282

MET.O.14

METEOROLOGICAL OFFICE
BOUNDARY LAYER RESEARCH BRANCH
TURBULENCE & DIFFUSION NOTE

T.D.N. No. 156

# 3-D Large Eddy Model
# Cyber 205 Version.

by

N.S. Callen & P. Dickinson.

November 1984.

Please note: Permission to quote from this unpublished note should be obtained from the Head of Met.O.14, Bracknell, Berks., U.K.

## Introduction

Two versions of the 3-D large eddy model have now been written for the Cyber 205.

Version 1 makes extensive use of concurrent I/O using one permanent and two temporary datasets. At any one time only three horizontal levels of the 3-D fields are held in core while the remaining levels are stored in the temporary datasets on disk. Unfortunately the ratio of CPU time to elapsed time for any model integration proved too low for this version to be considered as an efficient working model.

Version 2 only uses concurrent I/O at the beginning and the end of a model run (or otherwise specified by user) using one permanent dataset on disk. All fields are held in core throughout. Consequently the maximum number of grid points allowable has been reduced from 64*64*64, for version 1, to 32*32*48 but the ratio of CPU time to elapsed time increases over twenty-fold to 0.85. This is a highly acceptable value and version 2 has been adopted as the current working model.

This note describes version 2 only. The purpose is not only to outline the mathematical formulation of the model but also to give details of the model's structure.

## 1. Basic Equations

The basic equations used in the model are the ensemble-averaged Navier Stokes equations and the continuity equation ie.

$$\frac{Du}{Dt} = -\frac{\partial p}{\partial x} - \frac{\partial p_o}{\partial x} + fv + \gamma(z)\big(u - u_o(z)\big) + \frac{\partial}{\partial x}\tau_{11} + \frac{\partial}{\partial y}\tau_{12} + \frac{\partial}{\partial z}\tau_{13} \qquad (1.1a)$$

$$\frac{Dv}{Dt} = -\frac{\partial p}{\partial y} - \frac{\partial p_o}{\partial y} - fu + \gamma(z)\big(v - v_o(z)\big) + \frac{\partial}{\partial x}\tau_{21} + \frac{\partial}{\partial y}\tau_{22} + \frac{\partial}{\partial z}\tau_{23} \qquad (1.1b)$$

$$\frac{Dw}{Dt} = -\frac{\partial p}{\partial z} + B + \gamma(z)(w) + \frac{\partial}{\partial x}\tau_{31} + \frac{\partial}{\partial y}\tau_{32} + \frac{\partial}{\partial z}\tau_{33} \qquad (1.1c)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \qquad (1.1d)$$

where $\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z}$

$u, v, w$ – the mean velocities in cartesian (x,y,z) coordinates

$f$ – the coriolis parameter.

$p$ – perturbution pressure

$p_o$ – linearly varying background pressure (ie. $\frac{\partial p_o}{\partial x}$ & $\frac{\partial p_o}{\partial y}$ are constants)

$u_o, v_o, w_o$ – initial velocity fields

$\gamma$ – rayleigh damping term.

$B$ – buoyancy ($= \frac{g}{\bar{T}}(T-\bar{T})$ in a dry Boussinesq atmosphere)

$\tau_{ij}$ – turbulent Reynolds stress tensor.

Eventually, two further scalar equations will be included ie. a) liquid water potential buoyancy and b) total water. Although the subroutines for timestepping these two extra fields still need to be written, the space for the necessary extra arrays has been allocated within the model. This should avoid the problem of having to re-arrange the storage space at a later stage.

## 2. Turbulence Parametrization

For a detailed explanation of the turbulent parametrization see Mason & Sykes (1982). Here, only an outline will be given together with reference to the array names used in the program. (Note: variable and array names used in the program will be printed in block capitals and underlined).

Following Smagorinsky (1963) we define

$$\tau_{ij} = \nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \qquad (2.1)$$

where $\nu = \nu_H$ (<u>VISH</u>), the horizontal viscosity

or $\nu = \nu_V$ (<u>VIS</u>), the vertical viscosity.

This horizontal viscosity, $\nu_H$ , is introduced to be used in conjunction with the horizontal derivatives of the horizontal velocity components.

We define the horizontal viscosity

$$\nu_H = l_0^2 S_H^{1/2} \qquad (2.2)$$

where $l_0$ is a mixing length scale and $S_H$ is the horizontal deformation (<u>DEFH</u>) given by

$$S_H = 2\left(\frac{\partial u}{\partial x}\right)^2 + 2\left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)^2 \qquad (2.3)$$

Similarly we define the vertical viscosity

$$\nu_V = l(z)^2 S^{1/2} \qquad (2.4)$$

where $l(z)$ is the mixing length given by

$$\frac{1}{l(z)} = \frac{1}{K(z+z_0)} + \frac{1}{l_0} \qquad (2.5)$$

$z_0$ : roughness length

$K$ : von Karmann's constant

and $S$ is the vertical deformation (DEFV) given by

$$S = S_H + 2\left(\frac{\partial w}{\partial z}\right)^2 + \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right)^2 \qquad (2.6)$$

So that isotropic viscosity is maintained away from the boundaries, the horizontal viscosity, $\nu_H$ , is set equal to the vertical viscosity, $\nu_V$ if its computed value is less than $\nu_V$ (i.e. let $\nu_H = \nu_V$ if computed $\nu_H < \nu_V$ ).

