Report No. 6

The ECHAM3 Atmospheric General Circulation Model

Edited by:
Deutsches Klimarechenzentrum,
Modellbetreuungsgruppe,
Hamburg, September 1993
Revision 2

Contents

1. SUMMARY PAGE

1.1 SHORT DESCRIPTION AND MAIN AUTHORS

The ECHAM model has been developed from the ECMWF model (cycle 31, November 1988). It contains several changes, mostly in the parameterization, in order to adjust the model for climate simulations.

The reference resolution is T42, but the model is set up to use resolutions in the range T21 to T106. Long term integrations have so far only been done for T21, T42 and T63. The lay-out is as follows
(⊕ indicates changes from the original ECMWF model):

1.1.1 Numerical solution

- prognostic variables:
  vorticity, divergence, temperature,
  log surface pressure, water vapour,
  ⊕ cloud water
- horizontal representation:
  spectral transform, triangular truncation (T21/T42/T63/T106)
- vertical representation:
  hybrid coordinate system, second order finite differences, 19 layers
- time integration:
  semi-implicit; leap frog with time filter,
  ⊕ Δt = 40 min (T21), Δt = 24 min (T42), Δt = 15 min (T63), Δt = 12 min (T106)

1.1.2 Surface boundary conditions

- SST and sea-ice:
  blended data set (Reynolds, 1988) and Alexander and Mobley (1976) (ECHAM1,ECHAM2)
  COLA/CAC AMIP SST and Sea-Ice Dataset (ECHAM3)
- orography:
  mean terrain heights computed from high resolution US Navy data set
  land-sea mask: from US Navy data set
- roughness length
  sea: Charnock formula, modified after Miller et al. (1992)
  sea-ice: constant (0.01 m)
  land: function of vegetation and orography (variance)
- vegetation:
  fraction of grid area covered by vegetation based on
  Wilson and Henderson - Sellers(1985) data

- albedo:
  ⊕ sea: function of solar zenith angle
  bare land: satellite data (Geleyn and Preuss (1983))
  ⊕ sea ice: function of temperature (Robock,1980)
  ⊕ land ice: function of temperature (Robock, 1980; Kukla and Robinson, 1980)
  ⊕ snow: function of temperature and fractional forest area (Robock, 1980; Matthews, 1983)

1.1.3 Physical parameterization

- ⊕ radiation (Hense et al.,1982; Rockel et al, 1991; Eickerling, 1989)
  - two-stream approximation
  - six spectral intervals in the terrestrial part
  - four spectral intervals in the solar part
  - gaseous absorbers: H_2O, CO_2 and O_3 (CO_2 and O_3 prescribed)
  - aerosols: prescribed
  - clouds: computed cloud optical depth and cloud cover
  - emissivity: function of cloud water path (Stephens, 1978)
  - continuum absorption: included
  - cloud overlap: maximum for contiguous clouds layer and random otherwise
  - diurnal cycle: included
  - radiation time step: 2 hours
- ⊕ clouds (Sundquist, 1978; Roeckner and Schlese, 1985; Roeckner et al, 1991)
  - cloud water transport equation
  - subgrid-scale condensation and cloud formation with different thresholds for convective and stratiform clouds (Xu and Krueger, 1991)
  - temperature dependent partitioning of liquid/ice phase (Matveev, 1984)
  - rain formation by auto-conversion of cloud droplets (Sundquist, 1978)
  - sedimentation of ice crystals (Heymsfield, 1977)
  - evaporation of cloud water
  - evaporation of precipitation
- convection (Tiedtke, 1989)
  - mass flux scheme for deep, shallow and mid-level convection
  - clouds are represented by a bulk model and include updraft and downdraft mass fluxes
  - convective momentum transport is parameterized according to Schneider and Lindzen (1976)
  - Evaporation of rain is parameterized according to Kessler (1969)
  - stratocumulus convection is parameterized as a vertical diffusion process with enhanced eddy diffusion coefficients (Tiedtke et al., 1988)

