Met O (APR) Turbulence and Diffusion Note No. 209

Investigation of a New Finite Difference Diffusion Scheme

by

J M Hobson

17th November 1993

Met O (APR)
(Atmospheric Processes Research)
Meteorological Office
London Road
Bracknell
Berks, RG12 2SZ

Note

This paper has not been published. Permission to quote from it should be obtained from the Assistant Director, Atmospheric Processes Research Division, Met O (APR), Meteorological Office, London Road, Bracknell, Berkshire, RG12 2SZ.

Investigation of a New Finite Difference Diffusion Scheme

J M Hobson

17th November 1993

1 Introduction

Diffusion is an important process in virtually all numerical models of atmospheric flow. These models often use a finite grid on which the diffusive terms have to be evaluated. A variety of numerical methods can be used to perform this evaluation. These methods range from a computationally cheap explicit forward time-step scheme to a very computationally expensive fully implicit scheme. The disadvantage of the explicit scheme is that it requires a very small time-step in order to ensure convergence of the solution so that the run will be stable. The implicit scheme imposes no such limit on the time-step, although a limit may be imposed by other factors. Clearly a compromise between having to run with a small, but quick time-step and a run with the possibility of having a larger time-step, but with high computation time would be of some benefit. Attempts at this compromise in the past have always had their own disadvantages. Here, some of the schemes used in the past are reviewed, then a new scheme suggested by Mason (priv. com) is described. The implementation of the scheme into the large-eddy simulation code (Derbyshire 1991) currently used in Met O (APR) and the results obtained are then discussed.

2 Background

Various finite difference schemes exist which can be used to solve the diffusion equation. A very simple one-dimensional model was written to investigate these schemes. Only schemes that are accurate to the 2nd order in ΔZ are considered. The model solves the simple one-dimensional diffusion equation:

$$\frac{\partial U}{\partial t} = \nu \frac{\partial^2 U}{\partial Z^2} \tag{1}$$

where U is the horizontal velocity of the fluid, t is time, ν is the kinematic viscosity and Z is the vertical height.

The model simulates the purely diffusive behaviour of the flow of a fluid with constant kinematic viscosity, ν, between two infinite plates, separated by a distance H. Both are initially moving at constant velocity U_H, then one of the plates is instantaneously stopped. Only diffusive terms are evaluated; advection, Coriolis and pressure gradient terms are

not considered. The imposed boundary conditions are that there is no slip at either the moving, or the stationary plate. The model has ten vertical points between the plates and points above the upper plate and below the lower plate in order to impose the boundary conditions. All points are equally separated by distance ΔZ. Various numerical methods were used to solve the diffusion equation (1).

1. First, the method currently used in the Met O (APR) large eddy model is discussed. It is a forward-in-time, explicit scheme. In finite difference form it is written as

$$\frac{U_k^{t+1} - U_k^{t-1}}{2\Delta t} = \frac{\nu}{\Delta Z}\left(\frac{U_{k+1}^{t-1} - U_k^{t-1}}{\Delta Z} - \frac{U_k^{t-1} - U_{k-1}^{t-1}}{\Delta Z}\right) \tag{2}$$

   where k refers to any one vertical level. U_k^t represents U on level k at time t, so similarly U_{k+1}^{t-1} represents U on level $k+1$ at time $t - \Delta t$.

   Here U is taken at $t - \Delta t$ in order to calculate U at $t + \Delta t$. A linear stability analysis shows that this method is *conditionally stable* on $\Delta t \leq \mathrm{T}_d$, where $\mathrm{T}_d = \frac{\Delta Z^2}{4\nu}$. For details of this analysis see Appendix A. When this limit is more restrictive than the limit imposed by the advection scheme the run is said to be viscosity-limited. When the advection scheme limits the time-step the run is velocity-limited.

2. Secondly, an explicit, centred in time scheme was considered where values for U at time t are used to calculate U at time $t + \Delta t$. For this scheme, Equation (1) is written in its finite difference form as

$$\frac{U_k^{t+1} - U_k^{t-1}}{2\Delta t} = \frac{\nu}{\Delta Z}\left(\frac{U_{k+1}^{t} - U_k^{t}}{\Delta Z} - \frac{U_k^{t} - U_{k-1}^{t}}{\Delta Z}\right) \tag{3}$$

   This scheme is very unstable. Figure 1 shows U profiles for this scheme at a) $t = \mathrm{T}_d$; b) $t = 2\mathrm{T}_d$; c) $t = 3\mathrm{T}_d$ and d) $t = 4\mathrm{T}_d$. A time-step of $\frac{1}{4}\mathrm{T}_d$ was used in the run. The run is 'blowing up'. The same result is obtained whatever time-step is used. This can also be shown analytically. The run is therefore, *unconditionally unstable.*

3. The third method considered was that devised by DuFort and Frankel (1953). This can be written as

$$\frac{U_k^{t+1} - U_k^{t-1}}{2\Delta t} = \frac{\nu}{\Delta Z}\left(\frac{U_{k+1}^{t} - \frac{1}{2}(U_k^{t+1} + U_k^{t-1})}{\Delta Z} - \frac{\frac{1}{2}(U_k^{t+1} + U_k^{t-1}) - U_{k-1}^{t}}{\Delta Z}\right) \tag{4}$$

   This is *unconditionally stable*, but can produce spurious oscillatory modes. Figure 2a shows the U profile calculated using this scheme. A time-step of $4\mathrm{T}_d$ has been used and the run is at t=$2\mathrm{T}_D$, where $\mathrm{T}_D = \frac{H^2}{4\nu}$. T_D has been chosen as an appropriate time-scale for diffusion over the domain depth H. Figure 2b shows the same run one time-step later. Figures 2c and 2d show the run 100 time-steps further on than 2a and 2b respectively. Note the large oscillations about the mean profile, but unlike Figure 1 the oscillations are not growing. These oscillations are a well known consequence of this scheme.

4. The fourth method considered was the fully implicit scheme. The finite difference form is

$$\frac{U_k^{t+1} - U_k^{t-1}}{2\Delta t} = \frac{\nu}{\Delta Z}\left(\frac{U_{k+1}^{t+1} - U_k^{t+1}}{\Delta Z} - \frac{U_k^{t+1} - U_{k-1}^{t+1}}{\Delta Z}\right) \tag{5}$$

Here U_{k+1}^{t+1} and U_{k-1}^{t+1} are unknown and U_k^{t+1} is the quantity required from the calculation. If the equation is re-written for all levels in the model, a tridiagonal *kkp* x *kkp* matrix equation, where *kkp* is the number of vertical levels in the model, can be written and solved. This is ***unconditionally stable*** and does not produce the unwanted oscillations, but is computationally very expensive. In the simple one-dimensional model this scheme used about 2½ times as much CPU time as the forward-in-time explicit scheme. Figure 3 shows a U profile at $t = 2\mathrm{T}_D$ for a run with a time-step of $4\mathrm{T}_d$. It should be noted that in most models, diffusion is not the only process that dictates the maximum size of the time-step, so beyond a certain point (which will be the time-step imposed by the next most limiting process), eliminating the restriction from the diffusion scheme becomes irrelevant. However, in this purely diffusive model any time-step can be used with this scheme without affecting the stability or the results (except for the influence the time-step has on truncation errors).

The above is only a selection of schemes that have been written to solve the diffusion equation, but illustrate the problems involved. In the next section the motivation behind the attempt to overcome these problems is described.

3 The Mason Scheme

Any information takes a finite time to diffuse a certain distance. In the explicit diffusion scheme currently being used in the large-eddy model (the scheme described by equation (2)), any level, k is only influenced in any one time-step by the $k+1$ and $k-1$ levels. This imposes a limitation on the time-step, such that Δt must be less than or equal to the time that any information takes to diffuse from one level to the next. By contrast, the fully implicit scheme allows every level to be influenced by every other level, every time-step and therefore, presents no limitation on the time-step. The Mason scheme can be described as a mixture of both these methods, in that the calculation for the k^{th} level uses implicit values for terms at the $k+1$ and $k-1$ levels, but explicit values for terms at the $k+2$ and $k-2$ levels. This allows information that is two levels away from the k^{th} to have an influence on it and hence should allow the time-step limitation to be somewhat relaxed. In fact, the stability analysis (see appendix A) shows that this scheme is ***unconditionally stable.*** This is discussed below in more detail.

For every level (e.g. the k^{th}) the following three equations can be written. As described above, the terms at $k+2$ and $k-2$ (labelled $(\mathcal{A})$ and $(\mathcal{B})$ below) are evaluated at time $t-1$. All other terms are at time $t+1$.

$$\frac{U_{k+1}^{t+1} - U_{k+1}^{t-1}}{2\Delta t} = \frac{\nu}{\Delta Z}\left(\overbrace{\frac{(U_{k+2}^{t-1} - U_{k+1}^{t+1})}{\Delta Z}}^{(\mathcal{A})} - \frac{(U_{k+1}^{t+1} - U_{k}^{t+1})}{\Delta Z}\right) \tag{6}$$

$$\frac{U_{k}^{t+1} - U_{k}^{t-1}}{2\Delta t} = \frac{\nu}{\Delta Z}\left(\frac{(U_{k+1}^{t+1} - U_{k}^{t+1})}{\Delta Z} - \frac{(U_{k}^{t+1} - U_{k-1}^{t+1})}{\Delta Z}\right) \tag{7}$$

$$\frac{U_{k-1}^{t+1} - U_{k-1}^{t-1}}{2\Delta t} = \frac{\nu}{\Delta Z}\left(\frac{(U_k^{t+1} - U_{k-1}^{t+1})}{\Delta Z} - \underbrace{\frac{(U_{k-1}^{t+1} - U_{k-2}^{t-1})}{\Delta Z}}_{(\mathcal{B})}\right) \tag{8}$$