## 3. Finite Difference Method

A semi-implicit method is used to integrate the momentum equations using both Du Fort Frankel and Central Difference schemes.

Omitting spacial subscripts equation, (1.1a) may be written in finite difference form as

$$\frac{u^{t+1}-u^{t-1}}{2\Delta t} = -u^t\frac{\Delta u^t}{\Delta x} - v^t\frac{\Delta u^t}{\Delta y} - \omega^t\frac{\Delta u^t}{\Delta z} - \frac{\Delta p^t}{\Delta x} - \frac{\Delta p_o}{\Delta x}$$

$$+ fv^t + \gamma(z)\left(u^t - u_o(z)\right) + \frac{\Delta}{\Delta x}\left(2\nu^t\frac{\Delta u^t}{\Delta x}\right) + \frac{\Delta}{\Delta y}\left(\nu^t\frac{\Delta v^t}{\Delta x} + \nu^t\frac{\Delta u^t}{\Delta y}\right)$$

$$+ \frac{\Delta}{\Delta z}\left(\nu^t\frac{\Delta \omega^t}{\Delta x} + \nu^t\frac{\Delta u^t}{\Delta z}\right) \qquad (3.1)$$

Expanding the last three terms on the RHS of (3.1), adding spacial subscripts and replacing $u^t_{i,j,k}$ by 0.5* ($u^{t+1}_{ijk}$ + $u^{t-1}_{ijk}$ ) wherever it occurs the form of the viscous terms is

$$\frac{2}{\Delta x^2}\left[\nu^t_{i+1,j}\left(u^t_{i+1,j} - u_R\right) - \nu^t_{i,j}\left(u_R - u^t_{i-1,j}\right)\right]$$

$$+\frac{1}{\Delta y}\left\{\frac{\left[\nu^t_{i,j}\left(v^t_{i+1,j} - v^t_{i,j}\right) - \nu^t_{i,j-1}\left(v^t_{i+1,j-1} - v^t_{i,j-1}\right)\right]}{\Delta x} + \frac{\left[\nu^t_{i,j}\left(u^t_{i,j+1} - u_R\right) - \nu^t_{i,j-1}\left(u_R - u^t_{i,j-1}\right)\right]}{\Delta y}\right\}$$

$$+\frac{1}{\Delta z}\left\{\frac{\left[\nu^t_{i,j}\left(\omega^t_{i+1,j} - \omega^t_{i,j}\right) - \nu^t_{i,j,k-1}\left(\omega^t_{i+1,j,k-1} - \omega^t_{i,j,k-1}\right)\right]}{\Delta x} + \frac{\left[\nu^t_{i,j}\left(u^t_{i,j,k+1} - u_R\right) - \nu^t_{i,j,k-1}\left(u_R - u^t_{i,j,k-1}\right)\right]}{\Delta z}\right\}$$

$$(3.2)$$

where $u_R$ = 0.5*( $u^{t+1}_{ijk}$ + $u^{t-1}_{ijk}$ ) and the third spacial subscript has been omitted where it equals k. (NB. no discrimination has been made between $\nu_H$ and $\nu_V$ in the representation of $\nu$ in (3.2)). Using (3.2), equation (3.1) may be written as

$$\frac{u^{t+1}-u^{t-1}}{2\Delta t} = -\frac{\Delta p^{t}}{\Delta x} + \underline{SLU} + \underline{SDFU}\left(u^{t+1}+u^{t-1}\right) \qquad (3.3a)$$

where <u>SDFU</u> contains the coefficients of ( $u^{t+1} - u^{t-1}$ ) and <u>SLU</u> contain all the remaining terms on the RHS except the pressure gradient term.

Similarly equations (1.1b & 1.1c) may be written in finite difference form as

$$\frac{v^{t+1}-v^{t-1}}{2\Delta t} = -\frac{\Delta p^{t}}{\Delta y} + \underline{SLV} + \underline{SDFV}\left(v^{t+1}+v^{t-1}\right) \qquad (3.3b)$$

$$\frac{w^{t+1}-w^{t-1}}{2\Delta t} = -\frac{\Delta p^{t}}{\Delta z} + \underline{SLW} + \underline{SDFW}\left(w^{t+1}+w^{t-1}\right) \qquad (3.3c)$$

The SL's are called the source terms and involve the central time level. The SDF's are the coefficients of the Du Fort Frankel terms and involve both the upper and lower time levels (t-1 and t+1).