- planetary boundary layer (Louis, 1979)
    - surface fluxes of momentum, heat, moisture and cloud water are calculated from Monin-Obukhov similarity theory with transfer coefficients depending on roughness length and Richardson number
    - above the surface layer: eddy diffusivity approach with coefficients depending on wind shear, thermal stability and mixing length
    - above the PBL: vertical diffusion only for unstable stratification
    - ⊕ cloud water
    - ⊕ “moist” Richardson number (Brinkop, 1991; 1992)
- ⊕ land-surface processes (Sellers et al., 1986; Blondin, 1989; Dümenil and Todini, 1992)
    - heat transfer: diffusion equation solved in a 5-layer model with zero heat flux at the bottom (10 m)
    - water budget equation for three reservoirs: soil moisture, interception reservoir (vegetation), snow
    - vegetation effects: stomatal control on evapotranspiration and interception of rain and snow
    - run-off scheme: based on catchment considerations including sub-grid scale variations of field capacity over inhomogeneous terrain
    - sea-ice temperature calculated from surface energy budget
- ⊕ horizontal diffusion (Laursen and Eliasen, 1989)
    - scale selective operator applied beyond a threshold wave number
- gravity wave drag (Palmer et al, 1986; Miller et al, 1989)
    - surface stress due to gravity waves, which are exited by stably stratified flow over irregular terrain is calculated from linear theory and dimensional considerations
    - orographic forcing prescribed as a directionally dependent sub-grid scale orographic variance computed from the high resolution US Navy data set
    - vertical structure of momentum flux induced by gravity waves calculated from a local wave Richardson number, which describes the onset of turbulence due to convective instability and the turbulent breakdown approaching a critical level
    - the GWD scheme is not used at T21 resolution

1.2 ACKNOWLEDGEMENTS

The ECHAM climate model has been developed from the ECMWF model (therefore the first part of its name: EC) and a comprehensive parametrisation package developed at Hamburg (therefore the abbreviation HAM). This documentation has been written in a similar fashion: A substantial part is based on the ECMWF documentation, which then had been modified to describe the newly implemented subroutines and the changes necessary for climate experiments.

2. MODEL DYNAMICS

2.1 INTRODUCTION

In this section the technical details of the adiabatic part of the ECHAM operational model are described. The first two sections describe the governing equations, the coordinates and the discretization schemes used in the operational ECHAM model. Attention is concentrated on the representation of the explicitly resolved adiabatic processes, but a derivation of the equations including terms requiring parametrization is included in Appendix A.

Detailed descriptions of the parametrizations themselves are given in section 3.

The ECHAM model is formulated in spherical harmonics. After the inter-model comparisons by Jarraud et al. (1981) and Girard and Jarraud (1982) truncated expansions in terms of spherical harmonics were adopted for the representation of dynamical fields. The transform technique developed by Elisasen et al. (1970), Orszag (1970) and Machenhauer and Rasmussen (1972) is used such that non-linear terms, including parameterizations, are evaluated at a set of almost regularly distributed grid points (Gaussian grid).

In the vertical, a flexible coordinate is used, enabling the model to use either the usual terrain-following sigma coordinate (Phillips, 1957), or a hybrid coordinate for which upper-level model surfaces “flatten” over steep terrain, becoming surfaces of constant pressure in the stratosphere (Simmons and Burridge, 1981, Simmons and Strüfing, 1981). Moist processes are treated in a consistent way in both the dynamical equations and parameterization schemes.

Section 2.2 presents the continuous form of the governing equations. Sections 2.3 and 2.4 give details of the spectral discretization and of the vertical coordinate and its associated vertical finite difference scheme. The temporal finite-difference scheme, which includes not only a conventional semi-implicit treatment of gravity-wave terms (Robert et al., 1972) but also a semi-implicit treatment of the advection of vorticity and moisture (Jarraud et al., 1982), is also described (section 2.5), as is the formulation chosen for horizontal diffusion (section 2.6).

2.2 THE CONTINUOUS EQUATIONS

Although the model has been programmed for one particular form of vertical coordinate, which is introduced in section 2.4, it is convenient to introduce the equations and their spectral discretization for a general pressure-based terrain-following vertical coordinate, $\eta(p, p_S)$. This must be a monotonic function of pressure, p, and depend also on surface pressure, p_S, in such a way that

$$\eta(0, p_S) = 0 \quad \text{and} \quad \eta(p_S, p_S) = 1$$

For such a coordinate, the continuous formulation of the primitive equations for a dry atmosphere may be directly derived from their basic height coordinate forms following Kasahara (1974).

During the design of the model, a detailed derivation of the corresponding equations for a moist atmosphere, including a separation into terms to be represented explicitly in the subsequent discretized form of the equations and terms to be parameterized, was carried out. It is shown in Appendix A that under certain approximations, the momentum, thermodynamic and moisture equations may be written:

$$\frac{\partial U}{\partial t} - (f+\xi)\cdot V + \dot{\eta}\frac{\partial U}{\partial \eta} + \frac{R_d T_v}{a}\frac{\partial}{\partial \lambda} ln\, p + \frac{1}{a}\frac{\partial}{\partial \lambda}(\phi + E) = P_U + K_U \tag{2.2.1}$$