A 3x3 matrix can then be written for every k level and solved for U_k^{t+1}. It is computationally a little more expensive than explicit methods, but is very much cheaper than the fully implicit scheme. Also, the scheme appears not to support the spurious oscillatory modes that the scheme by DuFort and Frankel exhibits. As noted above, the stability analysis suggests that the scheme is unconditionally stable, when the physical arguments might suggest that it should be conditionally stable on $\Delta t \leq \frac{(2\Delta Z)^2}{4\nu} = \frac{Z^2}{\nu}$. The scheme therefore, appears to give us something for nothing! Figure 4 shows U profiles at a) $t = \frac{1}{2}\mathrm{T}_D$; b) $t = \mathrm{T}_D$; c) $t = 1\frac{1}{2}\mathrm{T}_D$ and d) $t = 2\mathrm{T}_D$. The solid lines are results from the Mason scheme, the dashed are the analytical result (Batchelor 1967, Chapter 4, Section 3). A time step of T_d has been used in the run. Note that the curves are smooth with no oscillatory modes, but that the Mason scheme appears to 'run slow', i.e. there is a time lag between the Mason scheme and the analytical solution it is trying to reproduce. This is not due to discretization errors as it does not occur for either forward-in-time explicit or fully implicit schemes with the same grid and time-step. The reason for the lag is worth further investigation. Assuming constant ΔZ and ν the Mason scheme can be re-written as

$$(2C^2 + 4C + 1)\left(\frac{U_k^{t+1} - U_k^{t-1}}{2\Delta t}\right) =$$

$$\nu\left(\underbrace{\frac{U_{k+1}^{t-1} - 2U_k^{t-1} + U_{k-1}^{t-1}}{\Delta Z^2}}_{(\mathcal{C})} + 4C\underbrace{\frac{(U_{k+2}^{t-1} - 2U_k^{t-1} + U_{k-2}^{t-1})}{(2\Delta Z)^2}}_{(\mathcal{D})}\right) \tag{9}$$

where $C = \frac{2\nu\Delta t}{\Delta Z^2}$ The right hand side of equation (9) can be divided into two halves. The first half, labelled $(\mathcal{C})$ is just the simple forward-in-time explicit scheme as shown in equation (2). The second half, labelled $(\mathcal{D})$ is the same scheme, but over two grid lengths, multiplied by a weighting factor. Symbolically, this approach can be represented by

$$\gamma\frac{\partial U}{\partial t} = \alpha\left[\frac{\partial U}{\partial t}\right]_{\Delta Z} + \beta\left[\frac{\partial U}{\partial t}\right]_{2\Delta Z} \tag{10}$$

where $\left[\frac{\partial U}{\partial t}\right]_{\Delta Z}$ represents the explicit scheme over distance ΔZ and $\left[\frac{\partial U}{\partial t}\right]_{2\Delta Z}$ represents the explicit scheme over distance $2\Delta Z$. Physically, the usual explicit scheme works well for small values of C so that, as discussed above, information can only travel one grid-space in one time-step. As C increases we require information from further away to influence U_k, so we would expect β to be proportional to C. For the Mason scheme

$$\alpha = 1 \quad , \quad \beta = 4C \quad \text{and} \quad \gamma = 2C^2 + 4C + 1.$$

As the two terms on the right hand side of equation (10) are complete diffusion schemes in their own right, we might expect that in order to get the correct amount of diffusion, we must impose $\alpha + \beta = \gamma$. Clearly this does not hold true in the Mason scheme, $\alpha + \beta < \gamma$,

so this scheme models less diffusion than it should, hence, there is a time lag. We might suggest using values for α, β and γ of

$$\alpha = 1 \quad , \quad \beta = 4C \quad \text{and} \quad \gamma = 4C + 1.$$

This leads to the diffusion scheme represented by

$$(4C+1)\left(\frac{U_k^{t+1} - U_k^{t-1}}{2\Delta t}\right) = \nu\left(\frac{U_{k+1}^{t-1} - 2U_k^{t-1} + U_{k-1}^{t-1}}{\Delta Z^2} + \frac{4C(U_{k+2}^{t-1} - 2U_k^{t-1} + U_{k-2}^{t-1})}{(2\Delta Z)^2}\right) \quad (11)$$

A stability analysis of this scheme leads to the requirement that

$$C \leq 2(1 - 1/8) \quad \text{or} \quad \Delta t \leq \frac{\Delta Z^2}{4\nu} \times 4(1 - 1/8)$$

which is similar in magnitude to requiring $\Delta t \leq \frac{(2\Delta Z)^2}{4\nu}$ that the physical argument would suggest. It is found that this scheme does not 'run slow'. However, when the boundary conditions were implemented naively, they put a much greater restriction on the time-step than the restriction from the scheme in the interior of the flow. Implementation of less restrictive boundary conditions was found to be non-trivial. This would make the use of such a scheme in a general model such as the large-eddy model difficult and would have uncertain consequences. The Mason scheme can be considered as being the same as this scheme, but with a time-step that is redefined, so that

$$\Delta t \longrightarrow \Delta t\left(\frac{4C+1}{2C^2+4C+1}\right) \quad (12)$$

If a linear stability analysis of the scheme with the modified time-step is performed, then stability requires

$$C < 2\left(\frac{2C^2+4C+1}{4C+1} - 1/8\right)$$

$$\text{i.e.} \qquad C < C\left(1 + \frac{24C+7}{4C(4C+1)}\right)$$

which is trivially satisfied for all $C > 0$.

From equation (12) above, the relative timing error of the Mason scheme is $\frac{2C^2}{2C^2+4C+1}$. For the type of applications that the Mason scheme will be used for and provided that C is kept small enough, the errors should be no more important than truncation errors. In non-linear models, it is probably sensible to run with $C = 1/2$. This would mean that the timing error would be $1/7$ or about 14%. If more accuracy is required then $C = 1/4$ gives a timing error of $1/17$, about 6%.

So far we have only considered the Mason scheme applied to a very simple linear model. We now look at its implementation in a rather more sophisticated model.

4 Adding the Mason Scheme to the LES

The LES model is much more complicated than the simple one-dimensional model discussed above, in a number of respects. Firstly, it is a three-dimensional model. Secondly, it models not only diffusion, but also other terms such as advection, Coriolis and pressure gradient. Thirdly, the model allows for a stretched vertical grid and finally it has a non-uniform, velocity dependent viscosity. Ignoring advection, Coriolis and pressure gradient terms, the diffusion equation (for U only) becomes -

$$\frac{\partial U}{\partial t} = \frac{\partial}{\partial x}\left(2\nu\frac{\partial U}{\partial x}\right) + \frac{\partial}{\partial y}\left(\nu\left[\frac{\partial U}{\partial y} + \overbrace{\frac{\partial V}{\partial x}}^{(\mathcal{A})}\right]\right) + \frac{\partial}{\partial z}\left(\nu\left[\overbrace{\frac{\partial U}{\partial z}}^{(\mathcal{B})} + \frac{\partial W}{\partial x}\right]\right) \tag{13}$$

Coding all these terms into a scheme of the type described above would be extremely complicated. The cross terms $(\mathcal{A})$ and $(\mathcal{B})$ in particular are very complicated to code in such a scheme. The resulting code may take more CPU time than it saves, though allowing a larger time-step. A comparison of the stability of each of the terms in the equation was performed in order to find which terms were most limiting to the stability, and hence the timestep, in model runs. In order to simplify the procedure, viscosity is assumed constant. In reality viscosity has strong gradients which may affect stability.

Term $(\mathcal{A})$ $\left(\frac{\partial}{\partial y}\left(\frac{\partial V}{\partial x}\right)\right)$ is analytically identical to $\frac{\partial}{\partial x}\left(\frac{\partial V}{\partial y}\right)$. It can be shown that this holds true even in finite difference terms. Similarly term $(\mathcal{B})$ $\left(\frac{\partial}{\partial z}\left(\frac{\partial W}{\partial x}\right)\right)$ can be re-written as $\frac{\partial}{\partial x}\left(\frac{\partial W}{\partial z}\right)$. Equation (13) can then be re-written as

$$\frac{\partial U}{\partial t} = \overbrace{2\nu\left(\frac{\partial^2 U}{\partial x^2}\right)}^{(\mathcal{C})} + \overbrace{\nu\left(\frac{\partial^2 U}{\partial y^2}\right)}^{(\mathcal{D})} + \overbrace{\nu\left(\frac{\partial^2 U}{\partial z^2}\right)}^{(\mathcal{E})} + \overbrace{\nu\frac{\partial}{\partial x}\left(\frac{\partial V}{\partial y} + \frac{\partial W}{\partial z}\right)}^{(\mathcal{F})} \tag{14}$$

The stability criterion for the explicit scheme currently used is that $\Delta t \leq \frac{\Delta^2}{4\nu}$, where Δ is a measure of the grid spacing. In many model runs Δx and Δy are very much larger than Δz. Therefore, term $(\mathcal{D})$ is very much more stable when using the explicit scheme than term $(\mathcal{E})$. Invoking mass continuity, $\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} = 0$, term $(\mathcal{F})$ becomes $-\nu\frac{\partial^2 U}{\partial x^2}$. Adding $(\mathcal{C})$ to $(\mathcal{F})$ in this linear analysis, now gives us $\nu\frac{\partial^2 U}{\partial x^2}$, which is similar in stability terms to $(\mathcal{D})$. Term $(\mathcal{E})$, therefore is the most restrictive to stability, so this term is evaluated using the Mason scheme. All other terms are evaluated explicitly as in the original model.

