Met O (APR) Turbulence and Diffusion Note No. 213

The Meteorological Office Large-Eddy Simulation Model

by

S.H.Derbyshire, A.R.Brown and A.P.Lock

21st October 1994

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

Note

The Meteorological Office Large-Eddy Simulation Model

S.H.Derbyshire, A.R.Brown and A.P.Lock

Contents

The Meteorological Office Large-Eddy Simulation Model

S.H.Derbyshire, A.R.Brown and A.P.Lock

21st October 1994

Abstract

The new version of the Meteorological Office large-eddy simulation model is described. This represents a complete recoding of P.J.Mason's original model, with many more options and many structural changes. New options include a deep anelastic (as opposed to incompressible Boussinesq) equation set, arbitrarily many scalar variables and detailed cloud microphysics. The subgrid model can be controlled fairly easily and stochastic backscatter may be invoked. There is a choice of advection schemes. The model can be run in 1-D, 2-D or 3-D mode.

A quick-look summary of the main equations and code-structure is available separately.

1 Introduction

Large-eddy simulation (LES) of turbulence has been applied in the Meteorological Office to numerous boundary layer problems with considerable success, as described e.g. by Mason and Callen (1986), Mason and Thomson (1987), Mason (1989), Mason and Derbyshire (1990), MacVean (1993).

With the advent of the CRAY-YMP in 1990, it was decided to produce a new 3-D vector code based on the structure of the IBM code but using vertical rather than horizontal slices. There is also an option, rarely invoked, to keep slices not currently being used (the 'packed' slices) as 32-bit rather than 64-bit fields in order to save memory. At the same time, it was decided to enhance the scientific flexibility of the model. The basic equations are changed from incompressible Boussinesq to 'deep anelastic', thus removing a restriction to the lower troposphere. A strictly incompressible Boussinesq system is easily recovered, if desired.

The present Meteorological Office large-eddy model (LEM) is based on a complete recoding and extension of P.J.Mason's original model, against which it has been checked. The present version was first written by S.H.Derbyshire, with contributions from A.R.Brown, M.E.B.Gray, G.J.Shutts and H.A.Swann. This was later revised and somewhat restructured by M.K.MacVean to produce a version compatible with the Cray Update facility, with efficient switching of storage allocation for the various options. As far as possible the code uses standard FORTRAN and avoids special or non-portable calls.

The LEM has undergone rapid development over the past few years, and this note only attempts to document the basic version. However this basic version has reached a relatively steady state with version 1.4, and various studies based on this version have been submitted for publication and/or conducted with external collaborators. Sources of further information will be indicated as appropriate.

In §2 we describe the LES philosophy; in §3 the main analytical equations are given; in §4 software aspects and model structure are discussed; finally the Annex considers numerical methods.

2 What is a large-eddy model?

Whether because LES is younger than numerical weather prediction, or because turbulence is such a vexed topic theoretically, 'philosophical' issues cannot be entirely avoided. For instance, some try to evaluate the 'Smagorinsky subgrid constant', while others dispute that it is a constant in any useful sense. Recent ideas about stochastic backscatter have important conceptual and qualitative implications. Mason (1994) provides the best reference to our approach to these and other issues, and even the 'pragmatic' reader is recommended to study that review paper before proceeding. Other approaches and applications to engineering flows may be found in preprints for the 1st ERCOFTAC conference on Direct and Large-Eddy Simulation, Guildford 1994 (A.R.Brown, N.Wood and S.H.Derbyshire have copies).

Meteorologists sometimes ask 'what is a large-eddy model?', compared to say a forecast model. It is not possible to define a LEM in terms of a specific complete equation set, any more than one can so define a climate model. Most of the equations, especially the 'physics', can be varied. Large-eddy simulation is, by definition, a technique specifically applied to turbulence. Essentially, it attempts to resolve explicitly the larger scale turbulent motions or eddies (which contain most of the turbulent energy and are responsible for most of the turbulent transport) leaving only small scale eddies (which are mainly responsible for the dissipation of kinetic energy) to be parametrised (for which a simple mixing-length closure should be sufficient). This definition of a large-eddy simulation is dependent on the model resolution and so perhaps it is useful to regard a 'Large-Eddy Model' as a code incorporating LES but possibly run sometimes in non-turbulent mode (ie. not completely resolving the energy containing eddies, see §2.1). The following broadly defines a LEM in the sense used in the Meteorological Office:

(i) *Aims*:
A LEM is a process research model which aims to represents a limited number of processes, including *high Reynolds-number turbulence*, as well as possible. By this means, and in conjunction with research-quality observations, we seek to improve our description of such processes, so that their parametrization can be improved in models which need to describe a wider range of processes in (necessarily) less detail. As with the physics, the geometry is normally kept to the minimum complexity required to study a particular problem. The word *simulation*, as opposed to forecast or simple computation, implies that we are really interested in ensemble properties, rather than following the exact time-evolution of single realizations, which in a 'chaotic' system is essentially impossible.

(ii) *Approximations*:
LES approximations in the subgrid model are geared to represent the 3-D isotropic inertial subrange, but also (especially in more recent work) to handle the marginally resolved case, e.g. near a rigid surface.

(iii) *Software*:
Finally we may regard the Meteorological Office LEM as a code which has proved itself successful in boundary layer applications, and is therefore a candidate for wider research use. Extensions should be judged on their merits rather than from any rigid definition of a LEM. However careful step-by-step validation of processes and numerical schemes is an essential criterion. Derbyshire and Kershaw (1993) discuss and illustrate some applications.

2.1 Filtering

The basis of LES is most simply described from the Navier-Stokes equations (although in meteorology we normally incorporate extensions for buoyancy, rotation and possibly moisture or

anelastic effects, see §3.1). Consider the Navier-Stokes equations

$$\frac{D\mathbf{u}}{Dt} = -\nabla p + \nu_{\text{mol}} \nabla^2 \mathbf{u} \tag{1}$$

where the pressure p is determined by the incompressibility constraint

$$\nabla.\mathbf{u} = 0 \tag{2}$$

together with appropriate boundary conditions. Here $\mathbf{u}$ is the flow velocity, $D/Dt \equiv \partial/\partial t + \mathbf{u}.\nabla$ the *material derivative* (rate-of-change following the flow) and ν_{mol} the molecular viscosity.

At high Reynolds numbers *Re*, i.e. at small ν_{mol}, turbulent solutions to Navier-Stokes equations typically develop structure on very small (sub-millimetre) scales and thus cannot be accurately discretized except with an ultra-fine grid. In other words 'Direct Simulation' at true atmospheric Reynolds numbers is virtually impossible. Even if, through enormous advances in supercomputing, this eventually became feasible (say in 50 years), such massive computation would be of doubtful value because we cannot specify surface boundary conditions in such detail anyway.

LES tackles this problem by *filtering* the fundamental equations. All forms of the technique rest on a decomposition of the velocity-field (and analogously of scalar fields) as

$$\mathbf{u} = \mathbf{u}_r + \mathbf{u}_{sf} \tag{3}$$

where by definition

$$\mathbf{u}_r = \mathcal{F}(\mathbf{u}) \tag{4}$$

is the 'resolved' field for some (usually implicit) filtering operator $\mathcal{F}$, which 'removes the smallest scales'. Strictly we should refer to $\mathbf{u}_{sf}$ as the 'subfilter' (rather than 'subgrid') flow, because the filter is not necessarily related to the grid (except that it cannot be *finer* than the grid).

There is considerable arbitrariness in the precise filter operation. The constraints are (i) that the filtered (i.e. resolved) fields be sufficiently smooth for numerical accuracy (ii) that we can parametrize $\mathbf{u}_{sf}$ in an acceptable manner. Furthermore, although some LES workers state that they use a specific (Gaussian/volume-average/other) filter, it is almost impossible to fix the filter in advance because the effective filter depends on parametrization (see Mason 1994). We take the view that the purpose of subgrid parametrization is to minimize the dependence of overall results on the filter scale, i.e. to obtain as nearly as possible the same results at moderate resolution as at high resolution.

Assuming that the operator $\mathcal{F}$ is linear and spatially homogeneous (these assumptions may not hold exactly, but in practice are the least of our worries), expansion of the Navier-Stokes equations gives

$$\frac{D_r \mathbf{u}_r}{Dt} = \nabla.\tau - \mathcal{F}(\nabla p) \tag{5}$$

where the second-rank tensor τ is given by

$$\tau = -\mathcal{F}(\mathbf{u}_r \mathbf{u}_{sf} + \mathbf{u}_{sf} \mathbf{u}_r + \mathbf{u}_{sf} \mathbf{u}_{sf}) \tag{6}$$

(using for compactness the 'dyadic' convention where the tensor $\mathbf{ab}$ is defined to have ijth component $a_i b_j$)

and

$$\frac{D_r}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{u}_r.\nabla \tag{7}$$

It will be noticed that (5) contains no term $\mathcal{F}(\nu_{\rm mol}\nabla^2\mathbf{u})$, and the reason is simply that this is negligibly small at high Re, because of the filter. The energy lost from resolved motions is transferred in the first instance to the subfilter scales, and *not* directly dissipated by molecular viscosity. In other words, to a very good approximation molecular viscosity acts only on the subfilter scales.

The third term on the R.H.S. of (6) resembles a conventional Reynolds-stress term. There has been a tendency to neglect the first two terms; indeed they *are* negligible if a scale separation exists between $\mathbf{u}_r$ and $\mathbf{u}_{sf}$. However, in general this is not so, and the only case where they exactly vanish is when the filter represents a sharp spectral cutoff (actually this includes the case when $\mathcal{F}$ simply represents the ensemble mean). Leonard (1974) has analyzed the role and parametrization of these terms, in addition to more conventional parametrizations of the third term. However the 'Leonard terms' have been controversial in practice and hard to separate from parametrization issues. Most of the debate on the Leonard terms has been superseded by more recent parametrization issues, and we attempt to parametrize τ directly without particular regard to the differences from a Reynolds-stress term, as discussed in Mason (1994). Brown *et al.* (1994) give further comments on what we can, and cannot, expect from LES.

In terms of standard spectral descriptions of turbulence, the basic LES assumption is that we do not need to resolve the whole of the inertial subrange in order to simulate correctly the fluxes and variances which are concentrated in the larger 'energy-containing' scales. In addressing atmospheric (high-Re) problems, the advantage over Direct Simulation (at necessarily lower Re) is particularly clear where the turbulence length-scale is restricted, e.g. by stable stratification.

2.2 Subgrid terms

Despite the strong arguments for distinguishing conceptually between filters and numerical grids (hence our notation $_{sf}$ above), the word 'subgrid' is almost universally used for convenience. Henceforth we shall conform to that terminology, but remembering that the effective filter scale is not simply the grid, but essentially part of the subgrid model. The subscript $_r$ for 'resolved' is dropped from this point.

The subgrid model is the procedure which computes (estimates for) the subgrid stress τ_{ij} (as defined above) and subgrid scalar fluxes, $h_{n,i}$ for scalars q_n. In LES, it is common to use a relatively simple subgrid model, preferring to devote computational resources to explicit resolution of turbulence. The hope is that in a well-resolved simulation, the results should be relatively insensitive both to the actual resolution and to the details of the subgrid parametrization. See Mason (1994) for a fuller discussion and some examples of sensitivity tests.

The present model uses a *first-order closure* (in fact a variant of the Smagorinsky-Lilly model), with eddy-viscosity ν and eddy-diffusivity ν_h computed pointwise as detailed in §3.2.2, from the resolved strain-rate, resolved scalar gradients and certain prescribed length scales. It is the 'local and instantaneous' (essentially local equilibrium) assumptions which makes it a first-order closure.

In spite of the above arguments for simplicity of the subgrid model, the introduction of 'stochastic backscatter', whilst remaining broadly within the Smagorinsky-Lilly framework, has recently been shown to improve simulations of both neutral and stable boundary layers. This means that certain random terms are added to allow for the fact that, even after filtering, subgrid fluxes of momentum and scalars are not fully determined by the resolved fields. Beneficial effects have been shown on boundary layer structure, especially in marginally resolved regions close to a rigid surface (see Mason 1994, Brown *et al.* 1994). See §3.2.3 for details concerning the implementation of backscatter in the present code.

3 Analytical formulation

The analytical formulation is broadly similar to the previous papers already cited, but with further options. In particular quasi-Boussinesq ('anelastic') rather than incompressible Boussinesq approximations (see Annex B for derivation and discussion of energetics) may be used. The subgrid model (§3.2) has a slightly revised stability dependence, and stochastic backscatter may be used (§3.2.3). Moist thermodynamics and cloud physics options will be described in §3.8 and §3.9. A quick-look summary of the main equations and code structure is available separately.

3.1 Basic equations

Let p be pressure, θ_v virtual potential temperature, $\mathbf{u}$ the flow velocity, τ the subgrid stress, Ω the Earth's angular velocity relative to a local Cartesian coordinate system (x, y, z right-handed, with z vertical) and $g \simeq 9.81\mathrm{ms}^{-2}$ the downward gravitational acceleration (assumed constant). We shall define certain reference profiles (functions of z only) via subscript $_s$, and primes denote perturbations from these reference profiles (e.g. $p' \equiv p - p_s$). These reference profiles and other variables will be discussed further below. We define also a buoyancy perturbation

$$B' = g\theta'_v/\theta_s \tag{8}$$

where θ'_v is the perturbation in virtual potential temperature (see below).

The momentum equation, using Einstein's summation convention of summation over any index repeated once (in a product or similar expression), is then

$$\frac{Du_i}{Dt} = -\frac{\partial}{\partial x_i}(p'/\rho_s) + \delta_{i3}B' + \rho_s^{-1}\frac{\partial \tau_{ij}}{\partial x_j} - 2\varepsilon_{ijk}\Omega_j u_k \tag{9}$$

(where ε_{ijk} is the alternating pseudotensor) subject to the mass-flux divergence constraint

$$\frac{\partial}{\partial x_i}(\rho_s u_i) = 0 \tag{10}$$

In the incompressible Boussinesq case (10) *is* the incompressibility constraint; in the anelastic case it means that compression/expansion purely reflects vertical motion with respect to the basic-state pressure field. As usual,

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + u_i\frac{\partial}{\partial x_i} \tag{11}$$

All the velocities mentioned explicitly here are 'resolved'; we have now dropped the subscript $_r$ which we used briefly in §2. Subgrid motions enter purely through the parametrizations of τ_{ij} and $h_{n,i}$.

The equation for a conserved scalar q_n moving with the flow, but mixing through small-scale turbulence, is

$$\rho_s\frac{Dq_n}{Dt} = -\frac{\partial h_{n,i}}{\partial x_i} \tag{12}$$

where $h_{n,i}$ is the subgrid scalar flux of q_n. Equation (12) applies in particular to the total water mixing ratio variable ('Q') and to the temperature-like variable handled in the code using the 'TH' routines, unless further cloud physics (condensation, precipitation) are invoked. Details of the thermodynamic variables are given in §3.8.

Virtual potential temperature is used purely as a measure of buoyancy, and defined in general by

$$\theta_v \equiv T_v/(p/p_r)^{R/c_p} \equiv p/[R\rho(p/p_r)^{R/c_p}] \tag{13}$$

where p_r is a constant reference pressure for converting T to θ (normally 1000hPa), and R, c_p are respectively the gas constant and constant-pressure heat capacity, both for dry air. If calculating buoyancy directly from density, this must be performed at the reference pressure p_s: this is part of the anelastic approximation (see Annex B). That is why it is more convenient to express buoyancy in terms of variables like θ_v which remove the pressure-dependence.

In the model, ρ is not carried as a variable and so θ_v is calculated from

$$\theta'_v = \theta' + \theta_s((R_v - 1)r_T - R_v r_L - r_R) \tag{14}$$

where R_v is the ratio of the molecular weights of dry air and water vapour (taken to be 1.608) and r_T, r_L and r_R are the total water, liquid water and rain mixing ratios respectively.

NB at time of writing p_r in the standard version 1.4 was set to the surface reference pressure. Many users will prefer a fixed reference pressure; code updates written by H. Swann are available which introduce a separate variable PSFR ($= p_r$), with default value 1000hPa.