$$\frac{\partial V}{\partial t} + (f+\xi)\cdot U + \dot{\eta}\frac{\partial V}{\partial \eta} + \frac{R_d T_v}{a}(1-\mu^2)\frac{\partial}{\partial \mu} ln\, p + \frac{(1-\mu^2)}{a}\frac{\partial}{\partial \mu}(\phi + E) = P_V + K_V \tag{2.2.2}$$

$$\frac{\partial T}{\partial t} + \frac{U}{a(1-\mu^2)}\frac{\partial T}{\partial \lambda} + \frac{V}{a}\frac{\partial T}{\partial \mu} + \dot{\eta}\frac{\partial T}{\partial \eta} - \frac{\kappa T_v \omega}{(1+(\delta-1)q_v)p} = P_T + K_T \tag{2.2.3}$$

$$\frac{\partial q_v}{\partial t} + \frac{U}{a(1-\mu^2)}\frac{\partial q_v}{\partial \lambda} + \frac{V}{a}\frac{\partial q_v}{\partial \mu} + \dot{\eta}\frac{\partial q_v}{\partial \eta} = P_{q_v} + K_{q_v} \tag{2.2.4}$$

$$\frac{\partial q_w}{\partial t} + \frac{U}{a(1-\mu^2)}\frac{\partial q_w}{\partial \lambda} + \frac{V}{a}\frac{\partial q_w}{\partial \mu} + \dot{\eta}\frac{\partial q_w}{\partial \eta} = P_{q_w} + K_{q_w} \tag{2.2.5}$$

where q_v is the water vapour mixing ratio and $q_w = q_l + q_i$ is the cloud water mixing ratio including the liquid q_l and the solid fraction q_i.

The continuity equation is

$$\frac{\partial}{\partial \eta}\left(\frac{\partial p}{\partial t}\right) + \nabla\cdot\left(\underset{\sim}{v}_h\frac{\partial p}{\partial \eta}\right) + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right) = 0 \tag{2.2.6}$$

and the hydrostatic equation takes the form

$$\frac{\partial \phi}{\partial \eta} = -\frac{R_d T_v}{p}\frac{\partial p}{\partial \eta} \tag{2.2.7}$$

The pressure coordinate vertical velocity is given by

$$\omega = \underset{\sim}{v}_h \nabla p - \int_0^\eta \nabla \cdot \left(\underset{\sim}{v}_h \frac{\partial p}{\partial \eta}\right) d\eta \tag{2.2.8}$$

and explicit expressions for the rate of change of surface pressure, and for $\dot{\eta}$, are obtained by integrating (2.2.6), using the boundary conditions $\dot{\eta} = 0$ at $\eta = 0$ and $\eta = 1$:

$$\frac{\partial p_S}{\partial t} = -\int_0^1 \nabla \cdot \left(\underset{\sim}{v}_h \frac{\partial p}{\partial \eta}\right) d\eta \tag{2.2.9}$$

and

$$\dot{\eta} \frac{\partial p}{\partial \eta} = -\frac{\partial p}{\partial t} - \int_0^\eta \nabla \cdot \left(\underset{\sim}{v}_h \frac{\partial p}{\partial \eta}\right) d\eta \tag{2.2.10}$$

(2.2.9) may also be written

$$\frac{\partial}{\partial t} ln p_S = -\frac{1}{p_S} \int_0^1 \nabla \cdot \left(\underset{\sim}{v}_h \frac{\partial p}{\partial \eta}\right) d\eta \tag{2.2.11}$$

Variables and constants are defined in Appendix B.2.

In the special case of sigma coordinates $\eta = \sigma = p/p_S$, the above equations are the same as those used in the first operational ECMWF model, apart from the factor $(1 + (\delta - 1) \cdot q_v)$ in (2.2.3), which differs from unity by an amount of the same order as the difference between temperature and virtual temperature, and apart also from differences in the terms written symbolically on the right-hand sides of (2.2.1) - (2.2.5), which are those requiring parametrization. Following the derivation given in Appendix A, the terms P_U, P_V, P_T, P_{q_v} and P_{q_w} are written:

$$P_U = -g\, cos\theta \left(\frac{\partial p}{\partial \eta}\right)^{-1} \frac{\partial J_U}{\partial \eta} \tag{2.2.12}$$

$$P_V = -g\, cos\theta \left(\frac{\partial p}{\partial \eta}\right)^{-1} \frac{\partial J_V}{\partial \eta} \tag{2.2.13}$$

$$P_T = \frac{1}{C_p} \left(Q_R + Q_L + Q_D - g \left(\frac{\partial p}{\partial \eta}\right)^{-1} \left[\frac{\partial J_S}{\partial \eta} - C_{pd} T (\delta - 1) \frac{\partial J_{q_v}}{\partial \eta} \right] \right) \tag{2.2.14}$$