P is the pressure derived from the finite difference Poisson equation. Taking the divergence of equations (3.3a, 3.3b and 3.3c) we obtain

$$\frac{1}{2\Delta t}\nabla.\underline{u}^{t+1} - \frac{1}{2\Delta t}\nabla.\underline{u}^{t-1} = -\nabla^{2}P^{t} + \nabla.\underline{SL} \qquad (3.4)$$

The SDF terms have been included in the source terms here since the central time level, t, must be used as an approximation to levels t+1 and t-1. To satisfy continuity we require $\nabla.\underline{u}^{t+1} = 0$ therefore P is obtained by solving

$$\nabla^2 p^t = \nabla . \underline{SL} + \frac{1}{2\Delta t} \nabla . \underline{u}^{t-1} = \underline{P} \qquad (3.5)$$

Owing to numerical precision, $\nabla . \underline{u}^{t-1}$ does not equal zero and must be included. The solution is obtained by a direct method using a fast fourier transformation in the horizontal and line inversion of the resulting tridiagonal matrices in the vertical.

## 4. Boundary Conditions

### a) Frame of reference

To relax the restriction imposed on the value of $\Delta$ t by the Courant-Friedrichs-Lewy condition for horizontal advection, the mesh of grid points is advected along with the mean flow. The mean velocity of the flow is calculated every timestep. Every time the program dumps data to disk it adjusts the mean velocity of the flow to equal zero. This mean velocity is denoted by the variables $\underline{UBC}$ and $\underline{VBC}$. Note that all the horizontal velocity fields are printed with the mean subtracted. The true velocity values are $\underline{u}$ + $\underline{UBC}$ and $\underline{v}$ + $\underline{VBC}$.

### b) Lower boundary conditions

The true values of u and v at the surface are zero. It follows from a) that these should be set to u = $-\underline{UBC}$ and v = $-\underline{VBC}$. Also w=0.

### c) Upper boundary conditions

The upper boundary is a stress free rigid lid

$$\text{i.e.} \quad \frac{\partial u}{\partial z} = \frac{\partial v}{\partial z} = 0, \quad w = 0 \qquad (4.1)$$

### d) Horizontal boundary conditions

For all fields the horizontal boundary conditions are periodic such that f(x+W)=f(x) where W is either the length or width of the model domain. Using periodic boundary conditions allows full use to be made of the Cyber's vectorisation capabilities. See section 6(b).

## 5. Numerical Grid

The variables are stored on a staggered grid which is stretched in the vertical. (See fig 1). The vertical velocity w is stored on the 'Z' grid and all other variables are stored on the 'ZN' grid. (See figure 2). The program uses the arrays Z and ZN to store the vertical heights of grid pts and the arrays DZ and DZN are used to store the distances between consecutive points on the two grids respectively. The grids are uniform in the horizontal (ie. X and Y directions) so DX and DY are constants.

## 6. Model Structure

### a) Parameter statement

The present maximum dimensions of the model are 32*32*48 (49152). That is, 32 grid pts are allowable in both the x and y horizontal directions and 48 grid points in the vertical. The values are controlled by the variables NI - max no of grid pts in x-direction

NJ - max no of grid pts in y-direction

NK - max no of grid pts in z-direction

which appear in the first line of the Parameter statement. Any alteration to the value of one of the variables must be made in every Parameter statement which appears at the beginning of every subroutine.

The remaining variables in the parameter statement are calculated using NI, NJ and NK and they are used to set the maximum dimensions for the various storage arrays.

### b) Fields

Since we are using Du Fort Frankel terms, the velocity fields are needed at two time levels, t and t-1. If we denote u at time t-1 by ZU etc then the fields used are U, V, W, T, VIS, VISH, ZU, ZV, ZW and ZT. Space is available in the common blocks for Q, ZQ, VISB and BUOY (moisture at times t and t-1, viscous term due to buoyancy and buoyancy itself) although these are not used at the moment. The above 10 fields are stored at every horizontal level in the array STORE. In order to calculate finite differences in the vertical, three levels of these fields need to be worked on at any one time. So the arrays U, V, W, T, VIS, VISH, ZU, ZV, ZW and ZT are dimensioned to hold 3 levels (eg K-1, K, K+1). All calculations are performed for the central level, k, while the fields are in these 10 arrays. Then either the upper or lower level is transferred back to STORE to be replaced by the next level. The source fields SLU, SLV, SLW, SDFU, SDFV, SDFW are not stored but are calculated immediately prior to their requirement. The pressure and pressure source fields, P, for all levels are held permanently in core, overwriting each other as each time step is made.

The number of grid points for any model integration is specified by the user. In the vertical there are KK (max NK) points, in the x-direction II (max NI) and in the y-direction JJ (max NJ). These variables are used to calculate the working size of the various arrays. Then, because the fields are periodic in the horizontal, they are expanded to maximise the vector length and consequently take full advantage of the Cybers vector operations.