The set $\rho_s(z)$, $\theta_s(z)$ and $p_s(z)$ constitute a basic hydrostatically balanced, (but not necessarily isentropic) reference state which is calculated by the model from the reference potential temperature, θ_s (THREF), profile which is input by the user. The perturbation potential temperature is given by $\theta' = \theta - \theta_s$, and in the incompressible Boussinesq case $\theta_s \equiv \theta_0$ (THREF0), a constant.

For the Earth's rotation the basic LEM makes the f-plane approximation $\boldsymbol{\Omega} = (0, 0, \frac{1}{2}f)$. This strictly applies only at the Earth's poles. The terms in Ω_1, Ω_2 could easily be inserted in the routine UVWSRCE, but are significant only for 3-D modes on timescales of a few hours or more (see Mason and Thomson 1987).

3.1.1 Pressure calculation

To compute equation (9) we need to know p'; this is determined diagnostically by continuity (10).

Taking the Eulerian time-derivative of (10) and substituting for $\frac{\partial \mathbf{u}}{\partial t}$ from (9) gives

$$0 = \frac{\partial}{\partial t}\left(\frac{\partial}{\partial x_i}(\rho_s u_i)\right) = -\frac{\partial}{\partial x_i}\left(\rho_s \frac{\partial}{\partial x_i}(p'/\rho_s)\right) + \frac{\partial}{\partial x_i}(\rho_s s_i) \tag{15}$$

where

$$s_i = g\delta_{i3}\frac{\theta'_v}{\theta_s} + \rho_s^{-1}\frac{\partial \tau_{ij}}{\partial x_j} - u_j\frac{\partial u_i}{\partial x_j} - 2\varepsilon_{ijk}\Omega_j u_k \tag{16}$$

i.e. the non-pressure part of the velocity tendency. This leads to a Poisson-like elliptic equation for p'/ρ_s (it could be converted into a true Poisson equation by coordinate transformation, but this is not necessary for our purposes):

$$\frac{\partial}{\partial x_i}\left(\rho_s \frac{\partial}{\partial x_i}(p'/\rho_s)\right) = \frac{\partial}{\partial x_i}(\rho_s s_i) \tag{17}$$

The boundary conditions on this equation are (for the rigid-lid case):

$$0 = \frac{\partial w}{\partial t} = s_3 - \frac{\partial}{\partial z}(p'/\rho_s) \tag{18}$$

at top and bottom.

We solve this by Fourier-transforming in the horizontal and then, for each mode, solving the remaining ordinary differential equation in the vertical.

Note that p' does not necessarily average horizontally to zero, so that p_s is not necessarily the horizontal-mean pressure. The same applies to other variables (e.g. θ').

A different procedure is required when running the model in 1-D. For one-dimensional modes which vary only in the vertical, these equations might seem over-specified in general, since after suppression of horizontal variations the remaining ODE can be immediately integrated, to give

$$\left[\rho_s \frac{\partial}{\partial z}(\tilde{p'}/\rho_s)\right]_{\text{bottom}}^{\text{top}} = [\rho_s \tilde{s_3}]_{\text{bottom}}^{\text{top}} \tag{19}$$

(where tildes denote the relevant Fourier component). This apparent solvability condition is in fact plainly consistent with (18) for arbitrary specifications of s_i, and thus from an analytical point of view entirely irrelevant. However, it does imply that any attempt to compute this 1-D mode on the same basis as the others is severely ill-conditioned numerically (a '0/0' calculation). Therefore, we simply solve the hydrostatic equation, (18), rather than the Poisson equation used in 2D and 3D, (17). (NB this point is purely computational and involves no approximation.)

3.2 The subgrid model

We now give details of the subgrid model, which was described in general terms in §2.

3.2.1 The first-order closure

The subgrid stress, τ_{ij}, and scalar flux, $h_{n,i}$, are specified by

$$\tau_{ij} = \rho_s \nu S_{ij} \tag{20}$$

$$h_{n,i} = -\rho_s \nu_h \partial q_n / \partial x_i \tag{21}$$

where ν is the subgrid eddy-viscosity, ν_h the corresponding diffusivity for scalars and

$$S_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \tag{22}$$

The tensor S_{ij} is a symmetric traceless second-rank tensor in three dimensions, and therefore has $9 - 3 - 1 = 5$ independent components. By rotation to a local coordinate system we could eliminate 2, but this still leaves 3 independent components. In practice we shall base our calculation of ν on the modulus of the tensor S_{ij}. This simplification implicitly draws an analogy

between local strain and mean shear, and makes assumptions not only about local equilibrium but also about the combination of different components of strain.

As specified above $\tau_{ij} = \rho_s \nu S_{ij}$ is traceless, in the incompressible Boussinesq case. In effect we have subtracted the trace or 'isotropic' part $\delta_{ij}\tau_{kk} \sim \delta_{ij}\frac{1}{3}\rho_s \mathcal{F}(\mathbf{u}_{\rm sf}^2)$ from the subgrid stress. This part has no dynamical effect on the flow because it is entirely compensated for by pressure. The only consequence is that, with the present definition of τ, p' in the above equations should strictly be interpreted as $p' + \frac{1}{3}\rho_s \mathcal{F}(\mathbf{u}_{\rm sf}{}^2)$.

In the anelastic case $S_{ii} \neq 0$ in general because of the different continuity equation. In principle we should perhaps modify our strain, and hence viscosity calculation. However the effect on the strain-modulus of removing the trace of S_{ij} is $O(l_f/H)^2$ where l_f is the filter scale and H the scale of variation of ρ_s. This effect is negligible compared to the other uncertainties in parametrization.

3.2.2 Specification of viscosity

The classical Smagorinsky-Lilly approach, in inertial subranges, is to set

$$\nu = (c_s \Delta)^2 S = \lambda_0^2 S$$

where Δ is the grid spacing (often assumed equal in all directions in the early literature) and

$$S = ||S_{ij}||/\sqrt{2} = \left(\frac{1}{2} \sum_{i,j=1,3} S_{ij}^2 \right)^{1/2} \tag{23}$$

(so that for the special case of parallel laminar flow S is just the wind-shear) and c_s is traditionally regarded as a constant (~ 0.2). However Mason and Callen (1986) argued that c_s was not best regarded as a universal constant. Instead they focused on the length scale, λ_0, and argued that this was related to the filter scale l_f. Larger values of c_s were associated with an increase in the filter scale relative to the grid scale. This may not be an optimum use of computational resources, but is not in any sense fundamentally wrong. Small values of c_s may lead to rough fields and hence discretization errors. They argued therefore not for a single value of c_s, but in effect for a broad range of acceptable values.

If the grid is strongly anisotropic then, because we are dealing with 3-D turbulence, it is the coarsest rather than finest direction which matters most. Hence, to compute c_s diagnostically it is better to compare λ_0 with an arithmetic mean $\Delta_{AM} = (\Delta x + \Delta y + \Delta z)/3$ than with a geometric mean $\Delta_{GM} = (\Delta x \Delta y \Delta z)^{1/3}$ because the latter implies that the filter scale vanishes as any one of $\Delta x, \Delta y, \Delta z \to 0$. (Surprisingly, the latter definition has occasionally been used in the literature.) In practice, in most of our simulations $\Delta z \lesssim \Delta x, \Delta y$ and hence λ_0 may as well be compared with the horizontal grid spacing.

In practice, we usually use values of λ_0 which correspond to $c_s \simeq 0.23$. In large-eddy simulations in which the resolution is likely to be marginal (e.g. pure shear flow or statically stable conditions) it is common to use rather smaller values ($\simeq 0.15$). This accepts the possibility of some finite difference errors in order to ensure that the turbulence remains reasonably well resolved. Following Mason and Callen, we argue that this reduction is made solely to make best use of the available resources, rather than as a result of any fundamental association of different values of c_s with different flow types. In the limit of poor resolution, when $\Delta \gtrsim$ the depth of the turbulent layer (H), the model is no longer acts as a true LES (at least locally), but more like a 1-D model or NWP model in its representation of turbulence. In such circumstances, rough

consistency with the Unified model boundary-layer scheme would suggest using $\lambda_0 \sim [0.15H]$. In our subgrid model, the basic mixing length, λ_0, is input directly through variable RMLMAX in namelist SUBMODEL.

We make two important extensions to this classical approach, reducing the mixing length in proximity to a wall, and modifying it to take account of the dynamical effects of the pointwise Richardson-number (Ri_p) through the stability functions f_m and f_h. In the dry case, Ri_p is defined by

$$Ri_p = (g/\theta_s)(\partial\theta/\partial z)/S^2 \tag{24}$$

A discussion of the calculation of the moist Richardson number can be found in §3.8.2. Viscosity and diffusivity are prescribed through

$$\nu = \lambda^2 S f_m(Ri_p) \quad \text{and} \quad \nu_h = \lambda^2 S f_h(Ri_p) \tag{25}$$

where

$$\frac{1}{\lambda^2} = \frac{1}{\lambda_0^2} + \frac{1}{[k(z+z_0)]^2} \tag{26}$$

Here z_0 is the roughness length of the surface and $k \simeq 0.4$ is the von Kármán constant. Equation (26) provides a smooth match between the interior of the flow, where $\lambda = \lambda_0$, and the near-wall regime where the characteristic length scale must reduce and become proportional to distance from the surface. In some simulations it is appropriate to reduce the length scale similarly towards a rigid upper boundary or strong capping inversion. See M.K.MacVean for details.

In the code λ, λ_0 are called RNEUTML, RMLMAX respectively.

The extension of the basic model to include buoyancy effects is particularly important in stably-stratified conditions, which are common in the atmosphere. In a true inertial subrange, i.e. at high resolution, Ri_p should be small, but the LEM must be able to handle regions where it is not. Without a substantial reduction in ν as $Ri_p \to 0.25$, spurious laminarization may occur. A detailed specification of the stability functions is given below. Separate functional forms are specified for statically unstable and statically stable conditions. We ensure that these functions match in value at $Ri_p = 0$ but do not demand a match in gradient.

In *statically unstable* conditions ($Ri_p < 0$) we use

$$f_m = (1 - cRi_p)^{1/2} \tag{27}$$

$$f_h = a(1 - bRi_p)^{1/2} \tag{28}$$

where a, b, and c are empirical constants. However, note that the powers of one half in equations (27) and (28) are required to ensure finite values of f_m and f_h in the limit $Ri_p \to -\infty$. These powers may not be optimum at less negative values of Ri but an adequate match with observations is possible.

The value of a is given by the value of $1/Pr_N$ where Pr_N is the value of the Prandtl number in neutral conditions. We choose a value of $Pr_N = 0.7$, consistent with the Kansas observations and values obtained in large-eddy simulations (Mason and Derbyshire, 1990, Mason and Thomson, 1992). The constants b and c are set to 40 and 16 respectively to give a reasonable fit to surface layer observations.

For *statically stable* conditions $Ri_p > 0$ we use

$$f_m = \left(1 - \frac{Ri_p}{Ri_c}\right)^r (1 - hRi_p) \tag{29}$$

$$f_h = f\left(1 - \frac{Ri_p}{Ri_c}\right)^r (1 - gRi_p) \tag{30}$$

where Ri_c is the critical pointwise Richardson number above which we assume $f_m = f_h = 0$. f, g, h and r are constants. f must be equal to a to ensure a match to the unstable function at $Ri_p = 0$. There are no constraints on the form of these functions in the stable limit but we demand that the Prandtl number remain finite at Ri_c. The primary dependence upon Ri_p is contained in the factor $(1 - Ri_p/Ri_c)^r$ and best agreement with the observations is obtained with $r = 4$ for a range of values of the other parameters. The critical value Ri_c is taken as 0.25. We use the constants $g = 1.2$, $h = 0.0$, giving a value of Prandtl number of 1 in the limit $Ri_p = Ri_c$.

This completes the specification of the subgrid model. However in interpreting the subgrid model it is useful to consider the pointwise flux Richardson number Rf_p (which is also used to determine Ri_p when moisture is present, see §3.7.2). This is a measure of the work done against gravity by the subgrid buoyancy flux to the viscous energy drain from resolved to subgrid scales. If we define a scalar τ bearing the same relation to the tensor τ_{ij} as S does to S_{ij}, then (without backscatter)

$$\rho_s \epsilon = \frac{1}{2}\tau_{ij} S_{ij}(1 - Rf_p) = \frac{1}{2}\rho_s \nu S_{ij} S_{ij}(1 - Rf_p) = \rho_s \nu S^2 (1 - Rf_p) \tag{31}$$

In a local-equilibrium subgrid model Rf_p should definitely not exceed 1, and probably not exceed around 0.25. Note the simple relationship

$$Rf_p / Ri_p = \nu_h / \nu \equiv Pr^{-1} \tag{32}$$

where Pr is the subgrid Prandtl number.

In general, subgrid models with only a weak dependence on stability (or none at all) run the risk of incorrect laminarization. The allowance for stability via Ri_p in Meteorological Office LES models is a significant factor in our success in simulating stable boundary layers.

The subgrid model can be switched off by setting INOVISP=1. This is useful in testing changes to other parts of the code.

3.2.3 Backscatter

As suggested in §2.2, the stochastic backscatter model of Mason and Thomson (1992), extended to include buoyancy effects by Brown *et al.* (1994), effectively adds random subgrid terms to equations (9), (12).

Significant impact is expected in any simulation in which the turbulence is poorly resolved, whether due to computer limitations, stable stratification or proximity to the surface. The problem of turbulence initialization is also often made less troublesome due to the fluctuations created by the backscatter. Note however that there is a significant overhead in terms of core memory, and that the CPU time per timestep is increased by approximately 70%. Some reduction in timestep is also often observed due to the increased energy on small scales.

To clarify terminology, note the following relation between three positive quantities:

$$\begin{array}{ccccc} \text{VISCOUS DRAIN} & = & \text{DISSIPATION} & + & \text{BACKSCATTER} \\ \nu S^2(1-Rf_p) & = & \epsilon & + & \left(\frac{\partial}{\partial t}\right)_{\text{sct}} \frac{1}{2}\overline{u_i''^2} \end{array}$$

(Here the double primes indicate perturbation from the horizontal mean).

Part of the viscous drain of turbulence energy from resolved to subgrid scales is lost to dissipation, but part is scattered back to the resolved scales. Both viscous drain and backscatter are 'model quantities'; the *dissipation* is the term directly comparable with theory or experiments.

Only a very brief description of the stochastic subgrid model will be given here. It gives a backscatter of turbulent energy into the resolved scales relative to the turbulent KE dissipation ϵ:

$$\left(\frac{\partial}{\partial t}\right)_{\text{sct}} \frac{1}{2}\overline{u_i''^2} = C_B \left(\frac{\lambda_r}{\lambda_0}\right)^5 \epsilon \tag{33}$$

where λ_r is a subgrid length scale, equal to λ in neutral conditions, but reduced in stable conditions (see Brown *et al.* , 1994). Similarly, scalar variance is backscattered, relative to the variance-dissipation rates, ϵ_θ and ϵ_q

$$\left(\frac{\partial}{\partial t}\right)_{\text{sct}} \frac{1}{2}\overline{\theta''^2} = C_{B\theta} \left(\frac{\lambda_r}{\lambda_0}\right)^5 \epsilon_\theta \tag{34}$$

$$\left(\frac{\partial}{\partial t}\right)_{\text{sct}} \frac{1}{2}\overline{q''^2} = C_{Bq} \left(\frac{\lambda_r}{\lambda_0}\right)^5 \epsilon_q \tag{35}$$

In the code, backscatter is switched on by setting parameter IBSCATP=1. This automatically applies backscatter to all fields in use, and the setting of other parameters (e.g. IBSCATQP) must be changed manually if this is not desired. Note that, although backscatter has been shown to be beneficial in some boundary layer simulations, it has not yet been tested in non-Boussinesq simulations, and has only been used in a limited number of runs involving moisture. Also, backscatter should only be used in three dimensional simulations. The constants controlling the amount of backscatter are input via namelist SUBMODEL and recommended values are SCT($= C_B$)=1.4, SCTT($= C_{B\theta}$)=0.45, SCTQ($= C_{Bq}$)=0.45 .