$$P_{q_v} = S_{q_v} - g \left(\frac{\partial p}{\partial \eta}\right)^{-1} \frac{\partial J_{q_v}}{\partial \eta} \tag{2.2.15}$$

$$P_{q_w} = S_{q_w} - g \left(\frac{\partial p}{\partial \eta}\right)^{-1} \frac{\partial J_{q_w}}{\partial \eta} \tag{2.2.16}$$

where

$$C_p = C_{pd} (1 + (\delta - 1)\, q_v)$$

In (2.2.12) - (2.2.16), J_U, J_V, J_s, J_{q_v} and J_{q_w} represent net parametrized vertical fluxes of momentum, dry static energy $(C_p \cdot T + \phi)$, moisture and cloud water. They include fluxes due to convection and boundary-layer turbulence. Q_R, Q_L and Q_D represent heating due respectively to radiation, internal phase changes (including the evaporation of precipitation) and to internal dissipation of kinetic energy associated with the P_U and P_V terms. S_{q_v} and S_{q_w} denote the rates of change of q_v and q_w due to condensation, evaporation and precipitation formation. Details of the calculation of these terms are given in section 3.6.

The terms K_U, K_V, K_T, K_{q_v} and K_{q_w} in (2.2.1) - (2.2.5) represent the influence of unresolved horizontal scales. Their treatment differs from that of the P_U, P_V, P_T, P_{q_v} and P_{q_w} terms in that it does not involve a physical model of sub-grid scale processes, but rather a numerically convenient form of scale-selective diffusion of a magnitude determined empirically to ensure a realistic behaviour of resolved scales. These terms are specified in section 2.6.

In order to apply the spectral method, Eqs. (2.2.1) and (2.2.2) are written in vorticity and divergence form (Bourke, 1972). They become

$$\frac{\partial \xi}{\partial t} = \frac{1}{a(1-\mu^2)}\frac{\partial}{\partial \lambda}(F_V + P_V) - \frac{1}{a}\frac{\partial}{\partial \mu}(F_U + P_U) + K_\xi \tag{2.2.17}$$

$$\frac{\partial D}{\partial t} = \frac{1}{a(1-\mu^2)}\frac{\partial}{\partial \lambda}(F_U + P_U) + \frac{1}{a}\frac{\partial}{\partial \mu}(F_V + P_V) - \nabla^2 G + K_D \tag{2.2.18}$$

where

$$F_U = (f+\xi)V - \dot{\eta}\frac{\partial U}{\partial \eta} - \frac{R_d T_v}{a}\frac{\partial}{\partial \lambda} lnp \tag{2.2.19}$$

$$F_V = -(f+\xi)U - \dot{\eta}\frac{\partial V}{\partial \eta} - \frac{R_d T_v}{a}(1-\mu^2)\frac{\partial}{\partial \mu} lnp \tag{2.2.20}$$

and

$$G = \phi + E \tag{2.2.21}$$

We also note that a streamfunction ψ and velocity potential χ may be introduced such that

$$\left.\begin{aligned} U &= \frac{1}{a}\left[-(1-\mu^2)\frac{\partial \psi}{\partial \mu} + \frac{\partial \chi}{\partial \lambda}\right] \\ V &= \frac{1}{a}\left[\frac{\partial \psi}{\partial \lambda} + (1-\mu^2)\frac{\partial \chi}{\partial \mu}\right] \\ \xi &= \nabla^2 \psi \\ &\text{and} \\ D &= \nabla^2 \chi \end{aligned}\right\} \tag{2.2.22}$$

2.3 HORIZONTAL DISCRETIZATION

2.3.1 Spectral representation

The basic prognostic variables of the model are ξ, D, T, q_v, q_w and $ln\ p_S$. They, and the surface geopotential ϕ_S, are represented in the horizontal by truncated series of spherical harmonics:

$$X(\lambda, \mu, \eta, t) = \sum_{m=-M}^{M} \sum_{n=m}^{N(M)} X_n^m(\eta, t)\, P_n^m(\mu)\, e^{im\lambda} \qquad (2.3.1.1)$$

where X is any variable, m is the zonal wave number and n is the meridional index. The $P_n^m(\mu)$ are the Associated Legendre Functions of the first kind, defined here by

$$P_n^m(\mu) = \sqrt{(2n+1)\frac{(n-m)!}{(n+m)!}}\,\frac{1}{2^n n!}(1-\mu^2)^{m/2}\frac{d^{(n+m)}}{d\mu^{(n+m)}}(\mu^2-1), \quad (m \geq 0) \qquad (2.3.1.2)$$

and

$$P_n^{-m}(\mu) = P_n^m(\mu)$$

This definition is such that

$$\frac{1}{2}\int_{-1}^{1} P_n^m(\mu)\, P_s^m(\mu)\, d\mu = \delta_{ns} \qquad (2.3.1.3)$$

where δ_{ns} is the Kronecker delta function.