Hence instead of doing calculations on JJ vectors of length II we do them on vectors either (II + 2) * (JJ + 2) or (II + 2) *JJ - 2 long. The horizontal fields U, V, W, T, VIS & VISH are all expanded to the former length while the remaining four fields are expanded to the latter. The fields at time t-1 are shorter because no derivatives need to be calculated from them and the two extra rows aren't needed. During the execution of the program all fields are stored in their expanded form but they are compressed again before being dumped to disk.

The only constraints on the values of II and JJ besides their maximum values is set by the fast fourier transform used to solve the Poisson equation (Temperton 1982). This requires that both II and JJ are of the form $II = 2^{l} 3^{m} 5^{n}$

$JJ = 2^{p} 3^{q} 5^{r}$ where l,m,n,p,q,r are all integers.

c) I/O File

Between consecutive runs of the program, the fields U,V,N,T,Q,ZU,ZV, ZW,ZT,ZQ are held on a permanent file, the "Restart" file. The fields are dumped to this file at the end of each run and they can also be dumped at various times, as specified by the user, during the run. All other fields, ie. viscosities, pressure and sources are calculated during the execution of the program and not retained between program runs. The fields when stored on the "Restart" dataset are all unexpanded i.e they are II * JJ long.

The number of disk sectors required for each level of the "Restart" dataset is

NRS = (10 * II*JJ-1)/512 + 1

There are 2 extra disk sectors at the start of the "Restart" file, containing control information about grids, timestep etc so total size of the "Restart" file is KK*NRS+2 disk sectors.

d) Common Blocks

At present there is a total of 12 common blocks

1. FIELDS It contains the velocity fields for levels K+1, K, K-1 at time t (or t+1 after time stepping). i.e. fields U,V,W,T,Q.

2. MFIELDS Contains the velocity fields for levels K+1, K, K-1 at time t-1 (or time t after time stepping) i.e. fields ZU, ZV, ZW, ZT, ZQ.

3. VISCOS Contains the viscosities and buoyancy for levels K+1, K, K-1 at time t. i.e. fields VIS, VISH, VISB, BUOY.

4. SOURCE Contains the source fields SLU, SLV, SLW, SDFU, SDFV, SDFW for levels K+1 and K at time t. It also contains the expanded pressure fields for levels K+1 and K.

5. CNTRL Contains the control parameters for the run i.e. grid dimensions, timestep value etc. The first 1024 parameters are those written to the first 2 disk sectors of the "Restart" dataset, and have been read from there.

6. EIGEN Contains the eigenvalues required by the Poisson solver, and the control arrays initialised in SETFIL and used in RPASSQ and QPASSQ, the fourier transform routines.

7. IN Contains the input buffer into which the datafields are read from STORE.

8. OUT Contains the output buffer from which the datafields are written back to STORE. It is also used to store a work field in the subroutine POISSON and to store vertical fields in the subroutine PRNTDIAG.

9. WORK This contains temporary work arrays used during the execution of the program.

10. RESTART Contains the buffer used to hold the datafields which are being transferred to or from the "Restart" dataset.

11. PRESS Contains the pressure source or pressure fields.

12. STORES Contains the fields U, V, W, T, VIS, VISH, ZU, ZV, ZW, ZT at all levels.

Since two of these common blocks, RESTART and CNTRL are used as I/O buffers it is essential to align them on large page boundaries, using either the GRLP or GROL parameters on the LOAD card (See Section 7). Provided common block RESTART is of a length divisible by 512 (the parameter statement ensures this) CNTRL may follow RESTART in its positioning on a large page.

N.B. Because CNTRL is of a length not divisible by 512 (1 small page) on no account must RESTART follow CNTRL on the same large page. The program will fail !

e) Set Up, Main Program + Subroutines

1. Set Up

As its name implies it is the set-up program used to create the "Restart" file, prior to running the 3-D large eddy model.

At present, input to this program is via a cyber dataset called A3DSETn (where n is in the range A → Z).

The program simply reads in the control parameters and 1-D fields from the dataset and writes these to the first 2 disk sectors on the "Restart" dataset. The initial fields are then read in, expanded into 3-D and then written to the "Restart" dataset, a level at a time.

This program only need be run once, at the very beginning of each model run.

## 2. MAIN Program

Firstly the set-up parameters are read in from the "Restart" file followed immediately by the reading of the datafields. As the first level (K=1) is read in bit patterns (see Cyber Manual, Section 14 p 25) are set up and various useful parameters generated for later subroutines. This is done simultaneously. As level K is read from disk the bit patterns are used to expand the datafields for level K-1. Viscosities are calculated and then all the fields are written to STORE.

The fields up to level KK are written to STORE. The top 3 fields are also retained in the working arrays U, V, etc ready to be used.

Finally in the program, eigenvalues are calculated for the Poisson routine before subroutine NSTEP is called.