3.2.4 Implementation in the code

The constants for the subgrid model are input via namelist SUBMODEL, with the FORTRAN variable name SUBB corresponding to the present notation b and so on. The following set of values is recommended. Note that the choice of an appropriate value for RMLMAX($= \lambda_0$) must be made by the user, noting the comments made earlier.

```
&SUBMODEL SUBB=40.,SUBC=16.,SUBG=1.20,SUBH=0.00,
 SUBP=1.,SUBQ=1.,SUBR=4.,ATH2_N=0.3,A2_N=0.23,PR_N=0.7,
 RIC=0.25,RMLMAX=??,SCT=1.4,SCTT=0.45,SCTQ=0.45 &END
```

Note that A2_N and ATH2_N, the neutral values of the stress-energy ratio and heat flux correlation coefficient, may be changed without affecting the resolved scales as they are only used in the diagnostic estimates of subgrid energy and scalar variance. For details see Brown *et al.* (1994).

SCT, SCTT and SCTQ are constants of the backscatter model, controlling the amount of backscatter of energy, TH and Q respectively (see next section). They are included in the namelist here for completeness, but note that there will be no backscatter if parameter IBSCATP=0, irrespective of the values of these constants.

3.3 Galilean transformation

Newtonian dynamics is invariant under the Galilean transformation

$$(\mathbf{x}, t, \mathbf{u}) \mapsto (\mathbf{x}', t', \mathbf{u}') = (\mathbf{x} - \mathbf{U}_{\rm Gal} t, t, \mathbf{u} - \mathbf{U}_{\rm Gal}) \tag{36}$$

where the constant $\mathbf{U}_{\rm Gal}$ is for present purposes assumed purely horizontal.

This invariance applies to all the analytical equations of §3 except for the Coriolis terms and the surface boundary conditions, which need to be evaluated from $\mathbf{u} = \mathbf{U}_{\rm Gal} + \mathbf{u}'$. (Coriolis forces are really non-Newtonian, reflecting a rotating frame, whilst surface boundary conditions obviously reflect the velocity relative to the Earth.) Note that horizontal periodicity is invariant under such Galilean transformations, because there are no 'walls'.

Spatial discretization is *not* Galilean-invariant: the grid is stationary in only one reference frame. By choosing

$$\mathbf{U}_{\rm Gal} = \frac{1}{2}[\min_{\rm domain}(\mathbf{u}) + \max_{\rm domain}(\mathbf{u})] \tag{37}$$

we can minimize $\max_{\rm domain}|\mathbf{u}'|$, and hence maximize the timestep permitted by the CFL restriction (see §4.2 for a discussion of CFL criteria).

The Galilean transformation can be switched off by setting IGALOFFP=1.

3.4 Lateral boundary conditions

The lateral boundary conditions are essentially periodic, but with an additional 'secular' (nonperiodic) term reflecting pressure gradients associated with weather systems on scales much longer than the horizontal domain extent. Formally we may write

$$\mathbf{u} = \mathbf{u}_{\rm periodic} + \mathbf{u}_{\rm sy} \tag{38}$$

$$p = p_{\rm periodic} + p_{\rm sy} \tag{39}$$

and so on, where $_{\rm sy}$ denotes the synoptic scales, making the primed quantities periodic. This is a bit like filtering in §2 but at the other end of the scale. In reality

$$\mathbf{u}_{\rm sy}(x + D, y, z) - \mathbf{u}_{\rm sy}(x, y, z) \sim D\partial\mathbf{u}_{\rm sy}/\partial x \tag{40}$$

where D is the domain size. Hence the systematic error in taking $\mathbf{u}_{\rm sy}$ constant is $O(D/l_{\rm sy})$. If $\mathbf{u}_{\rm sy}$ *is* constant, then we might as well absorb it into $\mathbf{u}$, and so in our code we need to insert explicitly only the synoptic pressure-gradient.

Similarly we may treat synoptic-scale temperature variation. This is of interest for two reasons: (i) it provides a simple and natural method of representing a 'stable' or 'unstable' airstream within our model (ii) it can represent a baroclinic zone which may support e.g. Eady waves. Of course baroclinicity is associated with a height-variation in the geostrophic wind, given by

$$\frac{\partial \mathbf{U}_g}{\partial z} = \frac{g}{f\theta_s}\mathbf{k} \times \nabla\theta_{\rm sy} \tag{41}$$

This thermal wind-balance, is implemented automatically in the code. If the baroclinic option is invoked through IBAROCLP=1 the geostrophic wind is permitted to vary linearly with height with specified surface value (UG0,VG0) and gradient (DUGDZ,DVGDZ). An advection term $-\mathbf{u}.\nabla\theta_{\rm sy}$, with $\nabla\theta_{\rm sy}$ calculated from (41) is then added to the equation for $\partial\theta/\partial t$ (=STH) in routine THSOURCE. If the baroclinic option is not so invoked, then DUGDZ and DVGDZ are ignored, and no such advection terms are added.

The LEM is not a quasi-geostrophic model and there is no instantaneous relation between geostrophic and actual wind, even when horizontally averaged. However it is recommended that initialization of mean winds, at least above the boundary layer, should normally be reasonably close to geostrophic, otherwise inertial oscillations will result (unless $f = 0$).

3.5 Gravity-wave damping

If the upper part of the model domain is stably-stratified, it is likely that gravity-waves will be reflected back. Such gravity-wave reflection may occur sometimes in reality, but in the model it will introduce an undesirable dependence on the domain depth, and usually we wish to have little or no reflection. In the quasi-hydrostatic case, an appropriate boundary condition in principle is $\tilde{p} = \tilde{w}N/|\mathbf{k}_h|$, where $\tilde{}$ denotes a horizontal Fourier transform, $\mathbf{k}_h$ the horizontal wavenumber and N the buoyancy frequency. A small section in code exists in POISSON (if IGWRBCP=1) to retrieve vertical velocity for this purpose, but such a calculation has not yet been fully implemented. Opinions differ as to whether such an approach can be satisfactory.

An alternative method is to implement a Newtonian damping layer in the code. This is achieved by setting the parameter IDAMPP=1 which will damp u, v, w, θ and the Q variables back towards their horizontal means at a rate given by

$$\frac{1}{\tau_{dmp}}\left[\exp\left(\frac{z - z_D}{H_D}\right) - 1\right] \tag{42}$$

for $z > z_D$ and zero for $z \leq z_D$. The values of $1/\tau_{dmp}$, z_D and H_D are input into the code through namelist DAMPNML as the variables DMPTIM, ZDMP and HDMP respectively.

Note that if the damping timescale is too fast, i.e. comparable to the time-step, a viscous instability could result. If the rate of increase of damping with height is too great then undesirable wave reflection can occur. As a general rule H_D should not be too much less than the depth of the damping layer, and the layer should cover at least ten vertical levels. Ideally, H_D should be greater than the vertical wavelength of typical waves.

3.6 Mean vertical motion

The basic LEM does not permit any mean vertical motion, after horizontal averaging over the domain. This is a direct consequence of the continuity equation and periodic boundary conditions. However boundary layers, and particularly capping stratocumulus layers, can be affected by vertical motions of magnitudes measured in cm/s and on the horizontal scales of weather systems. This effect indeed explains much of the difference between cyclonic and anticyclonic weather. Such scales cannot normally be computed explicitly in LES but may be added in 'by hand' without formal inconsistency. Specifically, for a general conserved variable q_x (e.g. θ), a term $-\overline{w}\,\partial\overline{q_x}/\partial z$ would be added to its tendency equation, where the overbar denotes horizontal mean.

Sources of such 'mean' vertical motion may be divided roughly into quasi-geostrophic processes (differential potential vorticity advection) and diabatic processes (basically 'Ekman pumping', giving systematic ascent in cyclonic and descent in anticyclonic situations). Ekman pumping velocities, in response to (the curl of) turbulent Reynolds stresses, are expected to rise from zero at the surface to a maximum near the boundary layer top and then slowly decline through the troposphere. There appear to be no exact theoretical constraints on $\overline{w}$ in the LEM except that it vanish at the surface. The usefulness of representing explicitly the link with Reynolds stresses, and hence for instance the likelihood of reduced Ekman pumping as turbulence lessens

overnight, obviously depends on the particular problem. There is no existing code for $\overline{w}$, but anyone interested in this should consult M.K.MacVean.

3.7 Surface boundary conditions

The surface boundary conditions are 'synthetic' ones derived from Monin-Obukhov stability functions; in fact versions of the Kansas (Businger-Dyer) functions (see Bull and Derbyshire 1990). So far we have not attempted to implement a full surface heat budget, but focused instead on the two extreme cases of (a) a specified surface heat-flux and (b) specified surface temperature, corresponding respectively to highly insulating or highly conducting surfaces. In each case efficiency of the solution method is an important consideration: the Monin-Obukhov relations are complex and may need to be solved by iterative methods.

The nub of the problem is to obtain u_* from the model horizontal velocities at the lowest grid-level in the model domain. First the absolute velocity relative to the surface is computed using the Galilean parameters. Then the Monin-Obukhov functions are inverted subject to the temperature or heat-flux boundary condition, as appropriate.

Surface fluxes can be switched off by INOSURFP=1.

3.7.1 Prescribed surface heat-flux

Selected by setting ITHBCP=1

If IFBCHGP=0, then constant surface fluxes are selected, with values set in namelist INPUT. SHFLX_SEN is the sensible heat flux (positive upwards) and SHFLX_LAT contains the fluxes of the Q-fields - note that all are energy fluxes i.e. units Wm^{-2}. These fluxes are combined in to a buoyancy flux (FBUOY), which includes both temperature and moisture effects (through coefficients CQ).

The stable, neutral and convective cases are handled differently. The neutral case may be solved trivially. The stable case (FBUOY<0) leads to a cubic equation for u_*, which is solved directly using Cardan's formula. This calculation is moderately complex, but well vectorized over J. The convective case (FBUOY> 0), however, requires iterative solution. To solve iteratively for each point separately would be expensive, so instead we use a look-up table, set up in subroutine SETLOOK. In calling LOOK to access the look-up table, the entry number is calculated from the logarithm of the wind-speed. Linear interpolation is used between table entries. For wind-speeds 'off the scale', extrapolation is made from the extreme values using either free-convection scaling (for low wind-speeds; so stress varies as wind-speed) or neutral scaling (for high wind-speeds; so u_* varies as wind-speed).

In this model it is possible to change the surface fluxes continuously by setting IFBCHGP=1. Times and corresponding values of sensible and latent heat fluxes are set through variables TIMHF, FSHFLX_SEN and FSHFLX_LAT in namelist TIMENML, and linear interpolation is used to obtain fluxes at intermediate times. In the convective case a new look-up table is required every time-step. However in most foreseen applications the prescribed heat-flux will change by only small amounts each time-step. The most attractive strategy is, therefore, to derive successive look-up tables by making small corrections to the previous ones. This is done by calls to the routine CHGLOOK.

CHGLOOK operates as follows. Using the new value of FBUOY, together with the old table of u_* (USTLOK), a temporary array VELNEW is calculated. This is compared with the (fixed) array VELLOK representing the target velocities of particular entries in the look-up table. The rate of change of log(windspeed) with respect to log(u_*) (at fixed FBUOY),

denoted by DVELUSTR, is estimated from the previous look-up table using a finite-difference approximation at each level. Then USTLOK is adjusted accordingly via the equation

```
USTLOK(IL)=USTLOK(IL)/(VELNEW(IL)/VELLOK(IL))**(1./DVELUSTR(IL))
```

The parameter DFBMAX limits the permissible absolute value of changes in FBUOY within one iteration. If the changes demanded exceed this, the change is divided up into smaller changes. In practice a value DFBMAX= 10^{-4} is found satisfactory. The CHGLOOK method, which resembles a Newton-Raphson iteration, converges very fast after a reasonable approximation has been obtained. By working in effect with logarithms we incur a slight computational penalty for the sake of peace of mind (simpler methods may allow the procedure to crash if FBUOY is reduced suddenly). As defined DVELUSTR is extremely well-behaved, varying smoothly between 1 (neutral conditions) and 7/4 or 2 (free convection). [Businger-Dyer give 7/4, whilst on theoretical grounds one would expect 2; this small difference is within experimental uncertainty.] This fact could be exploited to provide an efficient and effective iterative scheme if more complicated convective boundary conditions required solution on a gridpoint-by-gridpoint basis.

3.7.2 Prescribed surface temperature

Selected by setting ITHBCP=2

An alternative simple surface boundary condition is of prescribed surface temperature. This is no more difficult in principle than the prescribed flux but the code is not currently in the model as standard. However, M.K.MacVean has been working on this case and code is now available.

3.8 Moist thermodynamics

Moist thermodynamics are invoked by IUSEQP=1 and IPASQP=0, whether the anelastic or incompressible Boussinesq option (switched by IANELP) has been selected.

3.8.1 Moist thermodynamic variable

The LES code uses the two variables liquid water temperature perturbation (T_L') and the total water mixing ratio (r_T) to model moist processes. The former is described in more detail in Shutts (1991). The definition of the unperturbed variable is :

$$T_L = T + \frac{(gz - L_v r_L)}{c_p} \tag{43}$$

where T is the temperature, g is the acceleration due to gravity, z is the height, L_v is the latent heat of vaporisation, r_L is liquid water mixing ratio and c_p is the specific heat capacity of air at constant pressure. This definition ensures that T_L is conserved under evaporation and condensation. The parametrized form of the thermodynamic equation then becomes :

$$\frac{DT_L}{Dt} = \frac{1}{\rho_s}\frac{\partial}{\partial x_i}\left[\rho_s \nu_h \frac{\partial T_L}{\partial x_i}\right] - \frac{L_v S_p}{c_p} \tag{44}$$

where S_p is the sink of total water associated with precipitation. The model variable TH, which is the perturbation of T_L from its reference state (TLREF), can be written as

$$T_L' = T' - \frac{L_v r_L}{c_p} \tag{45}$$

and the source calculation for TH then includes a contribution from the vertical advection of the reference state.

[The user inputs the namelist variables THREF and THINIT (reference and initial) profiles as potential temperatures which are converted to T_L' assuming the atmosphere is initially unsaturated]

The vertical momentum equation still requires θ_v' (and therefore θ' and r_L) for the buoyancy term (see (9) and (13)), so a method must be found to calculate these from T_L' and r_T. Once r_L is known, T' follows easily from the definition of T_L', (45). For saturated air (r_L non-zero), r_L can be determined from r_T if $r_{\rm sat}(T,p)$, the saturation mixing ratio at temperature T and pressure p, is known. A Taylor's expansion about the reference state temperature, T_s (calculated from θ_s), gives a sufficiently accurate approximation for $r_{\rm sat}(T,p)$:

$$r_{\rm sat}(T,p) = r_{\rm sat}(T_s,p_s) + (T-T_s)\left(\frac{\partial r_{\rm sat}}{\partial T}\right)_{T_s,p_s} \tag{46}$$

Noting that $T - T_s = T' = T_L' + L_v r_L/c_p$ and substituting $r_{\rm sat}(T,p) = r_T - r_L$ in (46) gives the closed expression for r_L,

$$r_L = \frac{r_T(T,p) - \left(r_{\rm sat}(T_s,p_s) + \frac{\partial r_{\rm sat}}{\partial T}_{T_s,p_s} T_L'\right)}{1 + \frac{L_v}{c_p}\left(\frac{\partial r_{\rm sat}}{\partial T}\right)_{T_s,p_s}} \tag{47}$$

For convective work a closer approximation is sometimes found by using a more accurate reference temperature in the Taylor's expansion (see Shutts and Gray,1994).

3.8.2 Moist Richardson number

Calculating the pointwise moist Richardson number as a measure of the local stability is not necessarily a trivial extension of the vertical differences used in the dry case,

$$Ri_p = \frac{\frac{g}{\theta_s}(\theta_{k+1} - \theta_k)}{S^2 \Delta z},$$

essentially because the numerator stems from the parametrization of the buoyancy flux in the flux Richardson number (see later) and buoyancy is no longer necessarily conserved under vertical motion. Two alternative approaches are at present coded into the LES model and they are selected using the parameter MOISTRIP.