The X_n^m are the complex-valued spectral coefficients of the field X, and they are given by

$$X_n^m(\eta, t) = \frac{1}{4\pi}\int_{-1}^{1}\int_{0}^{2\pi} X(\lambda, \mu, \eta, t)\, P_n^m(\mu)\, e^{-im\lambda} d\lambda d\mu \qquad (2.3.1.4)$$

Since X is real

$$X_n^{-m} = (X_n^m)^* \qquad (2.3.1.5)$$

where ( )* denotes the complex conjugate. The model thus deals explicitly only with the X_n^m for $m \geq 0$.

The Fourier coefficients of X, $X_m(\mu, \eta, t)$ are defined by

$$X_m(\mu, \eta, t) = \frac{1}{2\pi}\int_{0}^{2\pi} X(\lambda, \mu, \eta, t)\, e^{-im\lambda} d\lambda \qquad (2.3.1.6)$$

or using (2.3.1.1), by

$$X_m(\mu, \eta, t) = \sum_{n=m}^{N(m)} X_n^m(\eta, t)\, P_n^m(\mu) \qquad (2.3.1.7)$$

with

$$X(\lambda, \mu, \eta, t) = \sum_{m=-M}^{M} X_m(\mu, \eta, t)\, e^{im\lambda} \tag{2.3.1.8}$$

Horizontal derivatives are given analytically by

$$\left(\frac{\partial X}{\partial \lambda}\right)_m = im\, X_m \tag{2.3.1.9}$$

and

$$\left(\frac{\partial X}{\partial \mu}\right)_m = \sum_{n=m}^{N(m)} X_n^m \frac{d}{d\mu} P_n^m \tag{2.3.1.10}$$

where the derivative of the Legendre Function is given by the recurrence relation:

$$(1-\mu^2)\frac{d}{d\mu}P_n^m = -n\varepsilon_{n+1}^m P_{n+1}^m + (n+1)\,\varepsilon_n^m P_{n-1}^m \tag{2.3.1.11}$$

with

$$\varepsilon_n^m = \left(\frac{n^2 - m^2}{4n^2 - 1}\right)^{1/2} \tag{2.3.1.12}$$

An important property of the spherical harmonics is:

$$\nabla^2 (P_n^m(\mu)\, e^{im\lambda}) = -\frac{n(n+1)}{a^2} P_n^m(\mu)\, e^{im\lambda} \tag{2.3.1.13}$$

Relationships (2.2.22) may thus be used to derive expressions for the Fourier velocity coefficients, U_m and V_m, in terms of the spectral coefficients ξ_n^m and D_n^m. It is convenient for later reference to write these expressions in the form

$$U_m = U_{\xi m} + U_{Dm} \tag{2.3.1.14}$$

$$V_m = V_{\xi m} + V_{Dm} \tag{2.3.1.15}$$

where

$$U_{\xi m} = -a \sum_{n=m}^{N(m)} \frac{1}{n(n+1)} \xi_n^m H_n^m(\mu) \tag{2.3.1.16}$$

$$U_{Dm} = -a \sum_{n=m}^{N(m)} \frac{im}{n(n+1)} D_n^m P_n^m(\mu) \tag{2.3.1.17}$$

$$V_{\xi m} = -a \sum_{n=m}^{N(m)} \frac{im}{n(n+1)} \xi_n^m P_n^m(\mu) \tag{2.3.1.18}$$

$$V_{Dm} = a \sum_{n=m}^{N(m)} \frac{1}{n(n+1)} D_n^m H_n^m(\mu) \tag{2.3.1.19}$$

and

$$H_n^m(\mu) = -(1-\mu^2)\frac{d}{d\mu}P_n^m \tag{2.3.1.20}$$

The H_n^m can be computed from the recurrence relation (2.3.1.11).

The model is programmed to allow for a flexible pentagonal truncation, depicted in Figure 1 (Baede et al, 1979). This truncation is completely defined by the three parameters J, K and M illustrated in the Figure. The common truncations are special cases of the pentagonal one:

Triangular	$M = J = K$
Rhomboidal	$K = J + M$
Trapezoidal	$K = J, K > M$

The summation limit, $N(m)$ is given by

$$N = J + |m| \qquad \text{if } J + |m| \le K,$$

and

$$N = K \qquad \text{if } J + |m| > K.$$

The standard truncation is triangular (i.e. $J = K = M = N$) at wave numbers 21, 42, 63 or 106 .