## 3. Subroutines

### (i) NSTEP

This subroutine steps the fields through NSTP timesteps. Data is read in from the STORE array starting at level KK-3. The source terms SLU, SLV etc and pressure source terms P are calculated and the fields are then read back into STORE. On reaching level 1 the pressure field is solved.

Then – travelling back up the array – data are read in from STORE, the sources are recalculated and the fields are timestepped before being written back to STORE. This is repeated NSTP (total number of timesteps) times.

At present no space is available in core to permanently store the source terms. They consequently need to be calculated twice.

The dumping of the fields back to disk is carried out every NDUMP steps. Selected slices are printed every NPRINT steps and also when a dump occurs. Every NDIAG steps a 2-D plot of the fields is output. NSTP, NDUMP, NPRINT & NDIAG are entered by the user with the JCL.

(ii) SOURCES (INDIC)

This subroutines generates the source functions. The argument INDIC takes the value 0 or 1. If INDIC=0 the arrays are being traversed from level KK to 1 (i.e. downwards). Here the arrays U, V and W at level K-1 hold values at time t and if INDIC=1 the arrays are being traversed from level 1 to KK (ie. upwards). Now the arrays U, V and W at level K-1 hold values at time t+1. These fields have already been timestepped. To generate the correct source terms the level K-1 values of U, V and W at time t are used. These having previously been stored in DUU, DUV and DUW respectively.

(iii) PSOURCE

Generates the pressure source functions for use in solving the pressure fields. Called on the downward traverse of the arrays directly after the SOURCE subroutine.

### (iv) VISCB (INDVIS)

Calculates the viscosities as the arrays are being traversed upwards therefore the vertical velocity, W, at level K-1 has already been timestepped and cannot be used in the calculations.

As mentioned above DUW holds the K-1 W fields at time t and it is used when INDVIS is set to equal 1. However, the array DUW is not properly defined until levels 2, 3 and 4 are being used (ie. K=3). Therefore when K=2 INDVIS=0 which allows the W field at time t+1 to be used. This is permissible since the W field at K-1=1 is equal to zero for all t.

### (v) POISSON

Generates the pressure field from pressure source by solving the Poisson equation. Upon entry to the routine P contains the source field. Upon return from the routine, P contains the generated pressure field.

### (vi) STEPMOM

Timesteps the momentum equations. It is here the arrays DUU, DUV and DUW are defined before timestepping. Timesmoothing also occurs directly after stepping. The value of the timesmoothing parameter, EPS, is set by the user and contained in the common block CNTRL.

### (vii) DOBC

Sets the horizontal boundary conditions for all fields.

### (viii) PRNTFLDS (MS, FIRST, PRINT, DIAG)

This subroutine is included for development purposes only and it will eventually be removed. The numerical values of U, V, W, T, P, VIS & VISH are printed for selected slices. At present 9 slices are printed ie. 3 in each of the XY, YZ and XZ planes.

The argument MS indicates the timestep. FIRST, PRINT and DIAG are logical variables. FIRST=.TRUE. before any timesteps have been executed. All fields are printed except P which is not yet defined.

Subsequently FIRST=.FALSE. and P is included in the printed fields. The values of PRINT and DIAG indicate whether the fields and/or their plots will be printed at step MS.

(ix) PRNTDIAG (N1, N2, N3)

Produces a plot of the fields. N2 indicates the field (ie. U, V, W, etc). The arguments N1 and N3, respectively, detail which slice and which plane (ie. XY, XZ, YZ) are to be printed.

(x) INFLDS (J1, J2, IK)

Sorts the fields into their individual arrays (ie. U, V etc) from the array FLDSIN. The array, FLDSIN, is really superfluous and at any time is an exact copy of one horizontal slice of STORE. It is a relic from when the model used temporary disk datasets and buffers were necessary. However, the routine is straightforward, efficient and small and removal of this subroutine would probably cause more trouble than its worth.

The arguments J1 and J2 indicate the range of fields to be sorted. IK indicates the level (ie. K-1, K or K+1).

(xi) OUTFLDS (J1, J2, IK)

Transfers and packs the fields from their individual arrays into the array FLDSUT before being transferred to STORE. Like FLDSIN the array is superfluous and could be skipped during the transfer.

But unlike FLDSIN the array is very large. This is because it doubles as work space in the Poisson subroutine and, when necessary, holds the sorted vertical slices in subroutine PRNTDIAG. The space is therefore necessary and since the routine is straightforward and efficient it has been kept for the same reasons as INFLDS. The arguments are defined as in INFLDS.

(xii) TIMER (SUB,I)

Records the flow through and time spent in each subroutine. Calls to TIMER must be made before the first executable statement of each subroutine and before each return point. The argument SUB indicates the subroutine name. I indicates whether the call is from the beginning or end of a routine. See program for fuller details.