If MOISTRIP=1, Ri_p is calculated from the excess virtual potential temperature at model level $k+1$ of a parcel lifted from level k whilst conserving its moist static energy and r_T (and thus T_L). Thus, the pointwise Richardson number is given by

$$Ri_{p_k} = \frac{\frac{g}{\theta_s}\left(\theta_{vk+1} - \tilde{\theta}_{vk+1}\right)}{S^2 \Delta z} \tag{48}$$

where, as before (see §3.1),

$$\theta_{vk} = \theta_k + \theta_{sk}\left((R_v - 1)r_{Tk} - R_v r_{Lk}\right)$$

and a quantity with a tilde has been raised from k to $k+1$, conserving its T_L, and has still to be calculated. From the definition of T_L, (43), its conservation implies

$$\tilde{T}_{k+1} - \frac{L_v \tilde{r}_{Lk+1}}{c_p} + \frac{g z_{k+1}}{c_p} \equiv \tilde{T}_{Lk+1} = T_{Lk} = T'_{Lk} + \frac{g z_k}{c_p} + T_{sk}. \tag{49}$$

This is an expression in terms of known quantities and the as yet unknown $\tilde{T}_{k+1}$ and $\tilde{r}_{Lk+1}$ which are required for (48). Since r_T is conserved, r_L can be calculated from $\tilde{r}_{Lk+1} = r_{Tk} - r_{\rm sat}(\tilde{T}_{k+1}, p_{k+1})$ with this $r_{\rm sat}$ estimated from (46) and so in terms of $\tilde{T}_{k+1}$. Substituting this expression into (49) and rearranging for $\tilde{T}_{k+1}$ gives

$$\tilde{T}_{k+1} = T_{sk+1} + \frac{T'_{Lk} + \frac{L_v}{c_p}\left(r_{Tk} - r_{\rm sat}(T_{sk+1}, p_{k+1})\right) + T_{sk} - T_{sk+1} + \frac{g}{c_p}(z_k - z_{k+1})}{1 + \frac{L_v}{c_p}\left(\frac{\partial r_{\rm sat}}{\partial T}\right)_{k+1}}. \tag{50}$$

This is the value of $\tilde{T}_{k+1}$ finally used to calculate $\tilde{r}_{Lk+1}$ and these two can then be substituted into (48).

If MOISTRIP=2, Ri_p is explicitly determined from the pointwise flux Richardson number, Rf_p, which in turn is calculated (at the w-point between two θ-levels—see §5.2) from the subgrid buoyancy flux, h_B, implied by exchanging a small fraction by volume (ξ) of upper grid level air between the two vertically adjacent layers and mixing (as detailed in MacVean and Mason (1990)). By calculating Ri_p in this way, some provision is being made for the possibility of the release of energy by entrainment of dry air at cloud boundaries. The size of ξ is arbitrary, in that the final expression for Ri_p does not depend on it, but it must be small enough that the saturation of the layers remains unchanged by the specified mixing process—a condition found not to be restrictive (in the code ξ is set as EPS= 0.01, after MacVean and Mason's ϵ). Briefly, for constant grid spacing Δz, the buoyancy flux in the pointwise flux Richardson number, $Rf_p = h_B/(\tau_{ij}S_{ij})$, is calculated from the change in total potential energy during the mixing process, δP, as

$$h_B = -\frac{\nu_h \delta P}{2\xi \Delta z^3}$$

while the denominator in Rf_p is given from standard K-theory closure as $-\nu S^2/2$ (where the factor of two reflects the fact that we require a vertical average over the two layers involved in the mixing). Thus

$$Ri_p = Pr\, Rf_p = \frac{\nu}{\nu_h}\frac{\nu_h \delta P}{\nu \xi S^2 \Delta z^3} = \frac{\delta P}{\xi S^2 \Delta z^3}.$$

Finally, writing the buoyancy at model level k before and after mixing as B_k and $\tilde{B}_k$ respectively, gives $\delta P = \left((B_{k+1} - \tilde{B}_{k+1}) - (B_k - \tilde{B}_k)\right)\Delta z^2/2$ which in turn implies

$$Ri_p = \frac{(B_{k+1} - \tilde{B}_{k+1} - (B_k - \tilde{B}_k))/(2\xi)}{S^2 \Delta z}. \tag{51}$$

In the model, B is actually only the non-conservative part of the virtual potential temperature (ie. lacking the r_T term), and note that the apparent dependence on ξ in (51) is illusory.

In both cases the numerators of (48) and (51) (to a factor of g/θ_s) are calculated in MOISTRI and MOISTRI2, while the denominators, being the same as for the dry case, are calculated in SMAG.

3.9 Cloud microphysics

A bulk-water warm microphysics scheme has been incorporated into the model with the addition of one extra model variable: the rain mixing ratio, r_R. The microphysics is invoked by setting IRAINP=1 or IRAINP=2 for different schemes and with the moist thermodynamics also invoked (see §3.8).

In subroutine MICROPHYS the rain source term, S_R, is calculated. S_R is the sole source of r_R and sink of r_T and also appears in the prognostic equation for T_L, equation (44). S_R is the sum of three terms representing the following microphysical processes:

1. autoconversion of cloud droplets to rain,
2. accretion of cloud droplets onto rain,
3. evaporation of rain into dry air.

The prognostic equations for r_T and r_R are

$$\frac{Dr_T}{Dt} = -\frac{1}{\rho_s}\frac{\partial h_{T,i}}{\partial x_i} - S_R$$

$$\frac{Dr_R}{Dt} = -\frac{1}{\rho_s}\frac{\partial h_{R,i}}{\partial x_i} + \frac{1}{\rho_s}\frac{\partial}{\partial z}(\rho_s r_R w_R) + S_R$$

where w_R is the rain fall speed which depends on air density and rain mixing-ratio as described below.

Two sets of warm microphysics formulae have been tested in the LES model; those due to Kessler, 1974 (invoked by setting IRAINP=1) and Lee, 1989 (invoked by setting IRAINP=2). Kessler's formulae were calculated analytically by assuming a Marshall-Palmer (inverse exponential) size distribution of rain drops while Lee's formulae were obtained by regression methods using data from a detailed cloud model that simulates cloud and rain drops with 18 drop size categories. The resulting formulae for each of the processes and the fall velocity are as follows:

Warm Microphysics Parameterization
Formulae for S_R (kg/kg/s) and w_R

Process	I.Y.Lee 1989	E.Kessler 1974
Autoconversion	$1.44\, r_L^{2.36}$	$10^{-4}(r_L - \frac{0.001}{\rho})$ $(r_L > \frac{0.001}{\rho})$
Accretion	$7.58\, r_L^{1.029} r_R^{1.042}$	$2.20\,(\rho r_R)^{0.875} r_L\, e^{\frac{z}{20000}}$
Evaporation	$-0.0136\, r_R^{0.42} r_{def}^{0.746}$	$-0.0485\,(\rho r_R)^{0.65} r_{def}$
Fall speed, w_R	$33.2\rho^{-0.15} r_R^{0.25}$	$43.24\rho^{1.485} r_R^{0.237}$

where the saturation deficit, r_{def} =MAX$[(r_T - r_{sat}), 0]$.

More recently, Swann (1993) has extended these warm rain parametrizations to include the ice phase (see §7.2), however, it remains likely that both of these schemes will be inadequate for boundary layer cloud, as they cannot properly represent drizzle.

4 Numerical methods

The aims of the numerical scheme are (i) to preserve a reasonably high order of accuracy and (ii) to satisfy certain principles, such as stability and energy conservation where possible. The adiabatic dynamics are handled with essentially second-order accuracy in space and time, though the

parametrized viscous/diffusive terms are treated with only first-order accuracy in time. We do not use the formally very high-order accuracy of spectral computations, or indeed sophisticated time-schemes. Such methods come into their own when all the eddies are well-resolved, but that is not normally the case in LES. A separate note giving details of the numerical methods in the LEM will be available.

4.1 Advection schemes

In the present LEM there is a choice of advection schemes. The basic Piacsek-Williams (1970) scheme is linearly and quadratically conserving (and therefore energy conserving), but not, in general, positive definite. For certain problems with sharp scalar gradients, more sophisticated, and consequently computationally expensive, 'TVD' advection schemes are preferred. These schemes are positive definite and linearly, but not quadratically, conserving. The Piacsek-Williams scheme treats the advection terms as centred in time (together with a weak time-filter to suppress time-splitting), whilst the TVD schemes operate forward in time.

The three advection routines can be selected individually for scalar and momentum advection using the parameters ITVDSCALP and ITVDUVWP: 0 gives a centred difference scheme, and 1 and 2 the total variation diminishing (TVD) schemes of Leonard (1991) and van Leer (1974) respectively. Of the TVD schemes, Leonard's is less diffusive but computationally more expensive.

Since both the TVD schemes are designed for forward step codes, there are two ways of implementing them in the model which carries fields at two adjacent time-levels, and the same method need not be used for both scalar and momentum fields. Either they can be implemented as above as a single forward step from time-level t to $t+1$, or as a double forward step from $t-1$ to $t+1$. These choices are controlled by the parameters IFORUVWP and IFORSCALP, 1 giving the former (single forward step) and 0 the latter (quasi-centred). Not all permutations have been thoroughly tested with regard to computational stability but various comments can be made on some of them which should be borne in mind:

- there is probably little to be gained from using TVD schemes only on the momentum fields,
- when using 'quasi-centred' TVD on the scalars only, the Courant number stability criterion is stronger for the double timestep and it can suffer from time-splitting as there is no connection between adjacent time-levels,
- the van Leer scheme is only stable for all flow directions if the CFL number is less than roughly 0.4 (see §4.2).
- TVD on the scalars only has been found, in quite extreme conditions, to develop a slow-growing instability associated with wave activity,
- the fully forward step code (with IFORUVWP=IFORSCALP= 1) will suffer from the unconditional instability associated with the Coriolis acceleration, although this is very slow-growing with a timescale of the order of years, $\sim 2/(f^2\Delta t)$ (it is also possible to evaluate this term partially implicitly but this has not been included, yet!)

In summary, the choice of advection scheme will be a compromise between expense, diffusiveness and susceptibility to various spurious spatial oscillations (either stable in the case of centred difference or unstable albeit slow-growing for TVD schemes when implemented in a fully forward step code), depending on the application in question.

4.2 CFL criteria

So far we have assumed that the timestep was sufficiently small that differences between finite-difference and analytical time-derivatives could be neglected. In practice, however, the timestep must be constrained in order to maintain numerical stability, and this constraint will ensure that time truncation errors are very small (hence high order time differencing is not crucially important) In addition, it is worth noting that dissipative terms in the finite difference equations must be lagged in time (in a centred difference framework) for computational stability while, as noted in the previous section, a fully forward step code with non-zero Coriolis parameter will also be unconditionally unstable (albeit on a long timescale). These and other issues are covered in more detail in the LES numerical methods documentation.

From a practical point of view, both advective and viscous terms independently place constraints on the timestep. These can be expressed in terms of restrictions on the relevant Courant number. In one dimension, the advective Courant number (or CFL number, after Courant, Friedrichs and Levy) is simply

$$\mathrm{CFL} = \frac{u\Delta t}{\Delta x}.$$

In our case, u will be the flow velocity minus the Galilean velocity (see §3.3).

For stability, *all* grid-point velocities must remain within the stable region and so when placing a constraint on the timestep (Δt) we must use the largest CFL number. In generalizing to three dimensions we must also combine the 'worst cases' in each direction. Thus our largest Courant number is given by

$$\mathrm{CVEL} = \Delta t \left(\left(\frac{|u|}{\Delta x} \right)_{\max} + \left(\frac{|v|}{\Delta y} \right)_{\max} + \left(\frac{|w|}{\Delta z} \right)_{\max} \right) \tag{52}$$

Similarly, the 'worst case' viscous stability parameter is given by

$$\mathrm{CVIS} = \max_{\text{flowdomain}} \left[4\Delta t \, \nu_{\max} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2} \right) \right]$$

where $\nu_{\max}$ is the larger of ν and ν_h at each grid-point.

These two are then combined using the namelist-input constraints CVISMAX and CVELMAX to give

$$\mathrm{CFLNO} = \frac{\mathrm{CVIS}}{\mathrm{CVISMAX}} + \frac{\mathrm{CVEL}}{\mathrm{CVELMAX}}.$$

If CFLNO lies outside the range 1-TOL to 1+TOL (where TOL is a namelist-input tolerance), a variable DTMNEW is changed to the value necessary to reset CFLNO to 1. The actual timestep is changed immediately to DTMNEW if needs to be reduced but, if an increase is called for, the change is made gradually (by factors of 1+RINCMAX) over the succeeding timesteps. A maximum timestep (DTMMAX) is also imposed and the program terminates if the timestep becomes too small (less than DTMMIN). This procedure is intended to ensure that, once a run has settled down, the timestep only changes very infrequently.

The viscous-stability timestep-limit is complicated by the non-constant viscosity. With a constant viscosity, CVISMAX needs to be less than 1. However, with eddy-viscosity itself dependent on gradients of wind and scalars, this limit needs to be reduced by a factor of at least 2. It can be shown that when $Ri_p \rightarrow Ri_c$, conventional viscous-stability criteria are grossly inadequate (S.H.Derbyshire has details; this can be demonstrated in 1-D models). However this is not normally an issue in 3-D stable boundary layer simulations, because timesteps are typically limited by advection rather than viscous criteria. If in doubt, CVISMAX=0.2 is a standard value.

The Courant-number limit CVELMAX must be less than 1.0 if the centred difference or ULTIMATE QUICKEST advection scheme is used but less than 0.4 if van Leer's scheme is used in more than one dimension (see §4.1).

4.3 Time-smoothing

Since our basic equations (§3.1) are first-order in time, there is a potential problem in carrying 2 time-levels in the centred difference scheme. There is a risk that these levels could decouple (see the LES numerical methods documentation). To prevent this a weak time-smoother is applied continuously, using a small parameter TSMTH (=0.01 typically).

The time-smoothing operation, performed after each time-step can be summarized by

$$\begin{pmatrix} \mathrm{FLD}(t) \\ \mathrm{FLD}(t-1) \end{pmatrix} \mapsto \begin{pmatrix} 1-\mathrm{TSMTH} & \mathrm{TSMTH} \\ \mathrm{TSMTH} & 1-\mathrm{TSMTH} \end{pmatrix} \begin{pmatrix} \mathrm{FLD}(t) \\ \mathrm{FLD}(t-1) \end{pmatrix} \tag{53}$$

Time-smoothing has the undesirable effect of slightly violating energy conservation. It can be shown that this time-smoother tends to increase the energy of an oscillatory mode with frequency ω at a rate $\sim \mathrm{TSMTH}\,\omega^2\Delta t$. This however is only really significant if very precise checks on energy conservation are being made. By linearized analysis it can be shown that this corresponds to the decay of the computational 'negative energy' mode.

In some unpublished work, P.J.Smart looked at time-smoothing applied to the Coriolis equations for inertial oscillation. Based on an idea by N.Wood, he showed that a hybrid between the present time-smoother and another time-smoother, due to Asselin, which tends to *dissipate* energy, was much closer to being energy-conserving.

S.H.Derbyshire has more details of this and other aspects of time-smoothing. NB version 1.4 contains a slight 'time-slowing' factor on the clock time, which has now been shown to be erroneous and should be removed. The only effect of this was that diagnostics and other timed events were very slightly mistimed.

5 Code structure and software aspects

5.1 Parameters in the model

In its strict sense as a FORTRAN language element, a parameter is simply a number which is used by the compiler to set up array dimensions, etc. and consequently cannot change during the run. In the large-eddy model, parameters are also used as switches for the various options which have been described above (see Table 1). These switches must be parameters (rather than namelist variables) because they not only select the required sections of code but also set up further array size parameters (which have not been included below) according to the options selected so as to keep storage requirements to a minimum, and allow the CRAY computer to do the maximum amount of optimization.