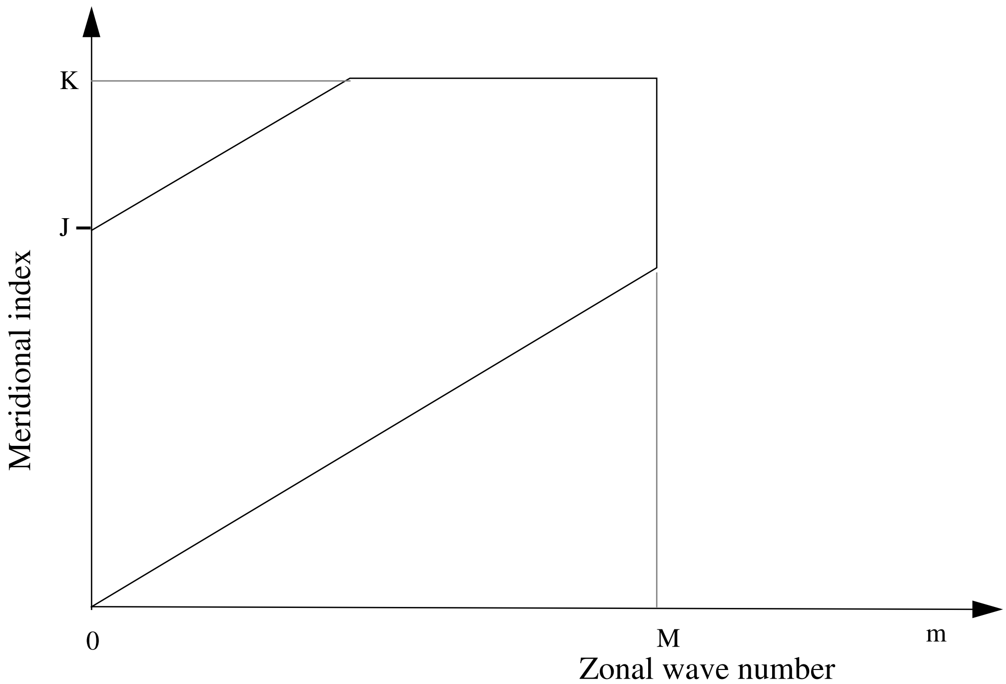


Figure 1 Pentagonal truncation

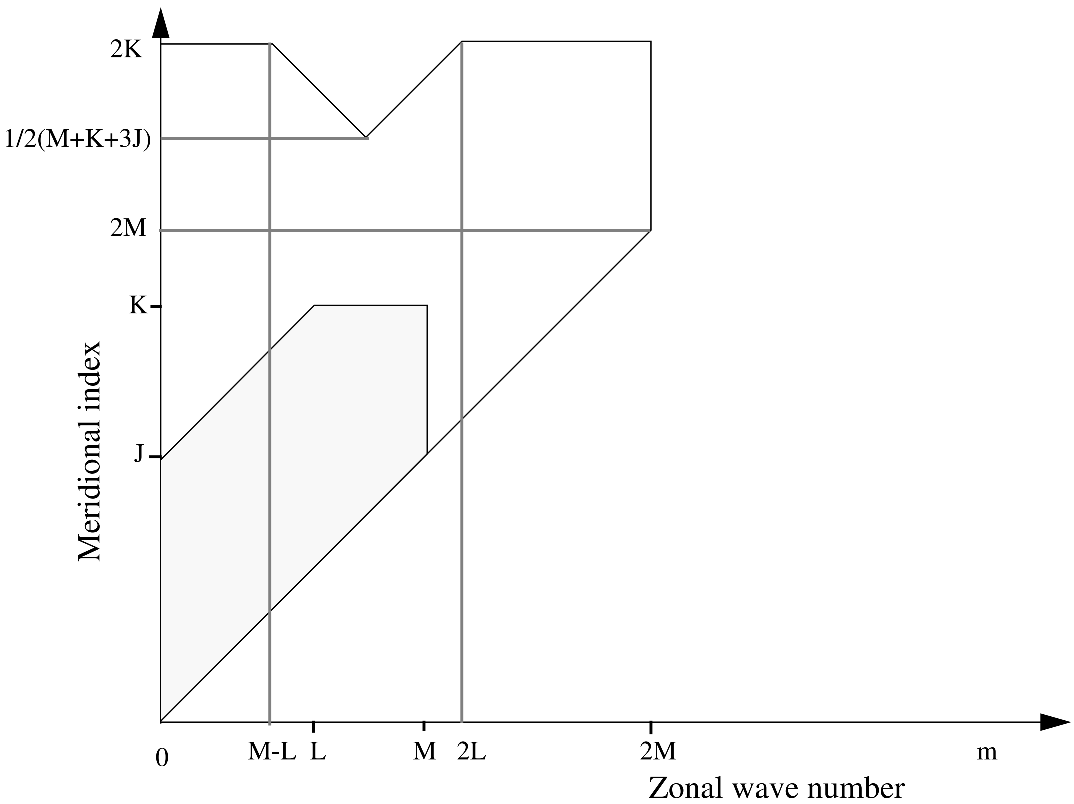


Figure 2 Product truncation

2.3.2 Spectral /grid-point transforms, and the evaluation of spectral tendencies

The general form of the calculations follows that of the early multi-level spectral models described by Bourke (1974) and Hoskins and Simmons (1975), although the present model differs in its use of an advective form for the temperature and moisture equations (2.2.17), (2.2.18), (2.2.3), (2.2.4), (2.2.5) and (2.2.11). Equations for the corresponding spectral coefficients are obtained by multiplying each side of these equations by $P_n^m e^{-im\lambda}$, and integrating over the sphere. This yields, from (2.3.1.4),

$$\frac{\partial}{\partial t}\xi_n^m = \frac{1}{4\pi a}\int_{-1}^{1}\int_{0}^{2\pi}\left(\frac{1}{1-\mu^2}\frac{\partial}{\partial\lambda}(F_V+P_V) - \frac{\partial}{\partial\mu}(F_U+P_U)\right)P_n^m(\mu)\,e^{-im\lambda}d\lambda d\mu + (K_\xi)_n^m \tag{2.3.2.1}$$

$$\frac{\partial}{\partial t}D_n^m = \frac{1}{4\pi a}\int_{-1}^{1}\int_{0}^{2\pi}\left(\frac{1}{1-\mu^2}\frac{\partial}{\partial\lambda}(F_U+P_U) + \frac{\partial}{\partial\mu}(F_V+P_V)\right)P_n^m(\mu)\,e^{-im\lambda}d\lambda d\mu$$
$$-\frac{1}{4\pi}\int_{-1}^{1}\int_{0}^{2\pi}(\nabla^2 G)\,P_n^m(\mu)\,e^{-im\lambda}d\lambda d\mu \quad + (K_D)_n^m \tag{2.3.2.2}$$