## 7. Running the program

### a) Dataset storage

As mentioned earlier the initial data for a program run is obtained from a Cyber dataset A3DSETn (n = A to Z). To create a new dataset it is first necessary to run the 1-D model (M14.SOURCE (AIN2D)) using the JCL on M14.JOBLIB (AIN2D). The parameters in the namelist being altered accordingly. The program will write, in the necessary format, all the relevant data to the dataset M14.ACYBSET. This dataset may now be transferred over the IBM-Cyber link for storage on the Cyber.

On the Cyber the set up program is stored as ASETUPn (n = A to Z) where the further the letter n is from A the later the version. The 3-D model is stored as ALGEDn (n as before). All the above Cyber datasets are stored on user No. 140002.

The latest versions of the set up program and the 3-D model are also stored in two IBM datasets, M14.A3DSTUP and M14.A3DCYB, respectively.

This allows any changes in either of the programs to be made on TSO before the program is transferred via the link.

The necessary JCL to create new versions of the model may be found on M14.JOBLIB (ACNEWRUN), whereas JCL for use with existing datasets may be found on M14.JOBLIB (ACOLDRUN).

b) Important points

i) In order to execute, the program requires Fast Fourier transforms (routine written by C. Temperton, Met O 11) and trignometric routines (written by A. Foreman, Met O 12).

The fast Fourier routines can be found as the decks SETFIL, RPASSQ, QPASSQ on the update library CDEXB in the pool P11. The trignometric routines can be found in the binary file TRIG64 in the pool P12LIB.

ii) For efficient execution the correct grouping of common blocks on large pages is essential. One large page contains 64K (65536) bytes. With the present maximum dimensions of the model it is necessary to utilise 13 large pages. The maximum available is 14 but space needs to be left for storage of the fortran. Table 1 gives details of common blocks sizes and how they have been arranged on the large pages. (NB 'RESTART' and 'CNTRL' are aligned on a large page boundary).

To arrange the common blocks the GRLP parameter needs to be used in the LOAD statement of the JCL. It is worth noting that the common blocks are arranged on each large page in the order that they are encountered in the program NOT in the order stated in the JCL.

c) Running times

Compared with the IBM version of the 3D large eddy model the CPU time per timestep is considerably reduced.

The CPU time per timestep for a 20*26*16 grid after 1200 timesteps is 1.97 secs for the IBM model reducing to 0.08 secs for the Cyber version. As yet, very few tests have been made on execution speed but the above figures are encouraging.

d) If required the Cyber version/model which uses considerable disk I/O may be found stored on the IBM datasets M14.ADISKSU (setup) and M14.ADISKMOD (model).

## Acknowledgement

We wish to thank Dr. P.J. Mason, who developed the equations used, for his helpful guidance in writing and structuring the model.

## References

Mason, P. J. & Sykes, R. I. 1982 "A two-dimensional numerical study of horizontal roll vortices in a inversion capped planetary boundary layer". Quart. J. R. Met. Soc. 108 pp 801-823.

Smagorinsky, J. 1963 "General circulation experiments with the primitive equations: 1 The basic experiments" Mon. Weath. Rev. 91 pp 99-164.

Temperton, C. 1982 Fast Real Fourier Transforms on the Cyber 205.
Met O 11 Technical Note No. 155.

| Common Block | Size (bytes) | Total Size | No of |
|---|---|---|---|
| | | For Group | Large Pages |
| RESTART | 10240 | | |
| CNTRL | 1460 | | |
| FIELDS | 17340 | 64002 | 1 |
| MFIELDS | 16290 | | |
| SOURCE | 18672 | | |
| VISCOS | 13872 | | |
| PRESS | 49152 | 65156 | 1 |
| EIGEN | 2132 | | |
| STORES | 541440 | | |
| IN | 11776 | 576280 | 9 |
| WORK | 23064 | | |
| OUT | 82944 | 82944 | 2 |
| TOTAL | | 788382 | 13 |

Table 1: Present grouping of common blocks on large pages (model dimensions 32*32*48) 1 Large page contains 64K (65536) bytes.