5.2 Grid, reference profiles and timesteps

The grid is of Arakawa's type C, i.e. each velocity component is staggered in its own direction, see Fig. 1. (Note: in NWP a B-grid, with horizontal velocity components collocated, is sometimes used, and gives a slightly more accurate representation of Coriolis terms; an A grid would mean

Parameter	*Meaning*
IIP, JJP, KKP	Number of grid points in x, y and z directions. IIP and JJP should factor into 2s, 3s and 5s (except IIP,JJP=1 for 1 and 2-D)
JMINP, JMAXP	Range for J loops (=0,JJP+1 aids vectorisation)
IBSCATP	=1 implements backscatter
NQSCTP	Number of scalars to be backscattered
NBEGSCATP	Step number on which to start backscattering
ITVDUVWP	=1 for TVD advection on velocities.
ITVDSCALP	=1 for TVD advection on scalars.
IFORSCALP, IFORUVWP	=1 for forward stepping of scalars and velocities respectively. If both =1, code becomes fully forward step.
IBAROCLP	=1 for geostrophic wind shear.
IUSETHP	=1 to use theta.
IPASTHP	=1 theta behaves as passive scalar, *i.e.* no buoyancy.
IUSEQP	=1 to use Q-fields.
IPASQP	=0 for moist thermodynamics. Must be =1 if IUSEQP=0.
NQP	number of Q-fields. Must be at least 1 even if IUSEQP=0
IANELP	=1 for anelastic formulation, =0 for Boussinesq.
IDAMPP	=1 for Newtonian damping.
INOVISP	=1 for no Smagorinsky viscosity or diffusion.
IGALOFF	=1 for no Galilean transformations on velocities.
INOSURFP	=1 for no surface fluxes.
ITHBCP	=1 for prescribed surface heat-flux boundary condition, =2 for specified surface temperature boundary condition, (see §3.7)
IADJANELP	specifies how the reference state is balanced.
NTIMP	Number of times for time series.
NSERP	Number of time series.
ITSERP	=1 to enable time series.
MOISTRIP	specifies the scheme used to calculate Ri_p.
IMICROP	=1 for cloud microphysics.
IRAINP	=1 for Kessler, 2 for Lee microphysics.
NMETEORP	Number of hydrometeors.
IQBCP	lower bc for Q - currently inactive
IGWRBCP	1 gives upper radiation b.c. -NOT READY YET
IFBCHGP	value 1 gives varying heat flux lower bc

Table 1: Model parameters

all model variables stored on the same points). It consists of IIP×JJP×KKP grid-points with uniform horizontal grid-spacings DX and DY. Various constants related to the grid follow:

```
RNHPTS=1./REAL(IIP*JJP)

CX=1./DX        CY=1./DY
CX2=CX*CX            CY2=CY*CY       CXY=CX*CY
TCX=0.25/DX        TCY=0.25/DY
```

Grid constants for backscatter:

```
EG=2.0*(CX2+CY2)+RDZ(K)**2+RDZ(K+1)**2
BFM_P(K)=512.*RHON(K)/(3.*EG)     RMLMAX_PM5=1.0/RMLMAX**5
CY_ON_RHON(K)=CY/RHON(K)           RDZ_ON_RHON(K)=RDZ(K)/RHON(K)
CX_ON_RHON(K)=CX/RHON(K)           CX_ON_RHO(K)=CX/RHO(K)
CY_ON_RHO(K)=CY/RHO(K)
```

Others:

```
CZB(K)=(RHO(K-1)/RHON(K))/(DZ(K)*DZN(K))
                ! _use for diffusion onto p-level from below
CZA(K)=(RHO(K)/RHON(K))/(DZ(K)*DZN(K+1))
                ! _use for diffusion onto p-level from above
CZG(K)=-CZB(K)-CZA(K)
CZE(K)=(RHON(K+1)/RHO(K))/(DZ(K+1)*DZN(K+1))
                ! _use for diffusion onto w-level from above
CZF(K)=(RHON(K)/RHO(K))/(DZ(K)*DZN(K+1))
                ! _use for diffusion onto w-level from below
TZC1(K)=0.25*RDZ(K)*RHO(K-1)/RHON(K)
                ! _for advection onto p-level from below
TZC2(K)=0.25*RDZ(K)*RHO(K)/RHON(K)
                ! _for advection onto p-level from above
TZD1(K)=0.25*RDZN(K+1)*RHON(K)/RHO(K)
                ! _advection onto w-level (K) from below
TZD2(K)=0.25*RDZN(K+1)*RHON(K+1)/RHO(K)
                ! _advection onto w-level (K) from above
```

Expanded arrays (2D versions of 1D arrays, required for vectorization) are prefixed by E: e.g. `ECZA(J,K)=CZA(K)` .

Constants related to moisture are:

```
CQ(1)=RATIO_MOL_WTS-1.=0.608
THCONA=RATIO_MOL_WTS*THREFO       THCONB=THCONA-THREFO
```

The reference profiles for calculating r_L (see §3.8) are

```
DELTA_T=1.0
PREFRCP(K)=(PSF/PREFN(K))**R_ON_CP
TREF(K)=THREF(K)*(PREFN(K)/PSF)**R_ON_CP
```

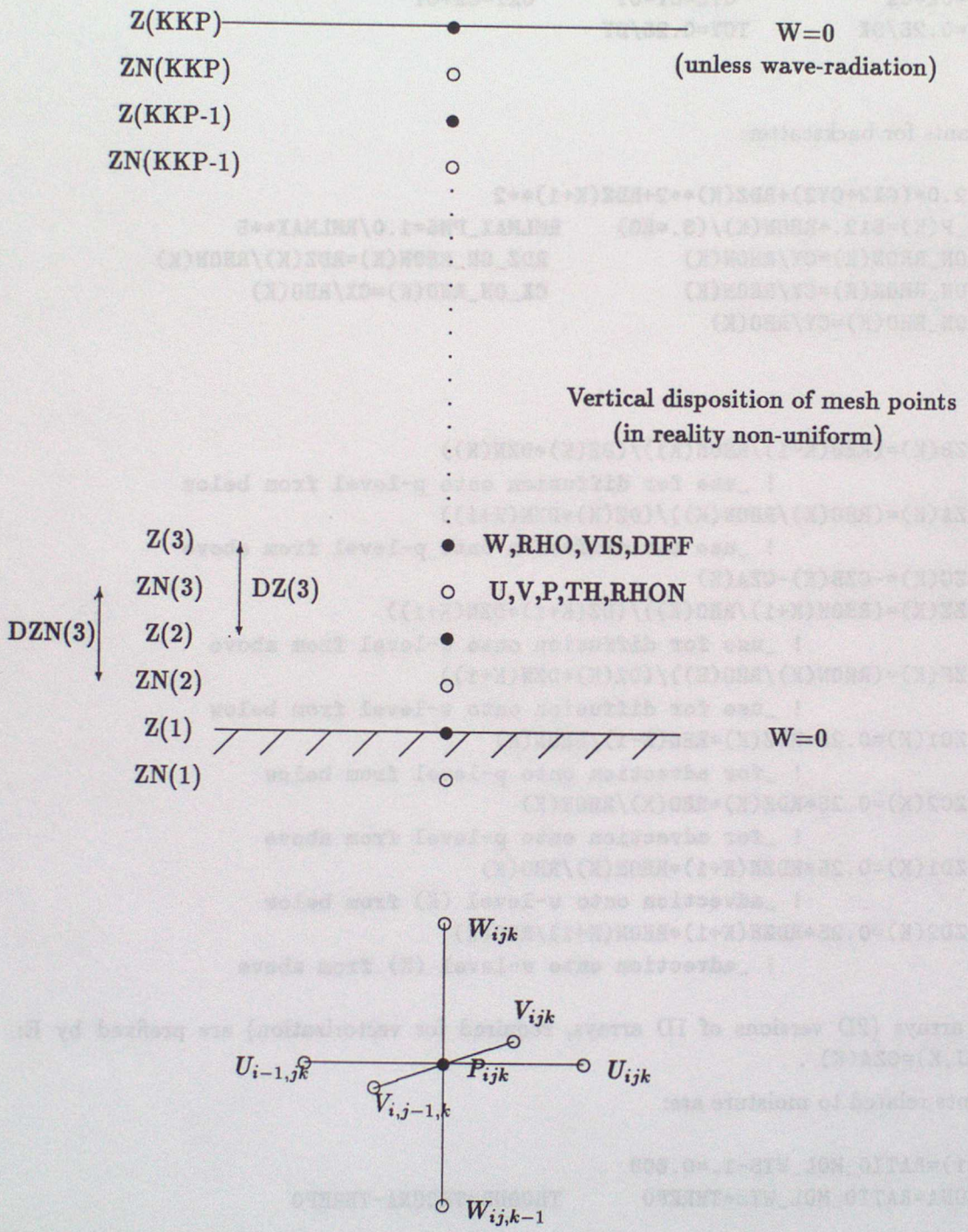


Figure 1: Staggered mesh (Arakawa-C) for the CRAY model

```
TLREF(K)=TREF(K) +(G/CP)*ZN(K)
TLREF(1)=TLREF(2)  ! as TREF - necessary for correct surface flux
DTLREF(K)=TLREF(K+1)-TLREF(K)
DTLREF(KKP)=0.
```

Timesteps: DTM is the timestep, entered initially by namelist and used to step fields. DTMNEW is the new target for DTM based on CFL and viscous stability limits. DTMOLD is the old value of DTM/(1+IFORSTEP), used *only* in PSTEP, because pressure-stepping of the old time-level occurs, slice by slice, at the beginning of the new one. Note that if the old timestep was forward, DTMOLD is only half the old DTM; this makes PSTEP independent of IFORSTEP. TIMINC is the forward increment of TIME itself.

```
DTM_X2=2.*DTM                 DTM_X4=4.*DTM
RDTM=1./DTM                   R_2DTM=1./DTM_X2
TSMTHC1=1.-TSMTH              TSMTHC2=2.*TSMTH-1.
```

Users should not need to worry too much about the mechanics of time-stepping unless they are trying a genuinely new time-scheme: for most purposes it is the source terms SF which are modified.

5.2.1 Variable names

All variables are in SI units (1mb $\mapsto$ 100Pa).

P means p'/ρ_s. PREF means p_s, THREF means θ_s, RHO means ρ_s (no fluctuating density variable is stored explicitly). Some of the basic state variables (PREF, RHO, Z) come in two arrays, with or without the suffix N (denoting storage on p-points); this notation is a vestige of the old IBM code. (Note however that the array DZN is obtained by differencing the array ZN and may be regarded as stored on w-points.) In the source routines TH means $\theta' = \theta - \theta_s$, or T'_L if moist thermodynamics are invoked. VIS and DIFF are the viscosities for momentum and heat, ν and ν_h, respectively. Note that all q-fields have viscosity ν_h.

5.3 Routines

Figure 2 shows schematically the progress of the model through the subroutines.

Routine BEGIN is always called first, to set up grid (CALL SET1D) and then either START (cold start) or RETRIEVE fields from dumped dataset.

Routine NNSTEPS organizes time-steps and is documented in Annex A.

The pressure calculation cannot be performed until the velocity source terms over the whole domain are calculated. This is obviously tricky to handle slice-by-slice. Basically on each slice the pressure source is accumulated in routine ADDTOPS. The elliptic equation for the whole pressure-field is solved in POISSON after the model has cycled through all slices for this timestep. The pressure-terms are applied to the velocity fields in PSTEP, after these fields are read back from packed storage. Thus during routine NNSTEPS those fields are packed in 'semi-stepped' form, i.e. not having been pressure-stepped and not yet satisfying the nondivergence constraint.

The routine EXXIT, with calling argument ISTOPU, is called when time or other flags indicate exiting from run, or other specific actions, may be required:

```
Argument ISTOPU governs flow of control as follows:
```

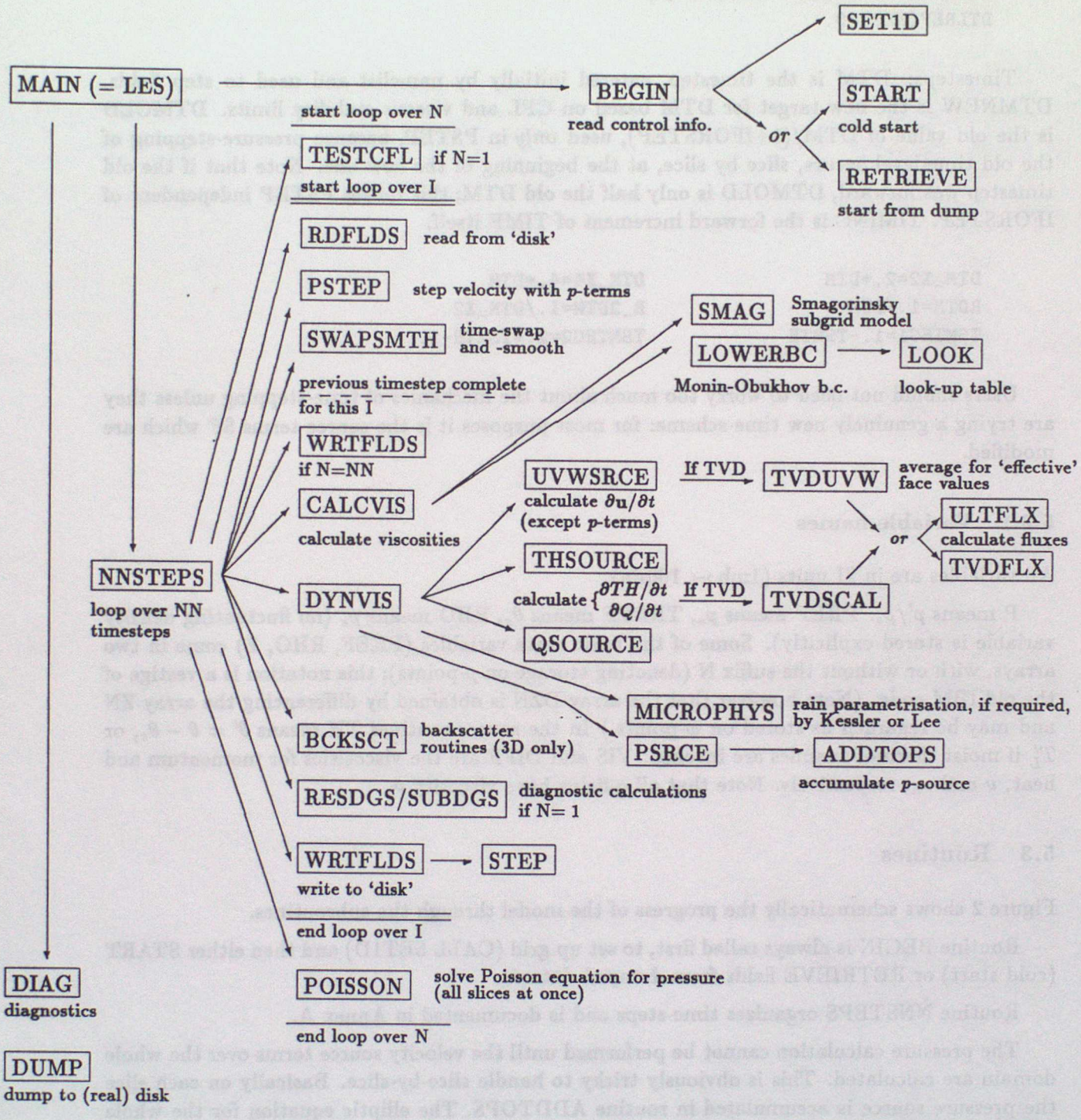


Figure 2: Outline structure of the model (version 1.4)
(arrows represent subroutine calls with RETURN implied)

```
  ---------------------------------------------------
-1 - writes dump and may write diagnostics; job continues
 0 - may write diagnostics; job continues
 NB...job continuation in above cases contingent on MSGFILE contents
      which may be altered interactively if you wish job to stop at
      the next dump.
 1 - after writing diags and dumping fields,
     terminate job in orderly way
 2 - bad error: terminate without dump
 3 - bad error: terminate in orderly way after writing diags
     and dumping fields (output may be useful for debugging)
     MSGFILE contents ensure that chain does not continue
```

5.3.1 1-D and 2-D routines

A feature of the current code which seems to be found useful is the easy switching between 1-D, 2-D and 3-D configurations. For 2-D, set IIP=1; for 1-D, set JJP=1 also. These automatically invoke specific 1-D or 2-D source routines.

In 1-D however remember that the parameter RMLMAX should be considered not in relation to the grid but in relation to the depth of the turbulent layer (see §3.2.2).

2-D turbulence is a slightly strange beast. In cases where it is driven by broadly 'convective' (sometimes 'Rayleigh-Taylor' instabilities), considerable insight can be obtained into the behaviour of the 3-D atmosphere. However 2-D shear flow simulations should be treated with considerable caution, because even the sign of mean momentum transport can differ from the 3-D result.