The Mason scheme is incorporated into the LES code in the source routines. These routines calculate the tendency (i.e. the change per unit time) of model variables for the current time-step. Firstly, in the section of code that deals with viscous effects the terms with second order vertical derivatives (such as term $(\mathcal{E})$ in equation (14)) are removed. Then a new section is added after all the other components such as advection, Coriolis etc. are calculated, but before the boundary conditions are dealt with. This section calculates the Mason vertical diffusion. The calculation is in the form of *kkp* 3x3 matrix equations. The elements in the matrix equation are first set up then the matrix is solved. This is done in a loop from $k = 3$ to $k = kkp - 2$. Levels $k = 2$, $k = kkp - 1$ and $k = kkp$ are dealt with separately, as they are affected by boundary conditions. The

model variables at level $k = 1$ are determined by the boundary conditions. The result of the calculation is the new value for the variable after the time-step , so a final loop calculates the tendency by subtracting the original value and dividing by the time-step. A copy of the code for the Mason scheme calculation for just U_k^{t+1} is given in Appendix B.

5 Testing the Mason Scheme

It was decided that a simple one-dimensional run with the LES code would be the best first test of the new code for de-bugging purposes. The model was actually run in three-dimensions with 10x10x40 grid points. Each column of model variables was calculated independently, but as all columns were initialized identically, they remained the same. Initially the model was run with a uniform grid in the vertical, again for de-bugging purposes. After satisfactory results were obtained, a vertically stretched grid was used. The model was given initial profiles for U and V (see Figure 5) and then it was run to a steady state. The original code was run as a control experiment. The mean profiles for U and V in the control experiment after 20 000 seconds of integration are shown in Figure 6. The mean profiles were produced by taking horizontal averages over the slices and then averaging over the previous 5 000 seconds. The code with the Mason scheme in place was then run, but with the same time-step criterion as the original model. The mean profiles for U and V are shown in Figure 7. The Mason scheme returned results that were the same to within 2% of the results returned from the run using the original scheme. The code could now be tested fully in three dimensions.

The test in three dimensions was again run with both schemes. The columns were not identical in this run. Small random perturbations are added to U and V at each grid-point at the start of each run to ensure they were different across the grid. The same random perturbations were used in all runs. A grid of 40x40x40 points was used. The grid was uniform in the horizontal directions, but was vertically stretched. The test was not designed to have any particular physical relevance, but the time-step used in the run had to be viscosity-limited with the original forward-in-time diffusion scheme. When the Mason scheme was used this limit was removed thus allowing the time-step to increase until it was limited by the time-step requirements of the advection scheme. The results in Figure 8 are from a run using the original diffusion scheme and the results in Figure 9 are with the Mason scheme. For completeness, the original scheme was tested with the time-step used in the Mason run. This run was unstable and failed in seconds. Note that the plots shown in Figures 8b and 9b show that that TBAR is considerably smoother with the Mason scheme. It is not clear why this should be, as in order to get this profile the three dimensional fields of TBAR have been averaged in both space and time. However, it may be that by calculating diffusion over more grid-spaces per time-step the Mason scheme gives smoother profiles. The parametrized turbulence (or 'Sub-grid') components are very similar in both runs. The sub-grid model within the large-eddy model has not been altered so any differences in these components are a consequence of feedback from differences in the resolved components. The resolved components are different, but have the same overall shape as each other.

Table 1 shows a comparison of run-time statistics for both the runs. The runs were

performed on a DEC Alpha machine (DEC 3000 Model 400 with 64Mb of memory).

Table 1

	Run with Original Scheme	Run with Mason Scheme
Model Time at end of run	20 000 s	20 000 s
Number of steps taken	98 800	10 500
Average time-step in run	0.2 s	1.9 s
Total CPU time taken	39 h	5¼ h
CPU time per step	1.43 s	1.82 s
CPU time per step per grid-point	2.23×10^{-5} s	2.84×10^{-5} s

Finally, a more realistic run was tested. A convective simulation was chosen which was viscosity-limited with the original explicit, forward-in-time diffusion scheme. The run had no mean wind and, as in the previous run, U and V were initialized with the same small random perturbations in all runs. All the previously mentioned runs have had zero surface heat-flux, in this simulation the heat-flux was allowed to be non-zero. The run had a horizontal resolution of 40x40 points and a vertical grid with 32 levels. The run with the Mason scheme calculated the time-step by requiring that the maximum value of C did not exceed 0.5. Figure 10 shows the results from the original diffusion scheme and figure 11 shows the same resulting quantities, but from a run using the Mason scheme. It can be seen again, that the results are quantitavely very similar. The time-series plot of energy in the model shows a very similar shape, with the model settling down at the same general energy level, but with small, short term variations.

The run-time statistics for these two runs are shown in Table 2.

Table 2

	Run with Original Scheme	Run with Mason Scheme
Model Time at end of run	20 000 s	20 000 s
Number of steps taken	97 500	42 600
Average time-step in run	0.2 s	0.46 s
Total CPU time taken	30.5 h	16 h
CPU time per step	1.13 s	1.34 s
CPU time per step per grid-point	2.21×10^{-5} s	2.62×10^{-5} s

6 Conclusions

We have looked at a few of the many possible ways of modelling diffusion in an atmospheric model. We have taken a close look at the new Mason scheme. This new scheme appears

to have avoided the problem of instability with increased time-step, although further investigation leads to the conclusion that the stability is retained only by a small effective reduction in the time-step. This has the effect of producing a time lag in the diffusion scheme with respect to model time. It was found that the time-step could be considerably increased with only small errors resulting from the time lag. The results from the tests we ran, suggest a time-step of order $\Delta t = \frac{\Delta Z^2}{4\nu}$ was satisfactory. This is the maximum limit for the time-step when using the explicit forward-in-time scheme. However, in non-linear models using the forward-in-time scheme the time-step would be limited by other possible instabilities in the non-linear terms and would, in fact, be very much smaller than the limit. We will, therefore, save considerable computational time by using the Mason scheme with the time-step limit of the explicit scheme. When running on a scalar CPU (a single processor that only performs one command at a time) the execution time taken for single time-step was about 20% to 30% longer than the forward-in-time explicit scheme for the same number of grid-points. The time-step can only be increased in runs that are limited by the viscous terms, and this increase is limited by the implied time lag. Even with all these restrictions it was seen that in some of the runs that were carried out, the execution-time was cut by as much as 50%.

Acknowledgements. I would like to thank Nigel Wood for all the help he gave me during the production of this document. I am also grateful to Fiona Hewer, Malcolm MacVean, Paul Mason and David Pick for their careful reviewing and helpful comments.

Appendix A

Stability Analysis for the Forward-in-Time Explicit Scheme

Re-writing Equation (2)

$$U_k^{t+1} = CU_{k+1}^{t-1} - 2CU_k^{t-1} + CU_{k-1}^{t-1} + U_k^{t-1}$$

where $C = \frac{2\nu\Delta t}{\Delta Z^2}$

Following the von Neumann stability analysis procedure (see Roache 1972, Chapter 3, Section A-5-b) a Fourier series expansion of the solution to the above equation can be written

$$U_k^t = A\lambda^{n\Delta t} e^{ikm\Delta Z}$$

where $A\lambda^{n\Delta t}$ is the amplitude function of the time-level n of a particular component whose wave number is k. The displacement in time and vertical distance are denoted by the integers n and m respectively. Substituting this expression into the finite difference equation leads to

$$\begin{aligned} \lambda^2 &= C\left(e^{ik\Delta Z} + e^{-ik\Delta Z}\right) - 2C + 1 \\ &= 2C\cos(k\Delta Z) - 2C + 1 \\ &= 2CX - 2C + 1 \end{aligned}$$

where $X = \cos(k\Delta Z)$

Stability or instability is determined by the decay or amplification of the Fourier solution, therefore, stability requires that $|\lambda| \leq 1$ i.e. $|\lambda^2| \leq 1$. Which gives

$$-1 \leq 2CX - 2C + 1 \leq 1$$

The right hand inequality simplifies to $X \leq 1$ which is trivially true.

The left hand inequality reduces to $\frac{C-1}{C} \leq X$. The minimum value for X is -1 and substituting for C gives

$$2\nu\frac{\Delta t}{\Delta Z^2} - 1 \leq -2\nu\frac{\Delta t}{\Delta Z^2}$$

which leads to $\Delta t \leq \frac{\Delta Z^2}{4\nu}$.

The scheme is, therefore, conditionally stable on

$$\Delta t \leq \frac{\Delta Z^2}{4\nu}$$

Stability Analysis for Mason Scheme

Equations (6), (7) and (8) reduce to

$$U_k^{t+1} = \frac{1}{2C^2+4C+1}\left(C^2U_{k+2}^{t-1} + CU_{k+1}^{t-1} + (1+2C)U_k^{t-1} + CU_{k-1}^{t-1} + C^2U_{k-2}^{t-1}\right)$$

where $C = \frac{2\nu\Delta t}{\Delta Z^2}$

Following the procedure above, we substitute $U_k^t = A\lambda^{n\Delta t}e^{ikm\Delta Z}$

$$\begin{aligned}
\lambda^2 &= \frac{1}{2C^2+4C+1}\left(C^2\left(e^{2ik\Delta Z} + e^{-2ik\Delta Z}\right) + (1+2C) + C\left(e^{ik\Delta Z} + e^{-ik\Delta Z}\right)\right) \\
&= \frac{1}{2C^2+4C+1}\left(2C^2\cos(2k\Delta Z) + (1+2C) + 2C\cos(k\Delta Z)\right) \\
&= \frac{1}{2C^2+4C+1}\left(2C^2(2X^2-1) + (1+2C) + 2CX\right) \qquad \text{again, } X = \cos(k\Delta Z) \\
&= \frac{1}{2C^2+4C+1}\left(4C^2X^2 + 2CX + 1 + 2C + 2C^2\right)
\end{aligned}$$

Stability requires $|\lambda| \leq 1$ i.e. $|\lambda^2| \leq 1$. Which gives

$$-(2C^2+4C+1) \leq 4C^2X^2 + 2CX + 1 + 2C - 2C^2 \leq 2C^2 + 4C + 1$$

The left hand inequality simplifies to

$$0 \leq 4C^2X^2 + 2CX + 2 + 6C$$

$4C^2X^2 + 2CX + 2 + 6C$ has a minimum at $8C^2X = -2C$ i.e. $X = -\frac{1}{4C}$. Its minimum value is, therefore, $\frac{1}{4} - \frac{1}{2} + 2 + 6C = 6C + 1\frac{3}{4}$. So the left hand inequality is true as $C > 0$ always.