$$\frac{\partial}{\partial t}T_n^m = \frac{1}{4\pi}\int_{-1}^{1}\int_{0}^{2\pi}(F_T+P_T)\,P_n^m(\mu)\,e^{-im\lambda}d\lambda d\mu \quad + (K_T)_n^m \tag{2.3.2.3}$$

$$\frac{\partial}{\partial t}(q_v)_n^m = \frac{1}{4\pi}\int_{-1}^{1}\int_{0}^{2\pi}(F_{q_v}+P_{q_v})\,P_n^m(\mu)\,e^{-im\lambda}d\lambda d\mu \quad + (K_{q_v})_n^m \tag{2.3.2.4}$$

$$\frac{\partial}{\partial t}(q_w)_n^m = \frac{1}{4\pi}\int_{-1}^{1}\int_{0}^{2\pi}(F_{q_w}+P_{q_w})\,P_n^m(\mu)\,e^{-im\lambda}d\lambda d\mu \quad + (K_{q_w})_n^m \tag{2.3.2.5}$$

and

$$\frac{\partial}{\partial t}(lnp_s)_n^m = \frac{1}{4\pi}\int_{-1}^{1}\int_{0}^{2\pi}F_P P_n^m(\mu)\,e^{-im\lambda}d\lambda d\mu \tag{2.3.2.6}$$

where F_U, F_V and G are given by (2.2.19) - (2.2.21), and

$$F_T = -\frac{U}{a(1-\mu^2)}\frac{\partial T}{\partial\lambda} - \frac{V}{a}\frac{\partial T}{\partial\mu} - \dot{\eta}\frac{\partial T}{\partial\eta} + \frac{\kappa T_v\omega}{(1+(\delta-1)q_v)p} \tag{2.3.2.7}$$

$$F_{q_v} = -\frac{U}{a(1-\mu^2)}\frac{\partial q_v}{\partial\lambda} - \frac{V}{a}\frac{\partial q_v}{\partial\mu} - \dot{\eta}\frac{\partial q_v}{\partial\eta} \tag{2.3.2.8}$$

$$F_{q_w} = -\frac{U}{a(1-\mu^2)}\frac{\partial q_w}{\partial\lambda} - \frac{V}{a}\frac{\partial q_w}{\partial\mu} - \dot{\eta}\frac{\partial q_w}{\partial\eta} \tag{2.3.2.9}$$

$$F_p = -\frac{1}{p_S}\int_0^1 \nabla \cdot \left(\underline{v}_h \frac{\partial p}{\partial \eta}\right) d\eta \tag{2.3.2.10}$$