5.3.2 List of routines

```
SUBROUTINES ETC. IN ORDER OF OCCURRENCE, BY SECTION:
---------------------------------------------------

    (A)   EXXIT, BLOCKDATA, BEGIN, SETSG,

          SET1D, SETANEL, SETPREF, SET2D,

    (B)   START, RETRIEVE, DUMP, TDUMP, RDFLDS, RESETBAS, CVELGAL

          WRTFLDS, STEPGAL, STEP, STEPW, HOLD, RDP, WRTP, RDW

    (C)   NNSTEPS, SETINDEX, REINDEX, SWAPSMTH, CALCVIS,

          DYNVIS, SMAG, SMAG2D, SMAG1D, SETFRI, LOWERBC,

          UVWSRCE, UVW2D, UVW1D, THSOURCE, TH2D, TH1D,

          QSOURCE, Q2D, Q1D, PSRCE, DIVERR, POISSON,

          FFANAL, FFSYNTH, FOURIER, SETFILT, PSTEP, INITPS,
```

```
      ADDTOPS, RDPS, WRAP, SETO, EQUATE, REQUATE,

      SETLOOK, LOOK, CHGLOOK, TESTCFL, INITCFL

(D)   TESTTIME, SETFLUX, DIAG, WRTFB, RDFB,

      ALLOCDGS, ALLOCATE, ALLOCQS,

      INDGNEW, INDGAV, RESDGS, SUBDGS, AVDG,

      TIMSER, INTIMSER, CHKPARMS, PRINT,

      MOISTRI, MOISTRI2, ENTOTHO, ENTOTH1, QSATURATION,

      TVDSCAL, TVDFLX, ULTFLX, TVDUVW,

      TVDSC2D, TVDF2D, ULTF2D, TVDUVW2D,

(G)   SETRAN, GETRAN, BCKSCT, BCTSCT, BCQSCT,

(H)   ENTOTH2, MPHYS
```

5.4 Namelists

There are a number of model variables which can be set, after compilation, through the namelists given below.

NAMELIST CNTRL	
ISTART	=1 if a set-up run
ITDUMP	=1 for a tape dump (for use on Cray) - not ready yet
NN	Number of steps between diagnostic evaluations (> 1)
NNDIAG	Number of diagnostic evaluations between dumps
NNDUMP	Number of dumps for this run

Note that the total number of steps in a run is thus NN*NNDIAG*NNDUMP.

NAMELIST TIMENML	
TIMCDG, TIMPDG, TIMRDG	Array of times at which to calculate, print or reset the diagnostics.
TIMDUMP	Array of times to dump fields to disk.
TIMHALT	Array of times to stop job (for chain runs).
TIMHF	Array of times for time-varying surface fluxes.
FSHFLX_SEN, FSHFLX_LAT	Time-varying surface sensible or latent heat flux.
NTMCDG, NTMPDG NTMRDG	Number of diagnostic calculating, printing or resetting times.
NTMDUM	Number of field dumping times.
NTMHALT	Number of halting times.
ITIMCDG, ITIMPDG, ITIMRDG, ITIMDUM, ITIMHALT	Flags for the above options.
NSTEPMAX	Maximum number of timesteps in this job
IPRTDG	Controls the diagnostic output (see below)
NTMHF	Number of surface flux times.

Diagnostic output is controlled by IPRTDG as follows: 0 for output only at times given by TIMPDG and TIMRDG, 1 for output after every job, 2 for output at set times (possibly many times within one job) to a single file, 3 for output at set times (possibly many times in one job) to separate files.

The NAMELIST JOBINFO contains character strings which can be used to name the files connected with the job.

NAMELIST INPUT	
NRUN	Run number
NDATE	A number (eq. the date) to help identify the run.
IUTARG,IVTARG, ITHTARG,IQTARG	=1 to hold fields to TARGET fields.
TIME	Integration time.
Z0	Surface roughness length for momentum
Z0TH	Surface roughness length for temperature
Z0Q	Surface roughness length for 'Q' variables
PSF	Surface pressure
SHFLX_SEN	Surface flux of $c_p\theta$
SSHFLX_LAT	Surface flux of $L_v r_T$
USHRTARG, VSHRTARG, RNSQTARG	Constant shear and N^2 for target fields (see routine HOLD)

NAMELIST GRID	
NSMTH	Number of 1-2-1 smoothings of grid heights.
HGD, KGD	Arrays of corresponding heights and grid levels.
ZZTOP	Height of domain top.

NAMELIST THPROF	
ZNREF_READ,THREF_READ	Arrays of corresponding heights and θ's for θ_s.
THREF0	θ_0
ZNINIT_READ,THINIT_READ	Arrays of corresponding heights and θ's for the initial profile.

The NAMELIST SUBMODEL is discussed in §3.2

NAMELIST DIAGNOST	
IHORSL	=1 for horizontal slices to be output.
NKTEST,KTEST	Number of horizontal slices and their grid-levels.
IDGU,IDGV,IDGW,IDGTH,IDGQ,	Individual field output control variables
IDGCL,IDGPD,IDGRR,IDGQKG	(see below)

For individual fields the control-variables IDGU etc. determine what if any fields are output. 0 means no fields output, 1 means vertical only, 2 vertical and horizontal, 3 horizontal only. If IHORSL = 0 there are no horizontal fields output, irrespective of the IDGs. The Vax reading-plotting routines are set up to cope automatically with such variations in storage format.

NAMELIST DYNAMICS	
FCORIOL	The Coriolis parameter
UG0,VG0	The components of the geostrophic wind.
DUGDZ,DVGDZ	The rate of change of geostrophic wind components with height.

NAMELIST NUMERICS	
DTM	Timestep.
TSMTH	Time-smoothing factor.
DXX,DYY	Horizontal grid spacings (constant).
RINCMAX	Incremental factor for increasing timestep.
DTMMAX,DTMMIN	Maximum and minimum values for the timestep.
CVISMAX,CVELMAX	Bounds for viscous and advective CFL numbers.
TOL	Tolerance for CFL numbers (see §4.2)

NAMELIST PHYSICS	
VK	von Karman's constant
ALPHAH,BETAM,GAMMAM, BETAH and GAMMAH	Monin-Obukhov coefficients
DFBMAX	Maximum change in buoyancy-flux between iterations in CHGBUOY
CQ	Array of coefficients for 'Q'-field contributions to buoyancy [the model overwrites the first element with CQ(1)=$R_v - 1$, see §3.1]

The NAMELIST DAMPNML is described in §3.5

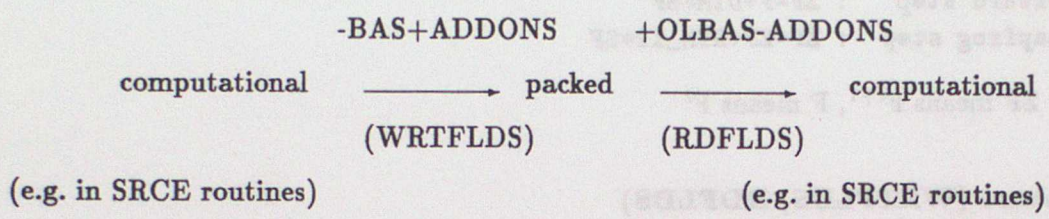
With the exceptions of CNTRL, TIMENML and JOBINFO, all of the above namelists are only read in a set-up job (ISTART=1). One other namelist, OVRIDE1, allows some of the namelist variables to be changed after the set-up.

5.5 BAS and BAR arrays

Associated with each of the fields U, V, TH, Q are two sets of 1-D arrays (over K), which must be carefully distinguished. The BAR arrays are horizontal means. The BAS arrays are somewhat arbitrary profiles subtracted and added to save rounding error which might be significant if packed slices are being kept as 32 rather than 64-bit. Neither of these has anything to do with the anelastic reference state profiles ρ_s, θ_s, p_s, which are represented by RHO, RHON, THREF, PREF, PREFN. The prefix 'OL' for 'old' BAS arrays should be considered essentially independent of prefix 'Z' used for old time-levels.

Think of the fields as existing in three possible states: (i) true, (ii) computation and (iii) packed. *True* fields mean the actual values you might measure or compare with theory. *Computation* values used in the source routines (UVWSRCE, THSOURCE etc.) differ because

Galilean transformations are applied to the horizontal velocities (to reduce absolute velocities for technical reasons). Similarly an anelastic reference state is subtracted from the θ-field so that in the source routines TH means $\theta' = \theta - \theta_{\rm ref}$, or similarly T'_L. *Packed* fields are for storage, not calculation and have had things (BAS arrays) subtracted from them to minimize (roughly) their absolute values. BAS arrays are approximately but not exactly the horizontal means of the true fields — they are somewhat arbitrary *unless you change them while the fields are still packed, in which case those changes will alter the fields when read back.*



Here ADDONS means things like UGAL and THREF. Note that in the computation (source) routines, terms with UGAL need to be added onto the Coriolis force to allow for the difference between computation and true values. Similarly terms in THREF appear in THSOURCE for the same reason. Note that the OLBAS used in RDFLDS may differ from the BAS used in WRTFLDS if it has been changed in HOLD to tweak the mean profiles.

RESETBAS is called after fields are safely packed away and takes the form

```
OLBAS=BAS     ! _ensure OLBAS for RDFLDS matches previous BAS used
              !   for WRTFLDS (unless changed later by HOLD)
BAS=BAR       ! _reset slightly arbitrarily
OLBAR=BAR     ! OLBAR, unlike BAS, will be time-swapped and -smoothed
```

This is for both time-levels (i.e. with and without the 'Z'). For the purposes of BARs and BASs the two time-levels are treated in the same way, so e.g. you have both OLBAS and OLZBAS arrays. The only difference is that in HOLD only the OLZ's are changed (changing the OLs would have no effect in the end as that time-level is effectively dead).

In principle BARs are completely separate. These are supposed to be *exactly* the horizontal means of the true fields and hence are time-smoothed like the fields. This is so that diagnostics can be calculated correctly. The only connection with the BAS arrays is that for convenience BAR is used as a sort of rough guess for BAS (as above). Also of course when HOLD modifies OLBAS, e.g. to add on a temperature increment matching to some measured profile, it has to modify BAR too for consistency.

Remember: OLBARS are for diagnostics, BAS for packing, OLBAS for expanding. BARs are purely for accumulation and not to be used for calculation except in RESETBAS — otherwise, particularly in START, you may generate some surprisingly large numbers.

5.6 Time levels

In effect, the model cycles through 4 phases:

- **Computation phase (source routines)**
  In NNSTEPS, ZF means F^{t-1}, F means F^t, where t indicates the latest time-level solution currently available; SF always means dF/dt. With a leapfrog step these arrays are passed straightforwardly through CALCVIS and DYNVIS to the source routines and SF is calculated using the ZF (old time-level) fields for some terms and the F (current time-level)

fields for others, in the usual manner. However with a *forward* step, CALCVIS and DYNVIS are called only with the F (current) fields, so that the source routines effectively use only the current time-level. Note that only SF (not F or ZF) is updated directly in the source routines.

- **Stepping phase**
  Basically ZF is the one updated.

```
Forward step    : ZF=F+DTM*SF
Leapfrog step   : ZF=ZF+DTM_X2*SF
```

  Hence now ZF means F^{t+1}, F means F^t

- **Packing phase (WRTFLDS, RDFLDS)**
  Still ZF means F^{t+1}, F means F^t

- **Swapping phase (SWAPSMTH)**
  SWAPSMTH swaps (and slightly time-smooths) F and ZF so that afterwards ZF means F^t, F means F^{t+1} and we are ready for another ...

- **Computation phase (source routines)**
  The cycle repeats, etc., etc.

5.7 Updating and running the model

The basic model is maintained through the Cray Update facility. Any external user making modifications to the Fortran should consult Bracknell contacts as to whether these can be recorded in Update form, to ease incorporation of new updates.

There are three Cray-specific routines (PACK, EXPAND, USSCTI); updates exist to convert these into standard Fortran for porting to workstations.

The model is run on various machines: on the Cray-YMP and on workstations (including DEC-αs). It has also been ported to a Unix-based Sun-workstation at Sheffield by A.Maguire, who has also converted the graphics to run on that system.

6 Output

Diagnostic files are produced by routine DIAG. This is called at the end of each run of the model (or each job of a chain) i.e. when all timesteps have been performed or when the simulation reaches one of the preset times in namelist TIMENML. Output may be controlled in relation to jobs using the namelist variable IPRTDG, as described in §4.

Each diagnostics file contains the following

- Instantaneous fields. Vertical slices of selected fields, and horizontal slices as selected by the user in namelist DIAGNOST are dumped.
- Time series. The model has NSERP time series, each with NTIMP bins. Information is written to the series every NN steps, and the first series contains the times at which this is done. Additional time series can be added in routine TIMSER.

- Time averaged statistics. On the first step of each NN, the model calculates horizontally averaged statistics of turbulence quantities in routines RESDGS (resolved quantities) and SUBDGS (subgrid quantities). A running time average is calculated in routine AVDG. Averages are reset at the times TIMRDG specified in namelist TIMENML.

  The user may add additional diagnostics by adding new pointers in COMMON/DGCNT/, allocating values to these pointers by calls to ALLOCATE from ALLOCDGS, and calculating the new diagnostic in RESDGS or SUBDGS as appropriate. Note that it may be necessary to increase parameter NDGSP (the maximum number of diagnostics).

  Note that a new averaging period will be required if diagnostics which were not in the original run are required. Another approach, used by some large-eddy groups, is to store complete model fields at a number of times within the averaging period, thus enabling any time-averaged diagnostic to be calculated at a later stage. However this is expensive in terms of disk space requirements, and is not used as standard with the present model.

Workstation graphics routines available include:

1. A program GRAPH14 (Hobson/Brown/Derbyshire) to profile statistics — used particularly for boundary layer applications.
2. A fields-plotting program SLICE14 (Gray).
3. Various pv-Wave procedures (MacVean/Brown/Dharssi/Brown).

7 Recent developments and conclusions

7.1 Ice Microphysics

The representation of ice phase microphysics can be crucial in the modelling of deep convection in order to simulate the effect that ice has on the transport of heat and moisture and also the effect that ice cloud has on radiation. The three-phase microphysics scheme developed for the LEM uses the parametrization of precipitating water particles based on Lin *et al.* (1983). Lin however does not detail the representation of cloud. The ice and liquid cloud processes are parametrized using a scheme based on Cotton *et al.* (1986). The resulting scheme is the basis of most microphysics schemes used in 3-D cloud-resolving models (see Flatau, Tripoli and Cotton (1989), McCumber, Tao and Simpson (1991), etc.) There are over 30 processes contributing to the conversion rates between water categories for a six variable *bulk water* scheme which are described in detail in Swann (1993).

The microphysical processes in a cloud are those processes leading to the formation, growth and depletion of the water particles. These particles can be liquid, ice or a combination of both and may have an irregular or regular shape. The model's scheme divides these particles into several categories commonly used in bulk water schemes: liquid cloud droplets(q_C), rain drops (q_R), ice crystals (q_I), snow crystals or aggregates (q_S), and graupel or hail (q_G). There is a further variable (N_I) which represents the number concentration of ice crystals. The mass mixing ratio *wrt* air of each water category is represented by a model variable. The size spectra of the precipitating particles (rain, snow and graupel) is assumed to be an inverse exponential distribution dependent on their mass mixing ratio. Cloud particles are assumed to form a monodisperse size spectrum of homogeneous droplets or crystals. The cloud droplets are assumed to have a constant number concentration which will depend on type of air mass, continental or maritime. The number concentration of ice crystals is modelled as a separate variable so that the various ice nucleation process can be parametrized.

7.2 Longwave radiation parametrization

The longwave parametrization scheme implemented within the cloud model is based on the scheme used by the Meteorological Office Unified Model, as described by Ingrams (1993) and Slingo and Wilderspin (1986). This parametrization scheme solves the radiative transfer problem for each model column in terms of a vertically upward and a vertically downward longwave flux. Such an approach is only strictly valid for a plane parallel atmosphere. For gaseous absorption, variations in the vertical are much larger than variation in the horizontal and thus the 'plane parallel' approximation will give good results. However clouds can show significant horizontal variation and it is not clear how valid the 'plane-parallel' approximation remains in this case.