The right hand inequality reduces to

$$4C^2X^2 + 2CX - 2C - 4C^2 \leq 0$$

Factorizing leads to $(X-1)(4C^2(X+1) + 2C) \leq 0$. $(X-1) \leq 0$, which requires $(4C^2(X+1) + 2C) > 0$.This leads to $X > -\frac{1}{2C} - 1$ which, again, is true for all $C > 0$.

The scheme is, therefore, unconditionally stable.

Appendix B

```
C..                  ************
C..MASON SCHEME FOR * U-SOURCE *
C..                  ************
        DO 410 K=3,KKP-2
          DO 410 J=JMINP,JMAXP
C..CALC COMPONENTS OF MATRIX
            TMP11=(1.0/(2.0*DTM)) +
     &             0.5*(VIS(J,K+1)+VISP1(J,K+1))*ECZA(J,K+1) +
     &             0.5*(VIS(J,K)+VISP1(J,K))*ECZB(J,K+1)
            TMP12=-0.5*(VIS(J,K)+VISP1(J,K))*ECZB(J,K+1)
            TMP21=-0.5*(VIS(J,K)+VISP1(J,K))*ECZA(J,K)
            TMP22=(1.0/(2.0*DTM)) +
     &             0.5*(VIS(J,K)+VISP1(J,K))*ECZA(J,K) +
     &             0.5*(VIS(J,K-1)+VISP1(J,K-1))*ECZB(J,K)
            TMP23=-0.5*(VIS(J,K-1)+VISP1(J,K-1))*ECZB(J,K)
            TMP32=-0.5*(VIS(J,K-1)+VISP1(J,K-1))*ECZA(J,K-1)
            TMP33=(1.0/(2.0*DTM)) +
     &             0.5*(VIS(J,K-1)+VISP1(J,K-1))*ECZA(J,K-1) +
     &             0.5*(VIS(J,K-2)+VISP1(J,K-2))*ECZB(J,K-1)
            TMPA =(ZU(J,K+1)/(2.0*DTM)) +
     &               SU(J,K+1) +
     &          0.5*(VIS(J,K+1)+VISP1(J,K+1))*ECZA(J,K+1)*ZU(J,K+2)
            TMPB =(ZU(J,K)/(2.0*DTM)) +
     &               SU(J,K)
            TMPC =(ZU(J,K-1)/(2.0*DTM)) +
     &               SU(J,K-1) +
     &          0.5*(VIS(J,K-2)+VISP1(J,K-2))*ECZB(J,K-1)*ZU(J,K-2)
C..SOLVE MATRIX
            UVWTMP(J,K)=
     &            (TMP11*(TMPB*TMP33-TMP23*TMPC)-TMPA*TMP21*TMP33) /
     &             (TMP11*(TMP22*TMP33-TMP23*TMP32)-TMP12*TMP21*TMP33)
  410   CONTINUE
        K=2
        DO 411 J=JMINP,JMAXP
C..CALC COMPONENTS OF MATRIX
            TMP11=(1.0/(2.0*DTM)) +
     &             0.5*(VIS(J,K+1)+VISP1(J,K+1))*CZA(K+1) +
     &             0.5*(VIS(J,K)+VISP1(J,K))*CZB(K+1)
            TMP12=-0.5*(VIS(J,K)+VISP1(J,K))*CZB(K+1)
            TMP21=-0.5*(VIS(J,K)+VISP1(J,K))*CZA(K)
            TMP22=(1.0/(2.0*DTM)) +
     &             0.5*(VIS(J,K)+VISP1(J,K))*CZA(K) +
     &             (VIS(J,K-1)+VISP1(J,K-1))*CZB(K)
            TMP23=0.0
            TMP32=0.0
            TMP33=0.0
            TMPA =(ZU(J,K+1)/(2.0*DTM)) +
     &               SU(J,K+1) +
     &             0.5*(VIS(J,K+1)+VISP1(J,K+1))*CZA(K+1)*ZU(J,K+2)
            TMPB =(ZU(J,K)/(2.0*DTM)) +
     &               SU(J,K)
     &               -(VIS(J,K-1)+VISP1(J,K-1))*CZB(K)*UGAL
            TMPC =0.0
C..SOLVE MATRIX
            UVWTMP(J,K)=
     &            (TMP11*TMPB-TMP21*TMPA) /
```

```
     &              (TMP11*TMP22-TMP21*TMP12)
  411   CONTINUE
        K=KKP-1
        DO 412 J=JMINP,JMAXP
C..CALC COMPONENTS OF MATRIX
            TMP11=(1.0/(2.0*DTM)) +
     &                0.5*(VIS(J,K)+VISP1(J,K))*CZB(K+1)
            TMP12=-0.5*(VIS(J,K)+VISP1(J,K))*CZB(K+1)
            TMP21=-0.5*(VIS(J,K)+VISP1(J,K))*CZA(K)
            TMP22=(1.0/(2.0*DTM)) +
     &             0.5*(VIS(J,K)+VISP1(J,K))*CZA(K) +
     &             0.5*(VIS(J,K-1)+VISP1(J,K-1))*CZB(K)
            TMP23=-0.5*(VIS(J,K-1)+VISP1(J,K-1))*CZB(K)
            TMP32=-0.5*(VIS(J,K-1)+VISP1(J,K-1))*CZA(K-1)
            TMP33=(1.0/(2.0*DTM)) +
     &             0.5*(VIS(J,K-1)+VISP1(J,K-1))*CZA(K-1) +
     &             0.5*(VIS(J,K-2)+VISP1(J,K-2))*CZB(K-1)
            TMPA =(ZU(J,K+1)/(2.0*DTM)) +
     &                SU(J,K+1)
            TMPB =(ZU(J,K)/(2.0*DTM)) +
     &                SU(J,K)
            TMPC =(ZU(J,K-1)/(2.0*DTM)) +
     &                SU(J,K-1) +
     &            0.5*(VIS(J,K-2)+VISP1(J,K-2))*CZB(K-1)*ZU(J,K-2)
C..SOLVE MATRIX
            UVWTMP(J,K)=
     &           (TMP11*(TMPB*TMP33-TMP23*TMPC)-TMPA*TMP21*TMP33) /
     &            (TMP11*(TMP22*TMP33-TMP23*TMP32)-TMP12*TMP21*TMP33)
  412   CONTINUE
        K=KKP
        DO 413 J=1,JJP
C..CALC COMPONENTS OF MATRIX
            TMP11=0.0
            TMP12=0.0
            TMP21=0.0
            TMP22=(1.0/(2.0*DTM)) +
     &             0.5*(VIS(J,K-1)+VISP1(J,K-1))*CZB(K)
            TMP23=-0.5*(VIS(J,K-1)+VISP1(J,K-1))*CZB(K)
            TMP32=-0.5*(VIS(J,K-1)+VISP1(J,K-1))*CZA(K-1)
            TMP33=(1.0/(2.0*DTM)) +
     &             0.5*(VIS(J,K-1)+VISP1(J,K-1))*CZA(K-1) +
     &             0.5*(VIS(J,K-2)+VISP1(J,K-2))*CZB(K-1)
            TMPA =0.0
            TMPB =(ZU(J,K)/(2.0*DTM)) +
     &              SU(J,K)
            TMPC =(ZU(J,K-1)/(2.0*DTM)) +
     &              SU(J,K-1) +
     &            0.5*(VIS(J,K-2)+VISP1(J,K-2))*CZB(K-1)*ZU(J,K-2)
C..SOLVE MATRIX
            UVWTMP(J,K)=
     &           (TMPB*TMP33-TMPC*TMP23) /
     &            (TMP22*TMP33-TMP32*TMP23)
  413   CONTINUE

        DO 414 K=2,KKP
          DO 414 J=1,JJP
            SU(J,K)=(UVWTMP(J,K)-ZU(J,K))/(2.0*DTM)
  414   CONTINUE
```

References

Batchelor, G.K. 1967 *An Introduction to Fluid Dynamics.* Cambridge University Press, 615 pp.

Derbyshire, S.H. 1991 The Met. Office 3-D Cartesian Large Eddy Simulation model for the CRAY-YMP, *Unpublished.*

DuFort, E.C. and Frankel, S.P. 1953 Stability Conditions in the Numerical Treatment of Parabolic Differential Equations. *Math. Tables and Other Aids to Computation.*, **7**, 135-152.

Roache, P.J. 1972 *Computational Fluid Dynamics.* Hermosa Publ., 446 pp.

