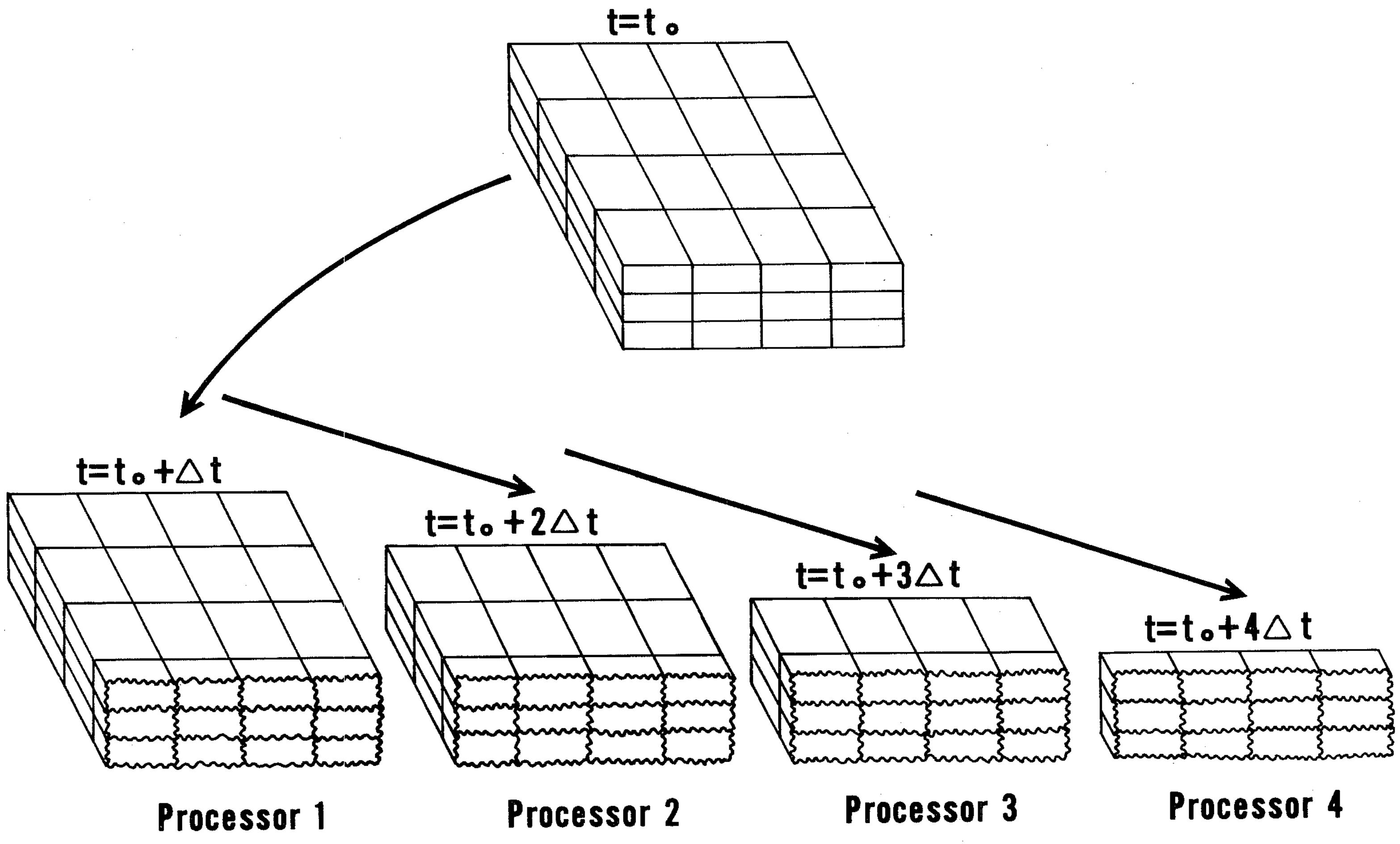

Figure 3 Task pipelining