The longwave radiation scheme includes absorption by water vapour, carbon dioxide, ozone, cloud water and cloud ice. Only the effects of absorption and emission are included and scattering is ignored. However, it is generally believed that scattering is not important for longwave radiation. For gaseous absorption a lookup table is used to obtain transmissivities from the absorber pathlength. This pathlength is scaled to include the effects of collision and doppler broadening of the absorption lines.

A model grid box is assumed to either be completely filled by cloud or to contain no cloud (partial cloud cover within a grid box is not allowed). For cloud water, the emissivity, ϵ_L, is calculated from an equation derived by Stephens (1984) using results from Mie theory. Only the cloud water path (cloud water content $\times$ layer width) is required. Thus the emissivity is given by

$$\epsilon_L = 1 - \exp\left(-\kappa_L \, CWP\right), \tag{54}$$

where CWP is the cloud water path and $\kappa_L = 130\, m^2 kg^{-1}$.

The longwave scheme treats absorption by cloud ice in a similar way to cloud water. The absorption properties of ice are assumed to depend only on the ice water content and the difference in the absorption properties of ice and water is accounted for by a scaling constant. Thus $\kappa_{\mathrm{Ice}} = 65\, m^2 kg^{-1}$. However, both experiment and theory show that this is an over-simplification and that the effective radius of the ice particles as well as the ice water content is important. Francis *et al.* (1994) measured values of κ_{Ice} ranging from $3\, m^2 kg^{-1}$ to $144\, m^2 kg^{-1}$. The reason for this large variation is that κ_{Ice} has an inverse dependence on the ice crystal effective radius, as predicted by simple theory. The value of κ_{Ice} used in the Unified model corresponds to an ice crystal effective radius of about $20\, \mu m$. The longwave scheme currently ignores the effect of rain, snow and graupel.

For more details, contact I.Dharssi at the JCMM, Reading University. Also, M.K.MacVean has implemented a rather simpler (and cheaper) longwave radiation scheme.

7.3 Other developments

Individual users have made further developments to the code, and should be consulted by anyone intending to use the model in the following areas:

- Nudging of model variables towards experiment (Derbyshire)
- Particle dispersion (Kemp)
- Development of improved advection schemes (Lock)
- Development of diffusion schemes less restrictive on model timestep (Hobson)
- Stretched grid in horizontal (Gray)

We would like to acknowledge the help of our many LEM colleagues in relation to both the model and the documentation.

Annex A : Routine NNSTEPS

The penalty (in terms of loss of transparency) of using slices rather than explicitly three-dimensional fields is incurred principally in the routine NNSTEPS. Anyone who wishes to modify this will require a good knowledge of the structure of the code, whereas the source routines (UVWSRCE, THSOURCE etc.) are designed to be relatively close to the analytical formulation, so that e.g. forcing functions could be introduced fairly straightforwardly.

The philosophy in NNSTEPS is to cycle through the vertical slices, storing only a few at a time in COMMON/LARGE/, and computing from these the 'source terms' $\partial \mathbf{u}/\partial t$, $\partial \theta/\partial t$ and so on (these terms are called SU, SV, STH etc. in the code). In writing back to the packed fields (or external storage) via WRTFLDS, note that the values held in /LARGE/ are not themselves changed (because some of them may still be needed in the current timestep). (The routine STEP, called by WRTFLDS, uses temporary arrays to avoid changing these slices; the non-packed slices U,V,W,TH,Q are *not themselves incremented*).

To avoid copying one array into another, the cycling is accomplished by reindexing. The indices are set up in SETINDEX and, when the current I is advanced by one, we call REINDEX, which reindexes the pointers. For example, the slice previously referred to as IP1 (i.e. slice I+1) becomes I0 (i.e. slice I). The slice previously referred to as IM2 (i.e. I−2) is no longer required and is overwritten. The details of which slices are held in COMMON/LARGE/ and the reindexing depends on whether backscatter is used, as explained below. VIS is viscosity, DIFF is diffusivity for scalar, DIS is drain of momentum, DIST is drain of scalar variance.

A.1 NNSTEPS without backscatter

- To step fields on slice I requires viscosities on slices I±1. This in turn requires fields on levels I±2 to have been read. This is done as follows.
- slice IFIRSTP= −1: read slices -1,0,1; compute VIS, DIFF on slice 0
- slice 0: read slice 2; compute VIS, DIFF on slice 1
- slice I (I=1,IIP): read slice I+2; compute VIS, DIFF on slice I+1; step fields on slice I; write new fields on slice I
- The subroutine RDFLDS is written so that CALL RDFLDS(I,...) is treated as CALL RDFLDS(I+IIP,...) for I < 1, and CALL RDFLDS(I-IIP,...) for I > IIP.
- Storage requirements: 5 slices of fields (I-2,I-1,I,I+1,I+2), 3 slices of viscosities (I-1,I,I+1).

A.2 NNSTEPS with backscatter

- To step fields on slice I requires viscosities on slices I±1 for the viscous terms (as before) AND dissipations on slices I±2 for the backscatter. To calculate these dissipations we need to have calculated viscosities on slices I±2, and therefore need to have read fields on slices I±3. This is done as follows.
- slice IFIRSTP= −3: read slices -3,-2,-1,0; compute VIS, DIFF, DIS, DIST on slice -1 (required for dissipations on I-2 when I= 1)
- slice -2: read slice 1; compute VIS, DIFF, DIS, DIST on slice 0
- slice -1: read slice 2; compute VIS, DIFF, DIS, DIST on slice 1

- slice 0: read slice 3; compute VIS, DIFF, DIS, DIST on slice 2
- slice I (I=1,IIP): read slice I+3; compute VIS, DIFF ,DIS, DIST on slice I+2; step fields on slice I; write new fields on slice I
- The subroutine RDFLDS is written so that CALL RDFLDS(I,...) is treated as CALL RDFLDS(I+IIP,...) for I < 1, and CALL RDFLDS(I-IIP,...,) for I > IIP.
- Storage requirements: 6 slices of fields (I-2,I-1,I,I+1,I+2,I+3), 4 slices of viscosities (I-1,I,I+1,I+2), 5 slices of dissipations (I-2,I-1,I,I+1,I+2).

A.3 Implementation in the code

The code is written so that the changes in the structure of NNSTEPS when using backscatter should be transparent to the user. Backscatter requires storage of extra slices of fields and dissipations. Extra pointers IPA, IPB, IPC and IVISIPA have been introduced. These are used to select mean-field slices in calling CALCVIS. IPA, IPB and IPC point to slices I, I+1 and I+2 without backscatter, and to slices I+1, I+2 and I+3 with backscatter. The extra slices required for backscatter are dimensioned and read only when the backscatter is switched on through parameter statements, so there should be minimal memory or CPU penalty when switched off.

If making changes to NNSTEPS be very careful to ensure that you know which slice you are dealing with. Note, also that 'I' in routines such as CALCVIS is not the same as that in NNSTEPS - 'I' in CALCVIS is the slice on which we are calculating viscosity; this is I+1 or I+2 of NNSTEPS. Please see S.H.Derbyshire, A.R.Brown or M.K.MacVean if in any doubt.

An abbreviated version of NNSTEPS is now given.

```
-----------------------------------begin loop over timesteps
      DO 1 N=0,NN

        IPRINT=0
        IF(N.GT.0)THEN
           various housekeeping, notably check on timestep
           carried out on first step on each NN
        ENDIF ! (N.GT.0)

        IF(IBSCATP.EQ.1.AND.MOD(NSTEP,2).EQ.0.AND.N.NE.NN)CALL SETRAN
           set up random numbers for use in backscatter routines

        CALL SETINDEX(IM2,IM1,IO,IP1,IP2,IP3,IVISIM1,IVISIO,IVISIP1,
     &   IVISIP2,IDISIM2,IDISIM1,IDISIO,IDISIP1,IDISIP2)
        IF(IBSCATP.EQ.0)THEN
          IPA=IO
          IPB=IP1
          IPC=IP2
          IVISIPA=IVISIP1
        ELSE
          IPA=IP1
          IPB=IP2
          IPC=IP3
          IVISIPA=IVISIP2
        ENDIF
            set up pointers - this allows same code to be used whether
            reading two slices ahead (standard) or three (backscatter)

        IF(JJP.GT.1)CALL INITPS
            initialize pressure source

-----------------------------------begin loop over SLICEs
        DO 2 I=IFIRST,IIP

          IF(I.EQ.IFIRST.AND.IIP.GT.1) THEN
             CALL RDFLDS(I,U(0,1,IO).....)
             CALL RDFLDS(I+1,U(0,1,IP1).....)
             IF(IBSCATP.EQ.1)THEN
               CALL RDFLDS(I+2,U(0,1,IP2).....)
             ENDIF
          ENDIF
            Reading in velocity slices when on first slice
            (I,I+1=-1,0 without backscatter;
             I,I+1,I+2=-3,-2,-1 with backscatter)

          CALL RDFLDS(I+2+IBSCATP,U(0,1,IP3).....)
            Reading in velocity slice (all slices)
            (I+2 without backscatter, in which case IP3=IP2;
             I+3 with backscatter)
```

```
      IF(N.GT.0)THEN
        IF(I.EQ.IFIRST.AND.IIP.GT.1)THEN
        Pressure step, time-swap and smooth fields read in on
        first slice (see calls to RDFLDS above)
      ENDIF

        IF(JJP.GT.1)THEN
          CALL PSTEP(I+2+IBSCATP,ZU(0,1,IP3).....)
        ENDIF
        CALL SWAPSMTH(IFORSTEP,ISMTH,I+2+IBSCATP,U(0,1,IP3).....)
        Pressure step, time-swap and smooth fields read in (all
        slices)
      ENDIF  ! (N.GT.0)

----- Previous timestep now completed for this I -----

      IF(N.EQ.NN)THEN
        IF(I.GT.0)CALL WRTFLDS(N,I,INOSTEP,U(0,1,IO),...SU...)
        No further stepping; simply call WRTFLDS and return

      ELSE
        calculations for new timestep

        IF(I.EQ.IFIRST.AND.IFBCHGP.EQ.1)THEN
          CALL SETFLUX(TIME)
          IF(IPASTHP.EQ.0.OR.IPASQP.EQ.0)CALL CHGLOOK
        ENDIF
        set new surface heat flux and look up table for lower
        boundary condition if appropriate

        IF(IFORSTEP.EQ.1)THEN
          CALL CALCVIS(N,I+1+IBSCATP,IFORSTEP,U(0,1,IPA),.....)
        ELSEIF(IFORSTEP.EQ.0.AND.IFORSCALP.EQ.0)THEN
          CALL CALCVIS(N,I+1+IBSCATP,IFORSTEP,ZU(0,1,IPA),.....)
        ELSE   !IFORSCALP=1 AND IFORSTEP=0
          CALL CALCVIS(N,I+1+IBSCATP,IFORSTEP,ZU(0,1,IPA),.....)
        ENDIF
        calculate viscosity (on I+1 without backscatter; on
        I+2 with backscatter; pointers eg IPA ensure that correct
        fields are passed to CALCVIS. Note that CALCVIS also
        calculates 'surface' viscosity in LOWERBC.

        IF(I.GT.0) THEN
          CALL DYNVIS(N,I,IFORSTEP,U(0,1,IM2),....)
        calculate source terms on slice I

          IF(IBSCATP.EQ.1.AND.NSTEP.GE.NBEGSCATP)THEN
            CALL BCKSCT(I,SU.....)
            IF(IBSCATTP.EQ.1)CALL BCTSCT(I,STH.....)
            IF(IBSCATQP.EQ.1)CALL BCTSCT(I,SQ.....)
```

```
        ENDIF
      backscatter routines - add random forcing to source terms
      on slice I

        IF(N.EQ.1)THEN
          IF(I.EQ.1)CALL INDGNEW
      calls to diagnostic routines on each slice on
      first step of each NN.
      RESDGS for resolved scale, SUBDGS for subgrid diagnostics.
      Fields passed into these routines depend on numerical
      scheme in use.
         IF(I.EQ.IIP)THEN
           IF(ITSERP.EQ.1)CALL TIMSER
           CALL AVDG
      On last slice, call TIMSER (time series) and AVDG to
      update running averages of diagnostics
         ENDIF
        ENDIF

        CALL WRTFLDS(N,I,IFORSTEP,U(0,1,I0),.....SU....)
      call WRTFLDS for slice I which steps field and writes
      to packed storage

        ENDIF !_ENDIF(I.GT.0)
      ENDIF              ! _ENDIF(N.NE.NN)

      CALL REINDEX(IM2,IM1.......)
      IF(IBSCATP.EQ.0)THEN
        IPA=I0 etc.
      ELSE
        IPA=IP1 etc.
      ENDIF
        Cycle pointers and set correctly both with and without
        backscatter

  2   CONTINUE
------------------------------------ end loop over I (SLICEs)
      IF(N.LT.NN)THEN

        IF(IIP.GT.1.OR.JJP.GT.1)CALL POISSON
        CALL SET2D
      ENDIF
          POISSON solves elliptic equation for pressure. Calculation
          overwrites some expanded arrays in COMMON/LARGE/. These are
          recalculated by SET2D.

  1 CONTINUE
------------------------------------ end loop over N (Timesteps)

          Finally check time compared to control times in TESTTIME.
```

Annex B: The anelastic equations

B.1 The anelastic equations

The present model is formulated alternatively as (i) an incompressible Boussinesq or (ii) a 'deep anelastic' (or 'quasi-Boussinesq') system. The latter set forms the natural generalization of the incompressible Boussinesq equations (which in meteorology are principally a boundary-layer set) to the troposphere as a whole. They are nonhydrostatic but exclude sound waves (which are undesirable in atmospheric models).

The original 'deep anelastic' equations were invented by Batchelor (1953), although most of the meteorological literature gives Ogura and Phillips (1962) as the first reference. The latter pointed out a technical condition, namely a lower limit on time-scales, to justify the equation set. Physically this point is rather obvious and corresponds essentially to the exclusion of sound waves. Unfortunately the generalization of the the Batchelor-Ogura-Phillips (BOP) formulation from an isentropic to a more general basic state is a vexed issue. Users should be aware that 'anelastic' is not an unambiguous term.

The approach outlined below is consistent with that standard at Imperial College for some decades. It is also found in Lipps and Hemler (1982). The continuity equation is approximated slightly better than the momentum equation (whereas Clark (1977) for instance uses an almost exact momentum equation), but a good analogue of energy conservation is found. The underlying reason is the presence of an isomorphism (though not exact physical equivalence) with a nonhydrostatic pressure-coordinate set (Miller and White 1984).

The anelastic equations can be modified to give the slightly simpler, though less accurate, incompressible Boussinesq set, simply by discarding the gas law for the reference density profile, and setting this constant. This is done in the code if IANELP=0.

B.2 Derivation

In forecasting, plotting isobars at constant height is essentially equivalent to contouring geopotential Φ at constant pressure, via the approximation $p'/\rho_s \sim \Phi'$. Both Boussinesq and quasi-Boussinesq (anelastic) sets rest on very similar approximations, linked to the assumption that pressure and density fluctuations about some hydrostatic basic state are 'small'.

Formally, let $p = p_s(z) + p'$, $\rho = \rho_s(z) + \rho'$, then we require $\rho' \ll \rho_s$, $\theta_v' \ll \theta_{vs}$ and so on. We also require that the variation in θ_{vs} be small over the vertical wavelength of all significant modes. For density variations to be small we require low Mach number (else sound-waves will interact strongly with the flow). The 'wavelength' condition can accommodate the incompressible Boussinesq limit (vertical wavelength $\ll$ scale height) and also the isentropic (BOP) limit (vertical wavelength $\sim$ scale height $\ll$ scale of variation in θ).