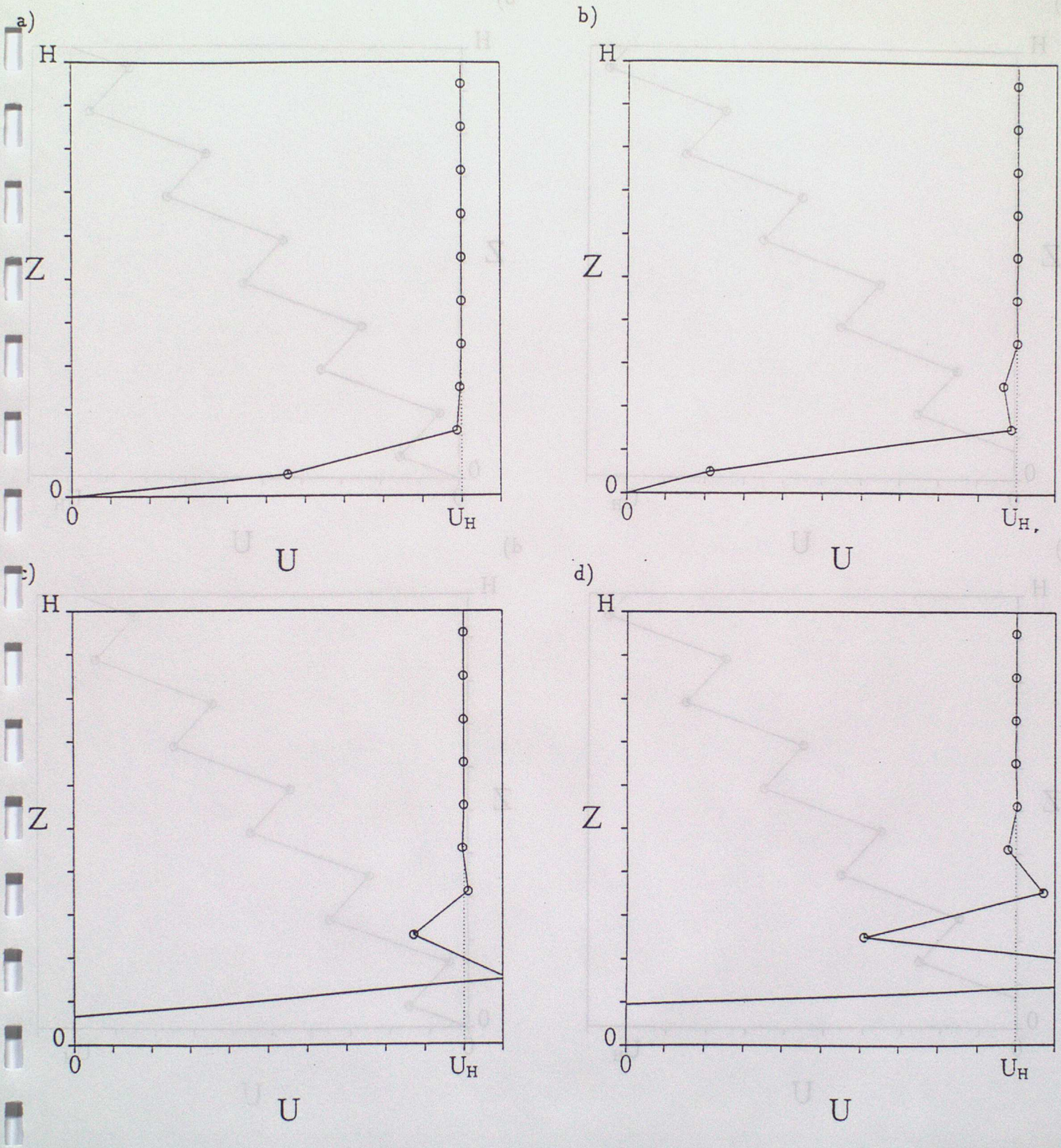

Figure 1: Profiles of velocity, U, from a simple one-dimensional constant viscosity model using an explicit, centred in time diffusion scheme. The profiles shown are at times a) $t = T_d$; b) $t = 2T_d$; c) $t = 3T_d$ and d) $t = 4T_d$. A time-step of $\frac{1}{4}T_d$ was used in the run.

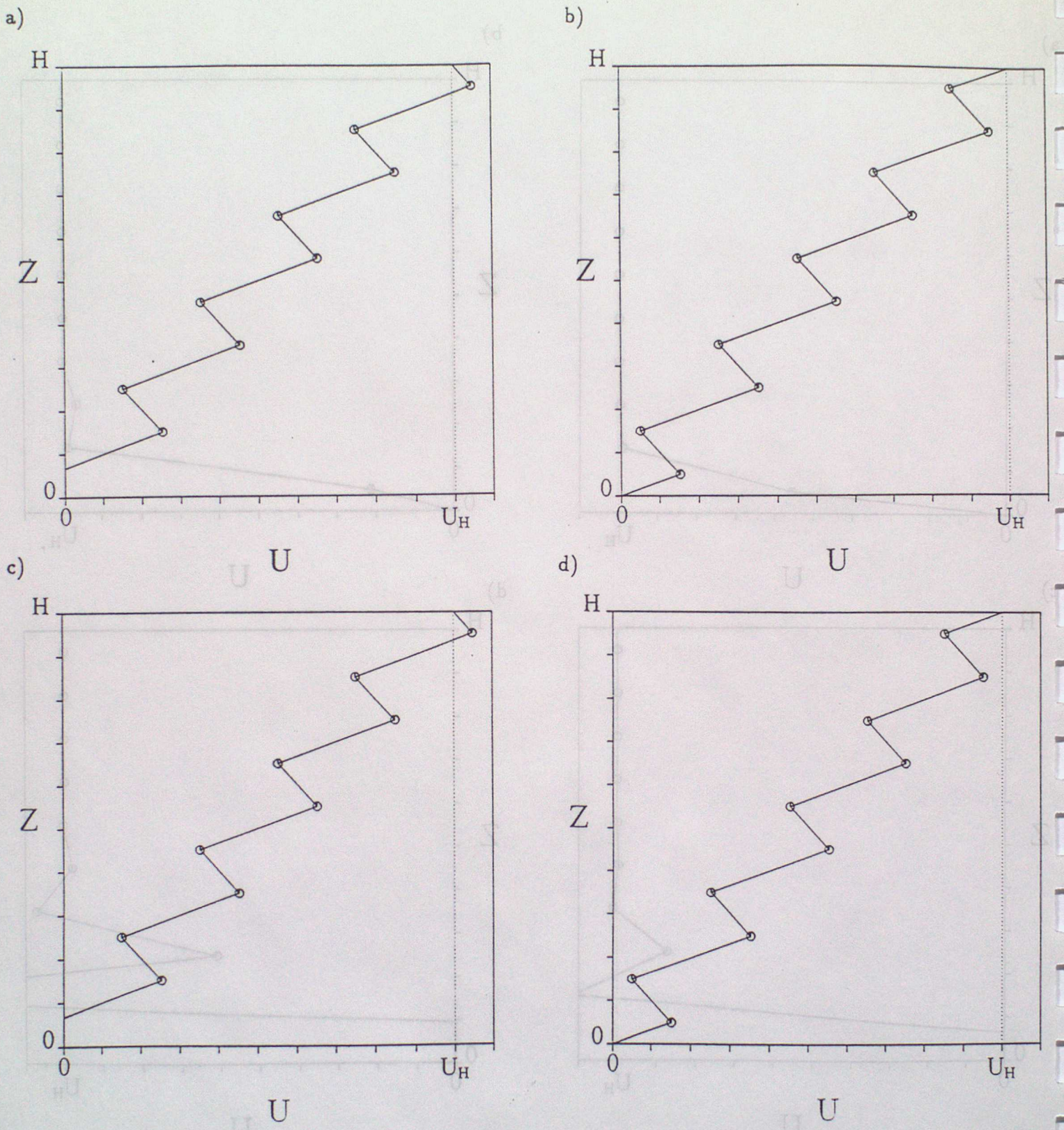

Figure 2: Profiles of velocity, U, from a simple one-dimensional constant viscosity model using the DuFort-Frankel diffusion scheme. A time-step of $4T_d$ was used in the run. The profile at $t=2T_D$ is shown in a). The profile one time-step on is shown in b). The profiles of the run 100 timesteps further on than a) and b) are shown in c) and d) respectively.

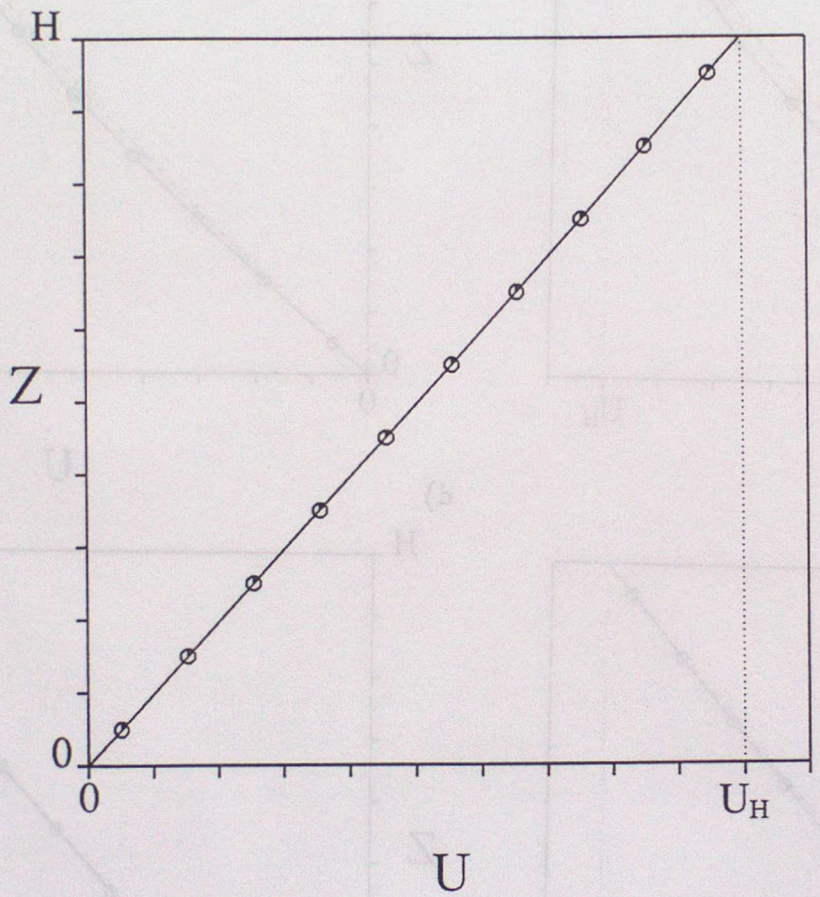

Figure 3: The profile of velocity, U, from a simple one-dimensional constant viscosity model using the fully implicit diffusion scheme at time $t = 2T_D$. A time-step of $4T_d$ was used in the run.