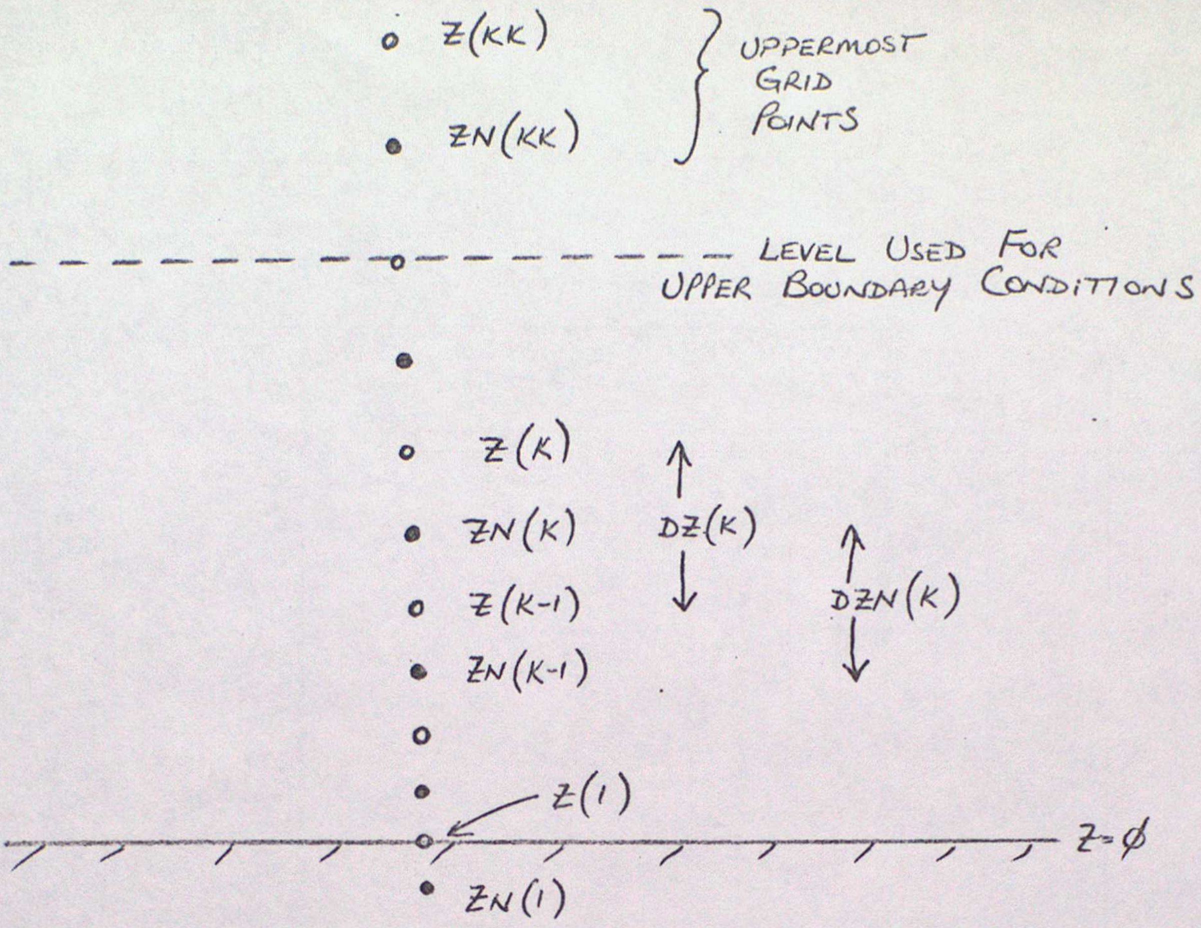


FIGURE 1 THE STRETCHED VERTICAL GRID

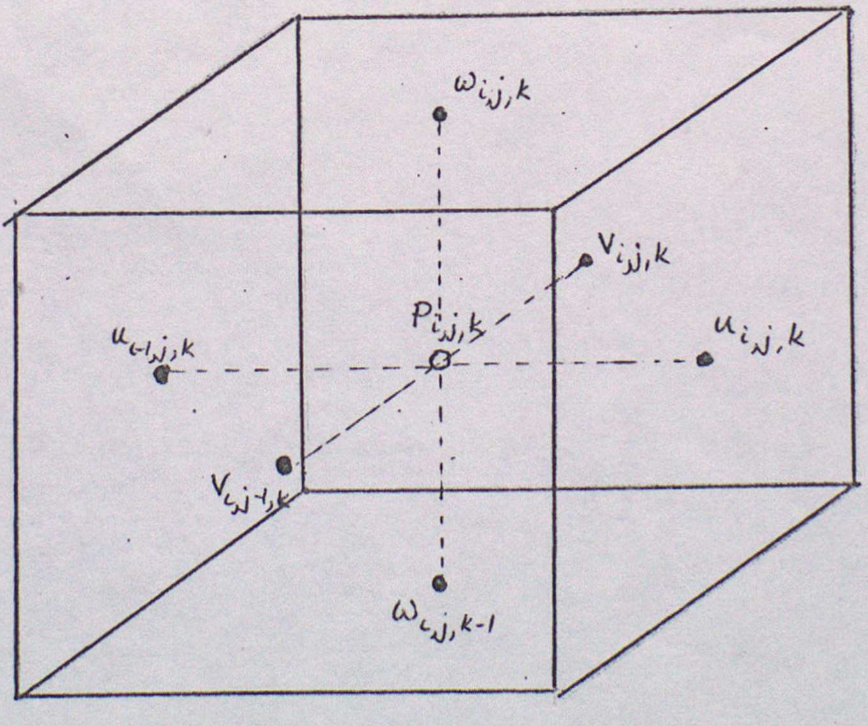
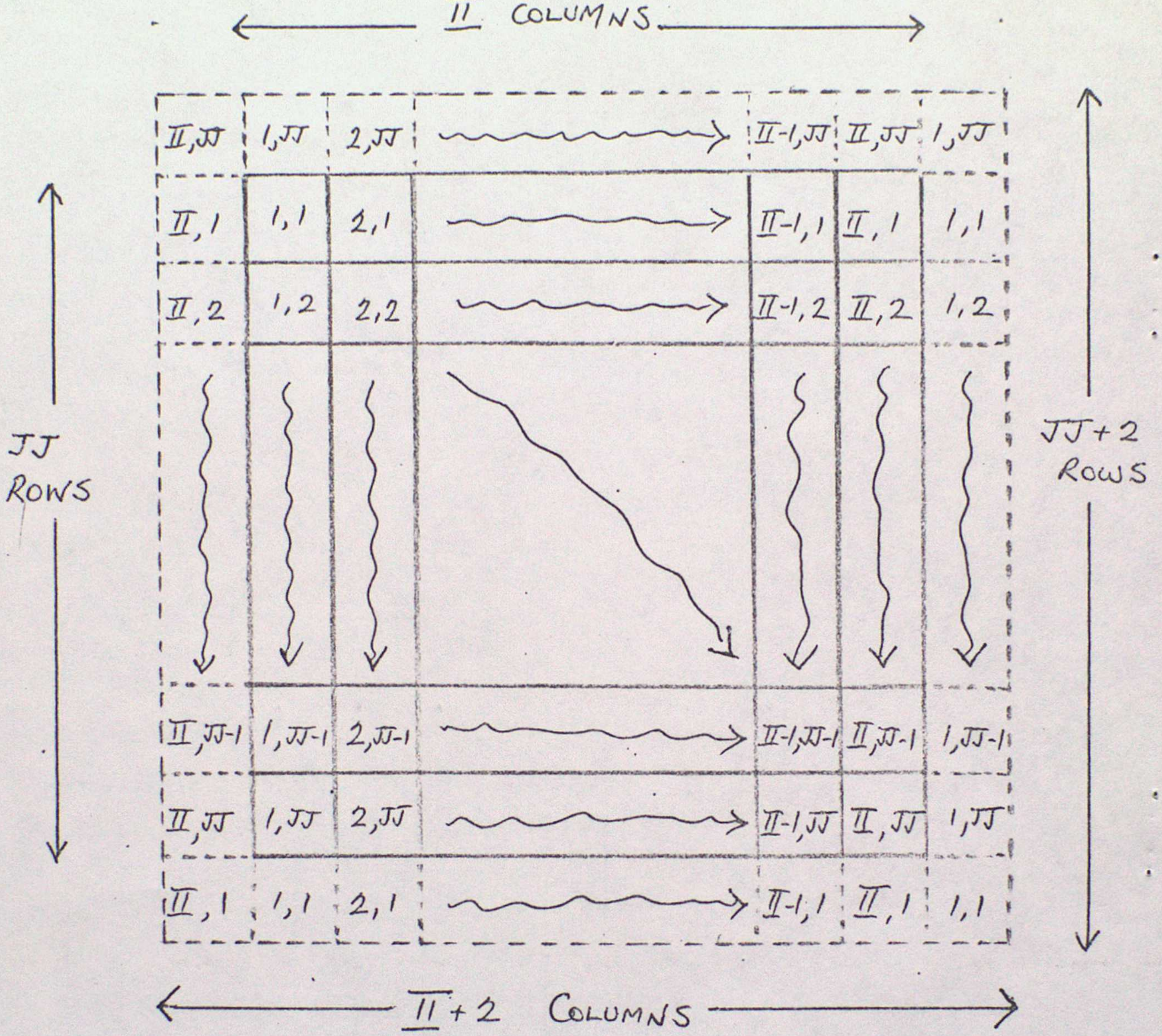


FIGURE 2 THE 3-D MESH SHOWING STORAGE POSITIONS OF VARIABLES.



| | | | | | | |
|---|---|---|---|---|---|---|
| II,JJ | 1,JJ | 2,JJ | ⟶ | II-1,JJ | II,JJ | 1,JJ |
| II,1 | 1,1 | 2,1 | ⟶ | II-1,1 | II,1 | 1,1 |
| II,2 | 1,2 | 2,2 | ⟶ | II-1,2 | II,2 | 1,2 |
| ↓ | ↓ | ↓ | ↘ | ↓ | ↓ | ↓ |
| II,JJ-1 | 1,JJ-1 | 2,JJ-1 | ⟶ | II-1,JJ-1 | II,JJ-1 | 1,JJ-1 |
| II,JJ | 1,JJ | 2,JJ | ⟶ | II-1,JJ | II,JJ | 1,JJ |
| II,1 | 1,1 | 2,1 | ⟶ | II-1,1 | II,1 | 1,1 |

FIGURE 3 EXPANSION OF THE HORIZONTAL GRID

— DENOTES THE ORIGINAL GRID

- - - DENOTES THE EXPANDED GRID