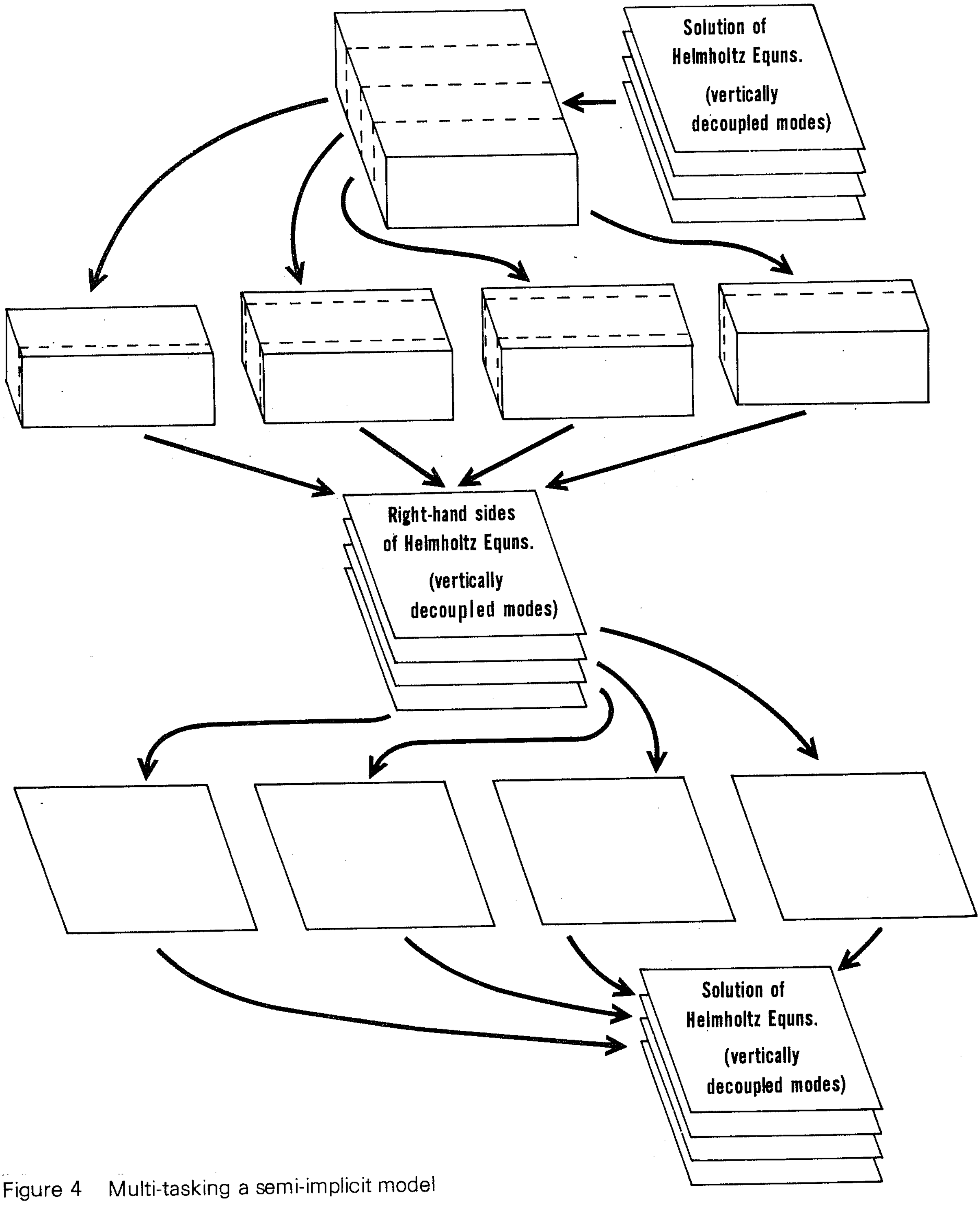

Figure 4 Multi-tasking a semi-implicit model