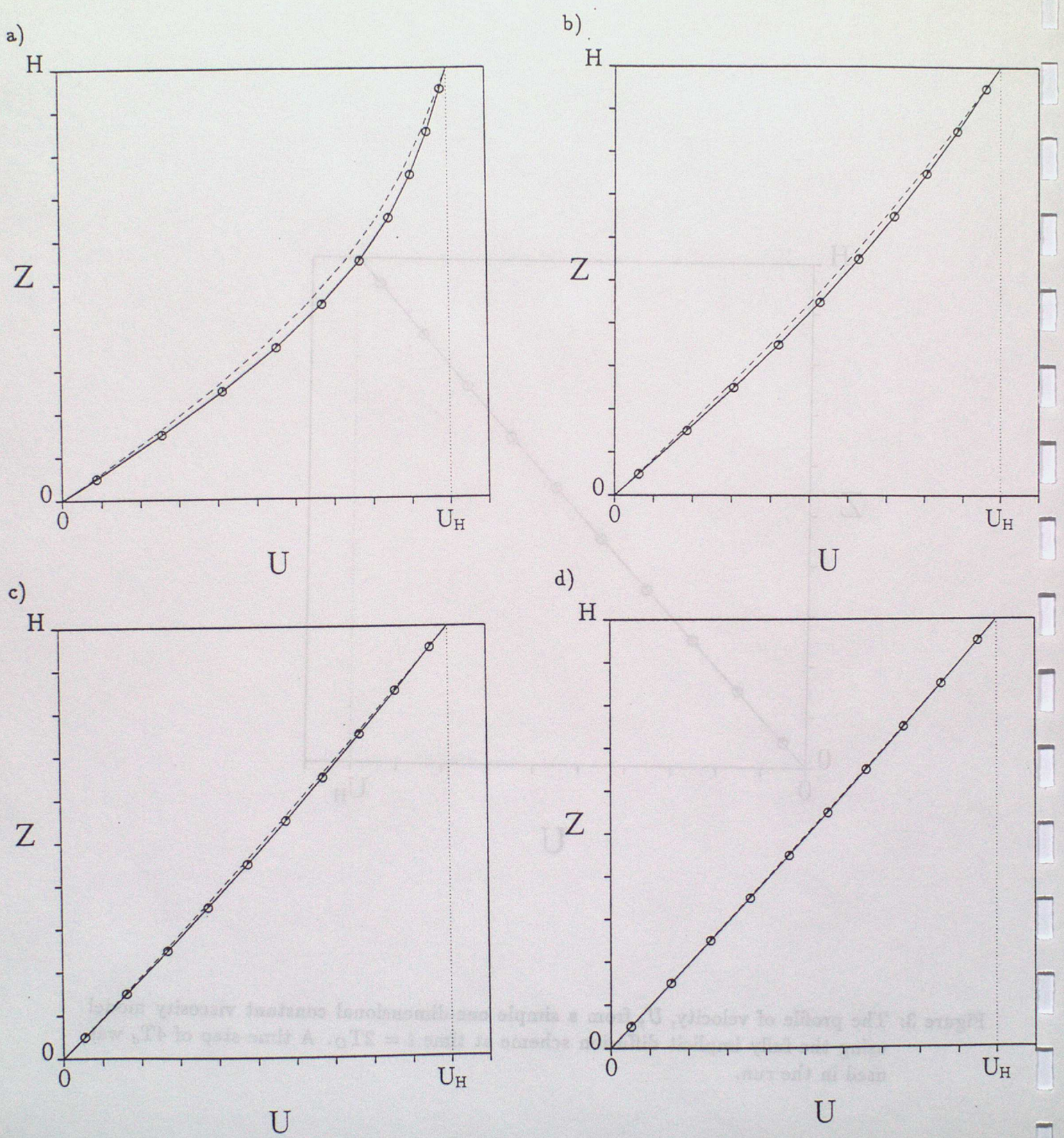

Figure 4: Solid lines are profiles of velocity, U, from a simple one-dimensional constant viscosity model using the Mason diffusion scheme. Dashed lines are the analyical solution to the same problem at the same model time. A time-step of $4T_d$ was used in the run. The profiles shown are at times a) $t = \frac{1}{2}T_D$; b) $t = T_D$; c) $t = 1\frac{1}{2}T_D$ and d) $t = 2T_D$.

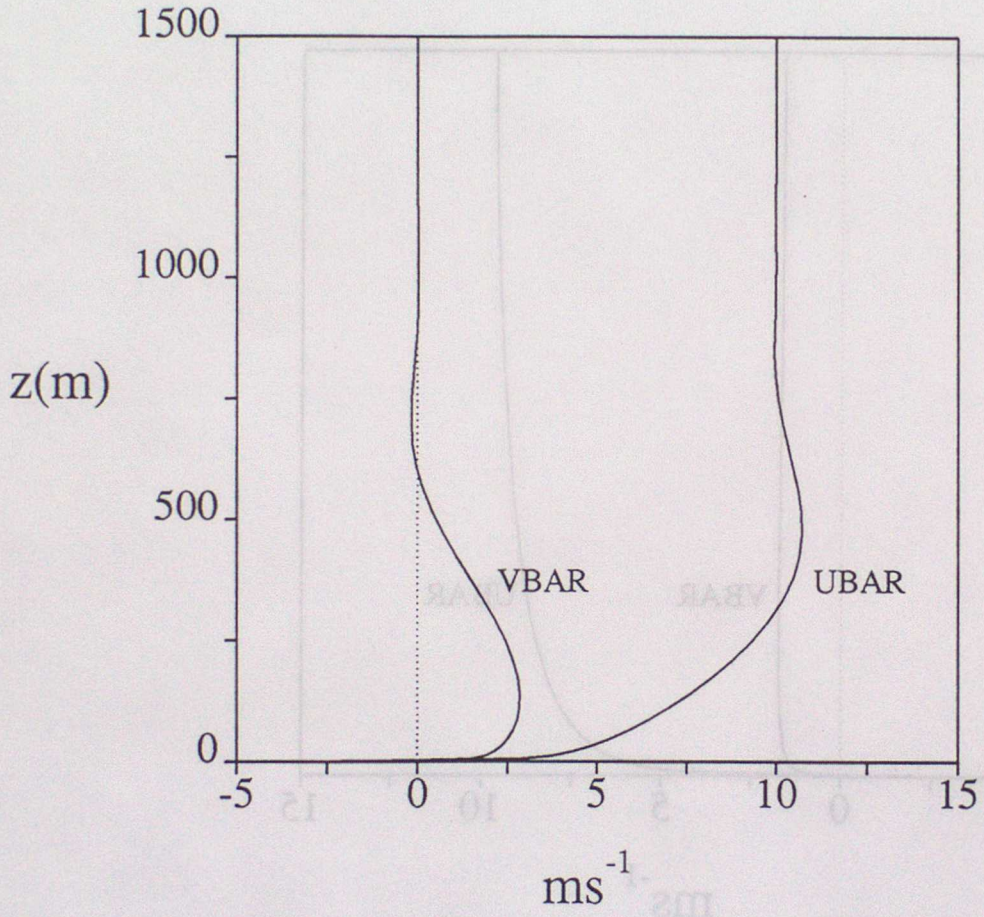

Figure 5: The initial vertical profiles for U and V used in the large-eddy model

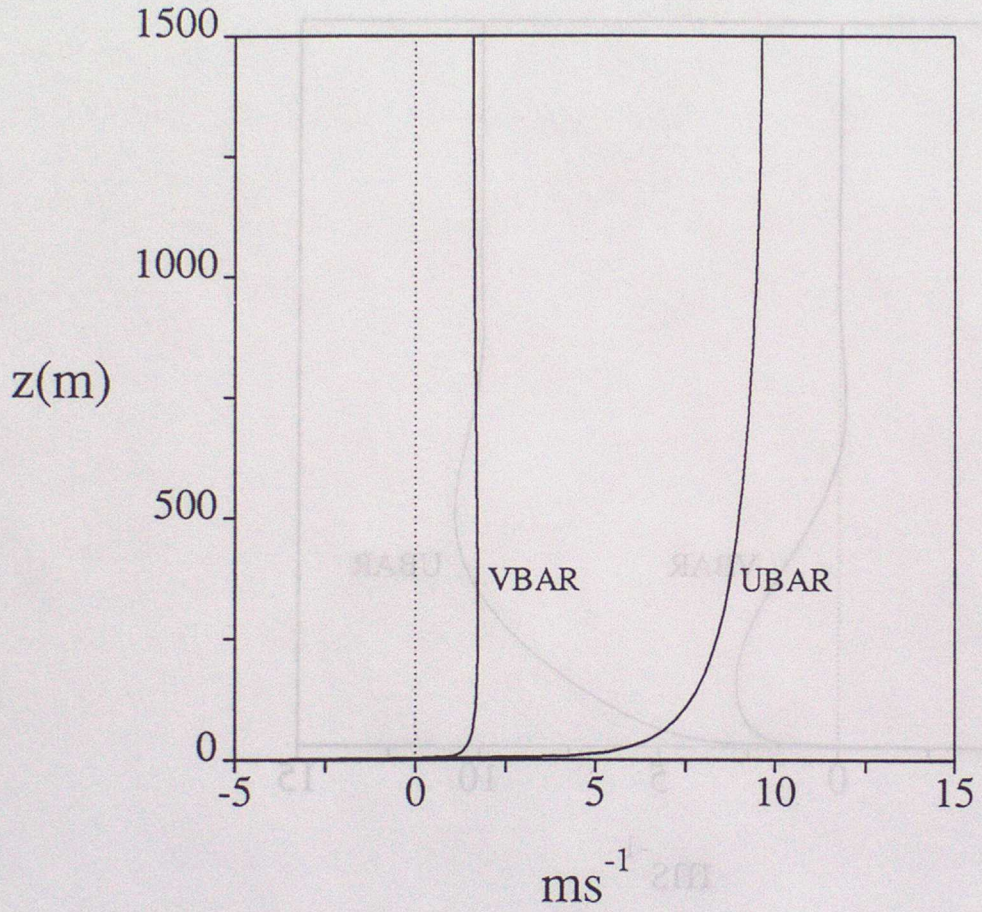

Figure 6: The verical profiles for U and V from a pseudo-one-dimensional run of large-eddy model using the original forward-in-time explicit scheme. The run is at $t = 20\ 000$ seconds