In the absence of viscosity, the exact momentum equation is

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \mathbf{g} \tag{B1}$$

and writing $\mathbf{g} = \rho_s^{-1}\nabla p_s$ gives

$$\frac{D\mathbf{u}}{Dt} = \left(\frac{1}{\rho_s} - \frac{1}{\rho}\right)\nabla p - \frac{1}{\rho_s}\nabla p' \sim \frac{\rho'}{\rho_s}\mathbf{g} - \frac{1}{\rho_s}\nabla p' \tag{B2}$$

But

$$\frac{\rho'}{\rho_s} \sim -\frac{\theta_v'}{\theta_{vs}} + \frac{p'}{\rho_s}\left(\frac{d\rho}{dp}\right)_{\theta_v} \tag{B3}$$

If we assume the basic profiles are nearly isentropic, i.e. approximate

$$\left(\frac{d\rho}{dp}\right)_{\theta_v} \sim \left(\frac{d\rho_s}{dp_s}\right) = \frac{d\rho_s}{dz} \Big/ \frac{dp_s}{dz} = -\frac{1}{g\rho_s}\frac{d\rho_s}{dz} \tag{B4}$$

then

$$\frac{D\mathbf{u}}{Dt} \sim g\frac{\theta_v'}{\theta_{vs}}\hat{\mathbf{z}} - \nabla(p'/\rho_s) \tag{B5}$$

A useful way of thinking about the above equations is to write ρ^* for the density of a parcel when compressed or expanded adiabatically to its *local* reference pressure $p_s(z)$, and similarly T_v^* for the virtual temperature after such an operation. The amount of such compression or expansion will be small, since by assumption $p' \ll p_s$. However, by this means we may separate out the effect of fluctuating pressure on buoyancy. Both ρ^* and T_v^* may be calculated directly from the conservative variables, via

$$\rho^* \equiv p_s/RT_v^* \equiv [p_s/R\theta_v(p_s/p_r)^\kappa] \tag{B6}$$

(i.e. $\rho_s/\rho^* = \theta_v/\theta_{vs}$) where $\kappa = R/c_p \simeq 2/7$. Then

$$\frac{D\mathbf{u}}{Dt} \sim -g\frac{\rho^{*\prime}}{\rho_s}\hat{\mathbf{z}} - \nabla(p'/\rho_s) \tag{B7}$$

Hence, in general, buoyancy should be calculated using the reference pressure p_s because the term $\nabla(p'/\rho_s)$ already incorporates (an adiabatic approximation to) the effect of pressure fluctuations p' on buoyancy. If you use density directly to calculate buoyancy, you should use ρ^* not ρ. However, in effect $\theta = \theta^*$ and $\theta_v = \theta_v^*$ because these are unaffected by pressure fluctuations.

The exact continuity equation is

$$\frac{\partial\rho}{\partial t} + \nabla.(\rho\mathbf{u}) = 0 \tag{B8}$$

In the anelastic set, for purposes of continuity ρ is regarded as a function of height alone (i.e. not x, y or t). Ogura and Phillips pointed out in effect that the condition $\rho' \ll \rho_s$ was not formally sufficient, because the explicit time-derivative could be large; one needs formally to assume

$$\frac{D\rho'}{Dt} \ll \rho_s \parallel \nabla\mathbf{u} \parallel \tag{B9}$$

But this condition will only be violated by sound waves, and so except at high Mach number can be neglected. Its neglect is also consistent with the neglect of an extra term in (A5) (neglected through the approximation in (A4)) because retention of one of these terms (but not the other) can be shown to give distorted gravity waves (see Green's unpublished Imperial College notes).

So the complete set, in the absence of water phase changes, is

$$\frac{D\mathbf{u}}{Dt} = -\nabla(p'/\rho_s) + g\frac{\theta_v'}{\theta_{vs}}\hat{\mathbf{z}} + \text{viscous terms} \tag{B10}$$

$$\frac{D\theta_v}{Dt} = \text{diffusive terms} \tag{B11}$$

subject to the divergence constraint

$$\nabla.(\rho_s \mathbf{u}) = 0 \tag{B12}$$

where ρ_s, θ_{vs} are fixed 'reference profiles'.

In §3 the notation θ_{vs} is not used, but replaced by θ_s (as it is in the model), thus effectively choosing dry reference profiles. Although this is very slightly less accurate, there is no formal inconsistency with the anelastic approximations.

In view of their non-hydrostatic nature and mass-flux divergence constraint, these anelastic equations form the natural generalization of the incompressible Boussinesq equations to cases where the variation of ρ_s across the domain is not small. They obey an exact 'gas law', except for the approximation of p by p_s. The incompressible Boussinesq equations can be regarded as an approximation to the more general anelastic equations by approximating $\rho_s(z) \sim \rho_0$, constant and $\theta_s(z) \sim \theta_0$, constant. The basic state density ρ_s does not then obey any exact gas law.

We shall take c_p as constant, despite a slight dependence on humidity. The errors involved are small — "even in extreme conditions the factor involving q [i.e. the effect of humidity on κ] is only 1% less than unity, and so usually ignored" (Gill, 1982, p.53).

[*Note* water phase changes can be handled by using the liquid water temperature T_L, which like θ_v in the dry case, is conserved (unless precipitation occurs). In that option it is the calculation of buoyancy which is more complicated, because in effect a mixed-phase system obeys a more complicated 'gas law'. The key approximation, as before, is that density variation with height is dominated by the pressure effect, and not by the variation in the adiabatically conserved variable.]

B.3 Energy properties

The following energy analysis does not consider phase changes or the T_L option in the model. The reader concerned with moist processes, moist energetics and moist variables is referred to Shutts (1991), who concludes that, generally, discretization errors at cloud boundaries are a greater concern than some of the niceties of moist thermodynamics. The remainder of this section refers therefore to the option coded by IMOISTP=0.

First note that any quantity q governed by an equation of the form

$$\frac{Dq}{Dt} = \mathbf{u}.\nabla\phi_q + \frac{1}{\rho_s}\nabla.(\rho_s \mathbf{F}_q) \tag{B13}$$

(for some $\phi_q, \mathbf{F}_q$, where typically but not necessarily $\mathbf{F}_q = -K_q \nabla q$ for some K_q) obeys an overall conservation law

$$\frac{d}{dt}\int \rho_s q \, d^3\mathbf{r} = \text{boundary fluxes} \tag{B14}$$

since $\rho_s \mathbf{u}.\nabla q = \nabla.(\rho_s \mathbf{u} q)$, using (B12). Taking the scalar product of $\mathbf{u}$ with (B10) gives (in the absence of dissipative fluxes)

$$\frac{D}{Dt}\left[\frac{1}{2}\mathbf{u}^2\right] = -\mathbf{u}.\nabla(p'/\rho_s) + g w \theta_v'/\theta_{vs} \tag{B15}$$

But note that

$$\left(\frac{DT_v^*}{Dt}\right)_{\theta_v} = \left(\frac{\partial T_v^*}{\partial p_s}\right)_{\theta_v} \frac{Dp_s}{Dt} = -g w \rho_s \left(\frac{DT_v^*}{Dp_s}\right)_{\theta_v} \tag{B16}$$

Now from basic thermodynamics

$$\left(\frac{DT_v^*}{Dp_s}\right)_{\theta_v} = 1/\rho^* c_p \tag{B17}$$

and hence

$$\left(\frac{DT_v^*}{Dt}\right)_{\theta_v} = -gw\rho_s/c_p\rho^* = -gw\theta_v/c_p\theta_{vs} = -gw/c_p - gw\theta_v'/c_p\theta_{vs} \tag{B18}$$

whence

$$\frac{D}{Dt}\left[\frac{1}{2}\mathbf{u}^2 + c_pT_v^* + gz\right] = -\mathbf{u}.\nabla(p'/\rho_s) \tag{B19}$$

leading (cf. (B13))to conservation of

$$\int \rho_s \left[\frac{1}{2}\mathbf{u}^2 + c_pT_v^*\right] d^3\mathbf{r} \tag{B20}$$

(since $\int gzd^3\mathbf{r}$ is constant).

The form of the energy integral may be unfamiliar to those brought up on incompressible fluid dynamics. However, by noting that

$$T_v^*/\theta_v = \Pi_s \tag{B21}$$

where $\Pi_s \equiv (p_s/p_r)^{R/c_p}$ is the Exner function of the reference state, we may transform it into something more familiar. The hydrostatic relation may be written

$$\frac{\partial \Pi_s}{\partial z} = -\frac{g}{c_p\theta_s} \tag{B22}$$

so that in particular for an *adiabatic* reference state Π_s is linearly related to the geometric height z. This motivates the definition of a Hoskins height-like coordinate

$$z_s = \frac{c_p\theta_0}{g}(1 - \Pi_s) \tag{B23}$$

where θ_0 is some fixed value of θ, so that for an isentropic reference profile ($\theta_s \equiv \theta_0$) we have $z_s \equiv z$. In general, of course, z_s will differ from z. But now we can write our energy integral as

$$\int \rho_s \left[\frac{1}{2}\mathbf{u}^2 - \frac{g}{\theta_0}\theta_v z_s\right] d^3\mathbf{r} + \text{const.} \tag{B24}$$

(where the 'constant' assumes conservation of $\int \rho_s\theta_v d^3\mathbf{r}$).

Our quasi-Boussinesq equations are isomorphic to another approximate set obtained from similar assumptions in non-hydrostatic pressure-coordinate models, which gives some confidence that we have not introduced qualitative errors (see Miller and White, 1984). Note in particular that the initially slightly puzzling anelastic term $\nabla(p'/\rho_s)$ (why should ρ_s go inside the bracket?) corresponds (at linearized level) to $\nabla_p\Phi$, i.e. the gradient of geopotential along pressure-surfaces.

Finally the validity of our anelastic equations can be examined from the energy integral (B20). In particular, propagating gravity-wave packets should be reasonably accurately represented even in (say) an isothermal atmosphere, because the energy-conservation law is essentially correct. This represents a distinct advantage of our set over the BOP equations.

References

Antonopoulos-Domis, M. (1981): Large-eddy simulation of a passive scalar in isotropic turbulence. *J. Fluid Mech.*, **104**, 55–79.

Batchelor, G.K. (1953): The condition for dynamical similarity of models of a frictionless perfect-gas atmosphere. *Quart. J. Roy. Meteor. Soc.*, **79**, 224–235.

Batchelor, G.K. (1967): An Introduction to Fluid Dynamics. CUP.

Brown, A.R., Derbyshire, S.H. and Mason, P.J. (1994): Large-eddy simulation of stable atmospheric boundary layers with a revised stochastic subgrid model. Accepted for publication in qj.

Bull, J.M. and Derbyshire, S.H. (1990): Numerical Solution of the Surface Layer Equations. MetO(P) Turbulence and Diffusion Note No. 197 (unpublished).

Clark, T. (1977): A small-scale dynamical model using a terrain-following coordinate system. *J. Comp. Phys.* **24**, 186-215.

Coleman, G.N. (1990): A numerical study of the stratified turbulent Ekman Layer. PhD thesis, Stanford University.

Cotton, Tripoli, Rauber and Mulvihill (1986): Numerical Simulation of the Effects of Varying Ice Crystal Nucleation Rates and Aggregation Processes on Orographic Snowfall, *J. Climate and Applied Met.*, **25**, 1658-1679.

Derbyshire, S.H. and Kershaw, R. (1993): Turbulence simulation in the Meteorological Office. *Meteorological Magazine*, **122**, 25-34.

Flatau, Tripoli and Cotton (1989): The SCU-RAMS Cloud microphysics module general theory and code documentation, *Colorado State University*, **451**, 88pp.

Francis P., Jones A., Saunders R., Shine K., Slingo A. and Sun Z. (1994): An observational and theoretical study of the radiative properties of cirrus: Some results from ICE'89. *Quart. J. Roy. Meteor. Soc.*, **120**, 809–848.

Gill, A.E. (1982): Atmosphere-Ocean Dynamics. Academic Press.

Gray, M.E.B. (1991): Tests of the large-eddy model. MetO(P) Internal Note.

Haltiner, G.J. and Williams, R.T. (1980): Numerical prediction and dynamical meteorology. Wiley.

Ingrams W. (1993): Radiation, *Unified model documentation paper 23*, United Kingdom Meteorological Office.

Kessler,E. (1974): Model precipitation and vertical air currents. *Tellus*, **26**, 519-542.

Lee,I.Y. (1989): Evaluation of cloud microphysics parameterizations. *Atmospheric Research*, **24** Nos.1-4 December 1989.

van Leer, B. (1974): Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second order scheme. *J. Comput. Phys.* **14**, 361-370.

Leonard, B.P. (1991): The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection. *Comput. Methods Appl. Mech. Eng.*, **19**, 17-74.

Leonard, A. (1974): Energy cascade in large-eddy simulations of turbulent fluid flows. *Adv. Geophys.* **18**A, 237-248.

Lilly (1990): Subgrid closures in large-eddy simulation. In: *Lecture notes on turbulence, from NCAR Summer School 1987.* World Scientific.

Lin, Farley and Orville (1983): Bulk Parametrization of the Snow Field in a Cloud Model, *J. Climate and Applied Met.*, **22**, 1065-1092.

Lipps, F.B. and Hemler, R.S. (1982): A scale analysis of deep moist convection and some related numerical calculations. *J. Atmos. Sci.*, **39**, 2192–2210.

MacVean, M.K. (1993) A numerical investigation of the criterion for cloud-top entrainment instability. *J. Atmos. Sci.*, **50**, 2481–2495.

MacVean, M.K. and Mason, P.J. (1990) Cloud-top entrainment instability through small-scale mixing and its parametrization in numerical models. *J. Atmos. Sci.*, **47**, 1012–1030.

Mason, P.J. (1989): Large-eddy simulation of the convective atmospheric boundary layer. *J. Atmos. Sci.*, **46**, 1492–1516.

Mason, P.J. (1994): Large-eddy simulation: A critical review of the technique. *Quart. J. Roy. Meteor. Soc.*, **120**, 1–26.

Mason, P.J. and Callen, N.S. (1986): On the magnitude of the subgrid-scale eddy coefficient in large-eddy simulations of turbulent channel flow. *J. Fluid Mech.* , **162**439462.

Mason, P.J. and Derbyshire, S.H. (1990): Large-eddy simulation of the Stably-Stratified Atmospheric Boundary Layer. *Bound. Layer Meteor.*, **53**, 117–162.

Mason, P.J. and Thomson, D.J. (1987): Large-eddy Simulation of the Neutral-Static-Stability Planetary Boundary Layer, *Quart. J. Roy. Meteor. Soc.*, **113**, 413–433.

Mason, P.J. and Thomson, D.J. (1992): Stochastic backscatter in Large-eddy simulations of boundary layers. *J. Fluid Mech.*, **242**, 51–78.

McCumber,Tao and Simpson (1991): Comparisons of Ice-Phase Microphysical Schemes. Using Numerical Simulations of tropical Convection. *J. Appl. Meteor.*, **30**, 7–985.1004

Miller, M.J. and White, A.A. (1984): *Quart. J. Roy. Meteor. Soc.*, **110**, 515–33.

Moin, P. and Kim, J. (1982): Numerical investigations of turbulent channel flow. *J. Fluid Mech.*, **118**, 341–377.

Ogura, Y. and Phillips, N.A. (1962): A scale analysis of deep and shallow convection in the atmosphere. *J. Atmos. Sci.*, **19**, 173–179.

Piacsek, S.A. and Williams, G.P. (1970): Conservation properties of convection difference schemes. *J. Comp. Phys.* **6**, 392-405.

Shutts, G.J. (1991): Liquid Water Static Energy- a moist thermodynamic variable for use in numerical models. JCMM Mesoscale Newsletter No.3, Oct. 1991.

Shutts, G.J. and Gray, M.E.B. (1994): A numerical modelling study of the geostrophic adjustment process following deep convection. *Quart. J. Roy. Meteor. Soc.*, **120**, 1145–1178.

Slingo, A. and Wilderspin, R. (1986): Development of a revised longwave radiation scheme for an atmospheric general circulation model. *Quart. J. Roy. Meteor. Soc.*, **112**, 371–386.

Stephens, G. (1984): The parametrization of radiation for numerical weather prediction and climate models, *Mon. Wea. Rev.*, **112**, 826-867.

Swann, H. (1993): Cloud Microphysical Processes — a description of the parameterization used in the large-eddy model. *JCMM Internal Report.*,**10**

Wood, N. (1989): The description of a partial implicit scheme applied to a 2-D turublence model with a linear stability analysis for both explicit and implicit schemes. MetO(P) Turbulence and Diffusion Note No. 192 (unpublished).