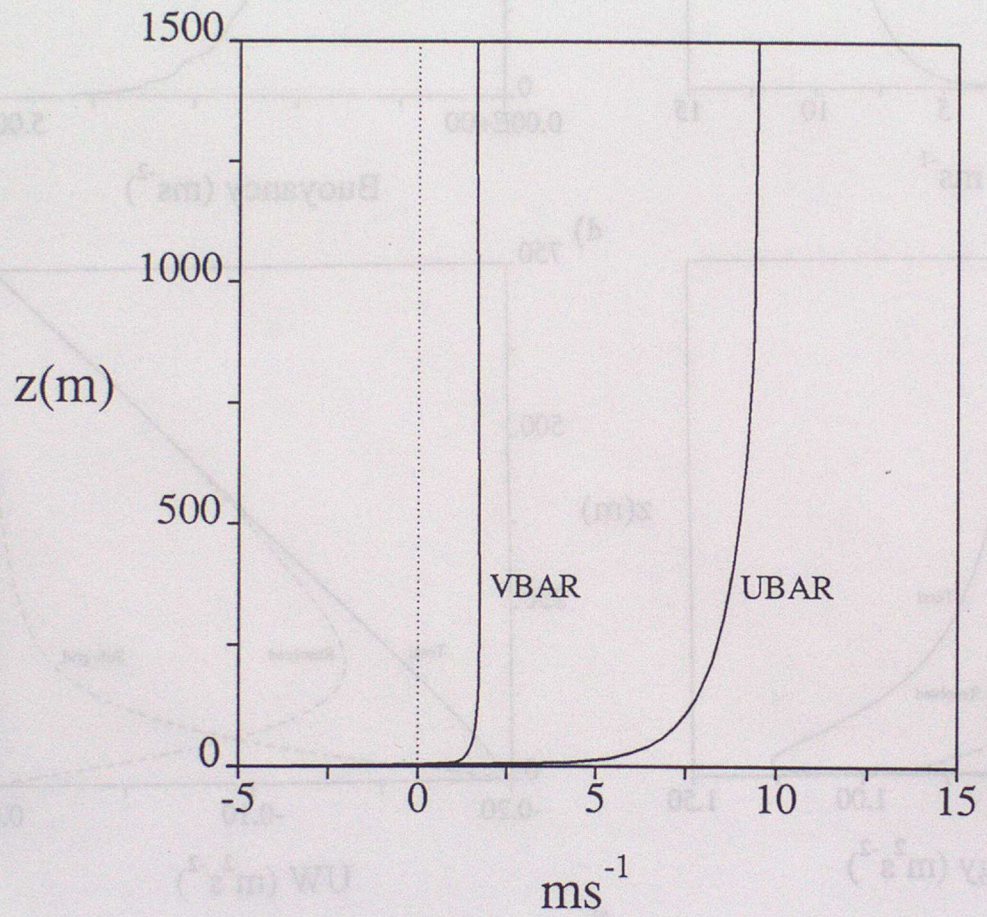

Figure 7: The vertical profiles for U and V from a pseudo-one-dimensional run of large-eddy model using the Mason scheme. The run is at $t = 20\ 000$ seconds

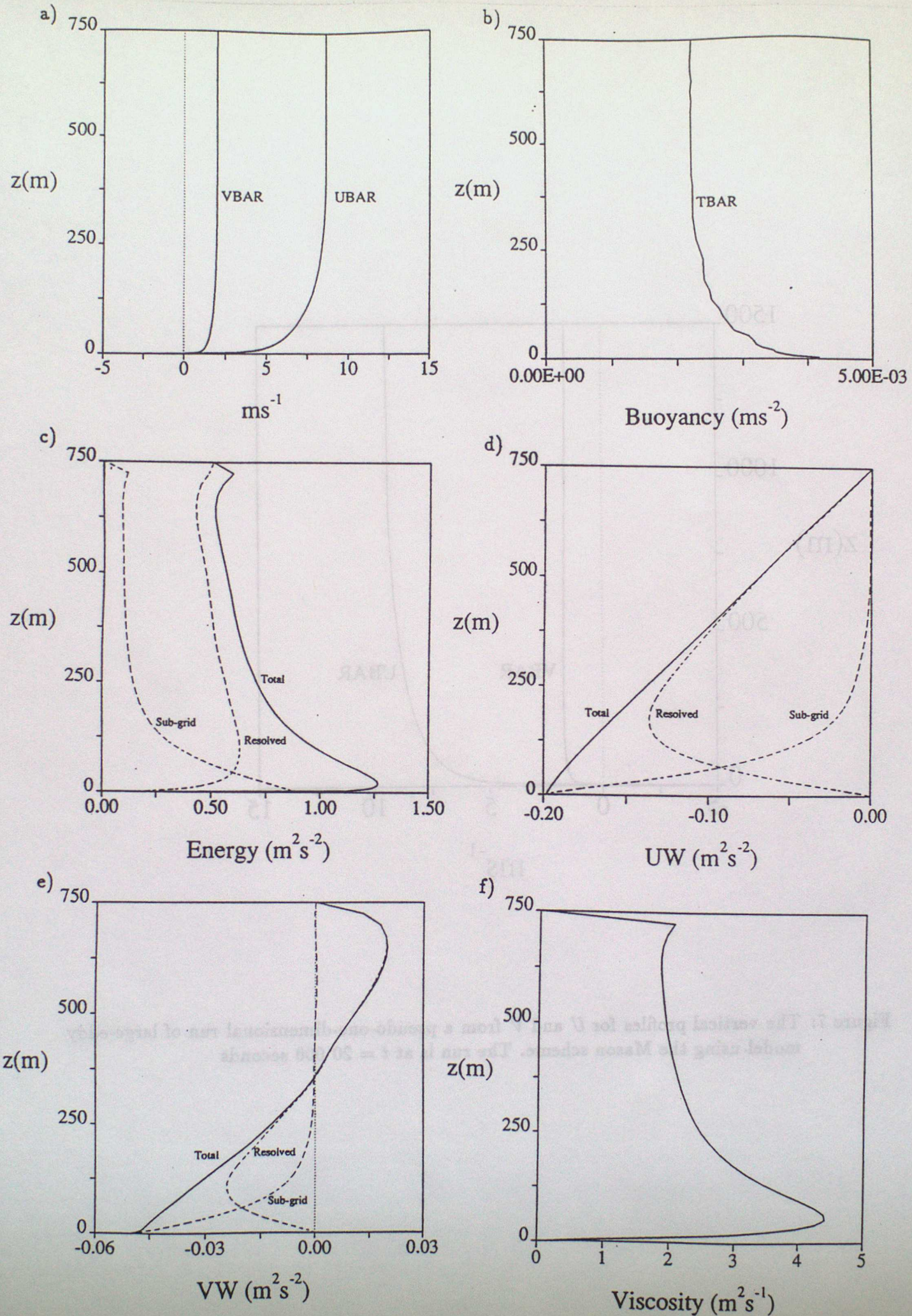

Figure 8: Results from a fully-three-dimensonal run of large-eddy model using the original forward-in-time explicit scheme. The run is at $t = 20\,000$ seconds. The following vertical profiles are shown – a) UBAR and VBAR; b) TBAR; c) Energy; d) UW; e)

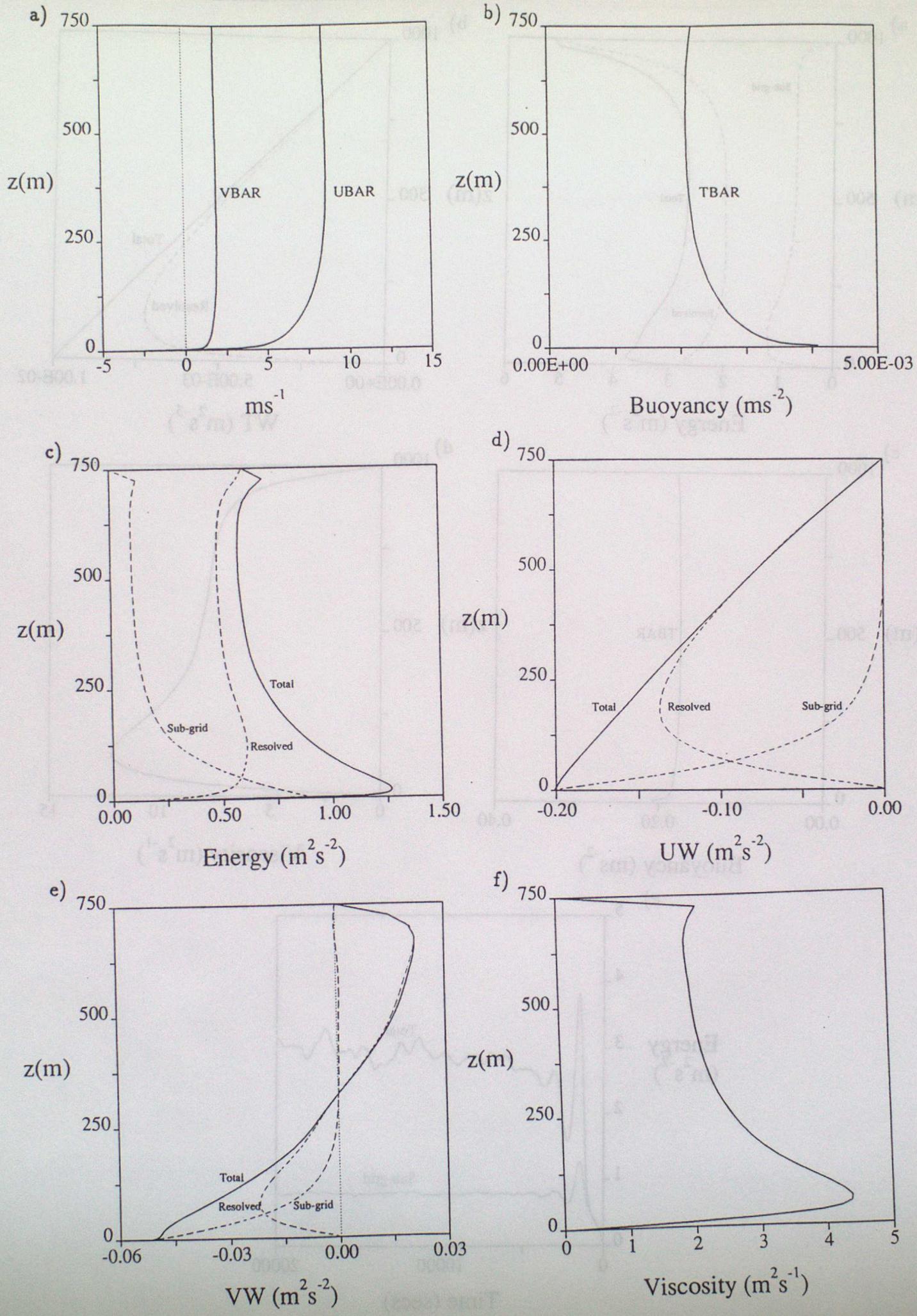

Figure 9: Results from a fully-three-dimensonal run of large-eddy model using the Mason scheme. The run is at $t = 20\,000$ seconds. The following vertical profiles are shown – a) UBAR and VBAR; b) TBAR; c) Energy; d) UW; e) VW; f) Viscosity.

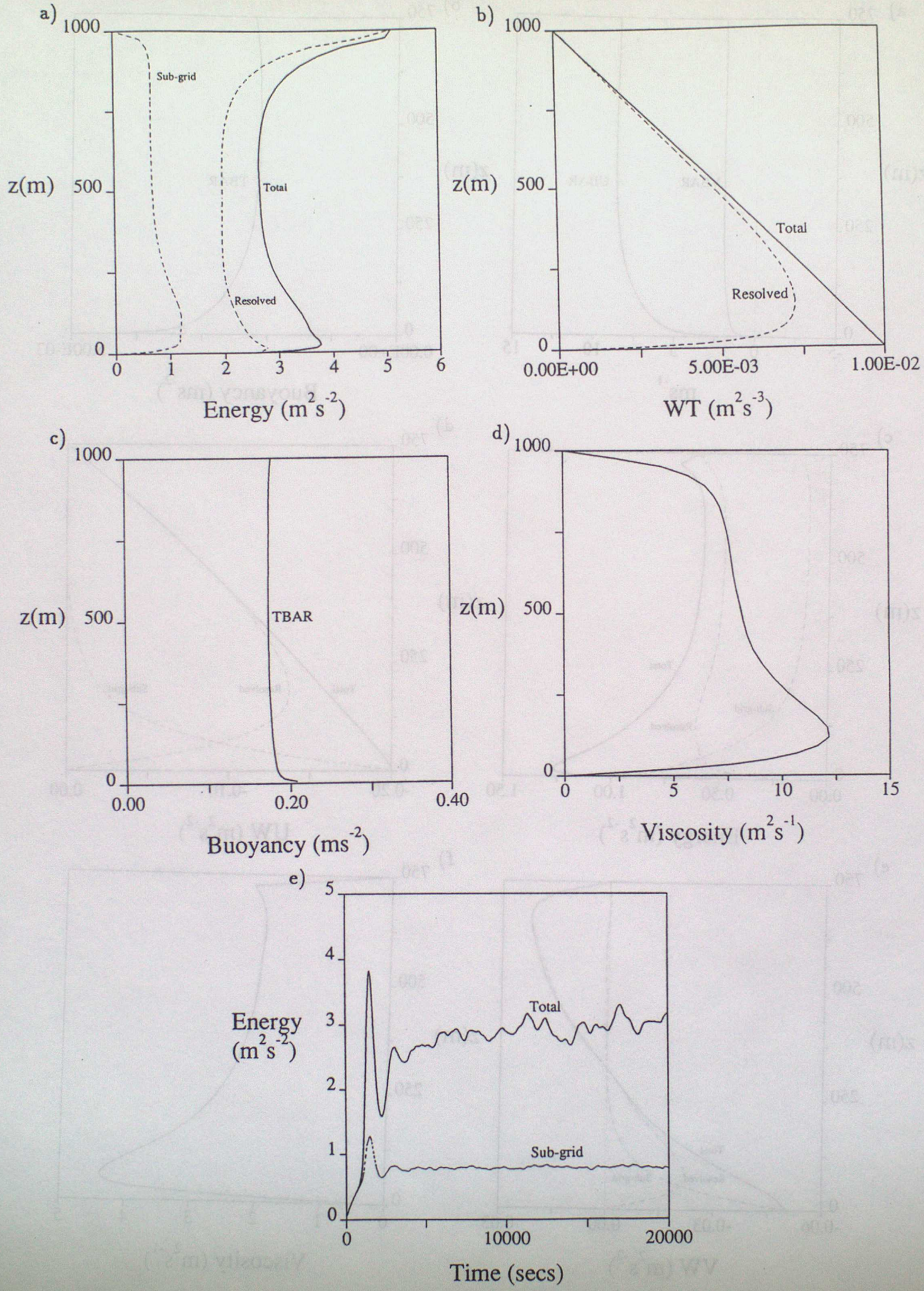

Figure 10: Results from a realistic, convective run of large-eddy model using the original forward-in-time explicit scheme. The run is at $t = 20\,000$ seconds. The following vertical profiles are shown – a) Energy; b) WT; c) TBAR; d) Viscosity. A time-series of Energy through-out the run is shown in e).

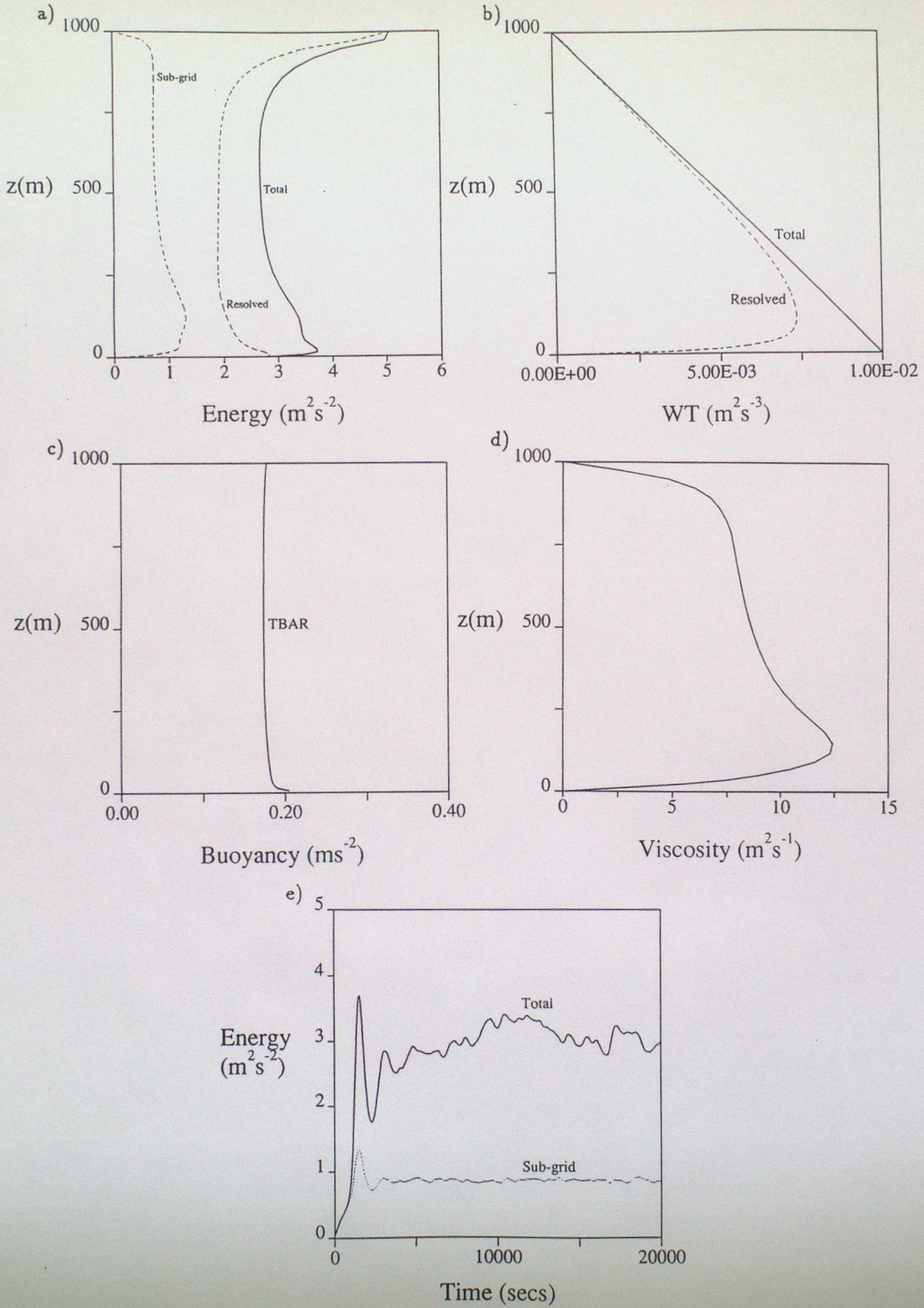

Figure 11: Results from a realistic, convective run of large-eddy model using the Mason scheme. The run is at $t = 20\ 000$ seconds. The following vertical profiles are shown – a) Energy; b) WT; c) TBAR; d) Viscosity. A time-series of Energy through-out the run is shown in e).