Equations (2.3.2.3) - (2.3.2.6) are in the form used in the model. The corresponding forms for the vorticity and divergence equations are obtained from (2.3.2.1) and (2.3.2.2) by integration by parts and use of (2.3.1.13):

$$\frac{\partial}{\partial t}\xi_n^m = \frac{1}{4\pi a}\int_{-1}^{1}\int_0^{2\pi} (1-\mu^2)^{-1}\,[im\,(F_V + P_V)\,P_n^m(\mu) - (F_U + P_U)\,H_n^m(\mu)]\,e^{-im\lambda}d\lambda d\mu$$

$$+ (K_\xi)_n^m \tag{2.3.2.11}$$

$$\frac{\partial}{\partial t}D_n^m = \frac{1}{4\pi a}\int_{-1}^{1}\int_0^{2\pi} (1-\mu^2)^{-1}\,[im\,(F_U + P_U)\,P_n^m(\mu) + (F_V + P_V)\,H_n^m(\mu)]\,e^{-im\lambda}d\lambda d\mu$$

$$+\frac{n\,(n+1)}{4\pi a^2}\int_{-1}^{1}\int_0^{2\pi} G\,P_n^m(\mu)\,e^{-im\lambda}d\lambda d\mu \quad + (K_D)_n^m \tag{2.3.2.12}$$

where $H_n^m(\mu)$ is given by (2.3.1.20).

An outline of the model's computation of spectral tendencies may now be given. First, a grid of points covering the sphere is defined. Using the basic definition of the spectral expansions (2.3.1.1) and equations (2.3.1.14) - (2.3.1.19), values of ξ, D, U, V, T, q_v, q_w and ln p_S are calculated at the gridpoints, as also are the derivatives

$$\frac{\partial T}{\partial \lambda}, \frac{\partial T}{\partial \mu}, \frac{\partial q_v}{\partial \lambda}, \frac{\partial q_v}{\partial \mu}, \frac{\partial q_w}{\partial \lambda}, \frac{\partial q_w}{\partial \mu}, \frac{\partial}{\partial \lambda} ln p_S \text{ and } \frac{\partial}{\partial \mu} ln p_S$$

using (2.3.1.9) and (2.3.1.10). The resulting gridpoint values are sufficient to calculate gridpoint values of F_U, F_V, F_T, F_{q_v}, F_{q_w}, F_p and G, together with the parametrized tendencies P_U, P_V, P_T, P_{q_v} and P_{q_w}, since prognostic surface fields associated with the parametrization are defined and updated on the same grid. The integrands of the prognostic equations (2.3.2.11), (2.3.2.12), (2.3.2.3) - (2.3.2.6) are thus known at each gridpoint, and spectral tendencies are calculated by numerical quadrature.

The grid on which the calculations are performed is chosen to give an exact (given the spectral truncation of the fields, and within round-off error) contribution to spectral tendencies from quadratic non-linear terms. The integrals with respect to λ involve the product of three trigonometric functions, and as shown by Machenhauer and Rasmussen (1972) they may be evaluated exactly using a regularly-spaced grid of at least $3 \cdot M + 1$ points. For the latitudinal integrals, Eliasen et al. (1970) showed that quadratic non-linear terms lead to integrands which are polynomials in μ of a certain order.

They may thus be computed exactly using Gaussian quadrature (e.g. Krylov, 1962), with points located at the (approximately equally-spaced) latitudes which satisfy $P^0_{N_G}(\mu) = 0$, for sufficiently large integer N_G. These latitudes form what are referred to as the "Gaussian latitudes".

In order to find the necessary number of Gaussian latitudes for the triangular truncation, and from the exactness condition for the Gaussian integration it may be shown that the number of Gaussian latitudes N_G must fulfil the following condition:

$$N_G \geq \frac{3 \cdot K + 1}{2} \quad .$$

An asymptotic property of the Legendre Functions which may be derived directly from the definition (2.3.1.2) is

$$P^m_n(\mu) \sim (1-\mu^2)^{m/2} \text{as } (\mu \to \pm 1) .$$

Thus for large m the functions become vanishingly small as the poles are approached, and the contributions to the integrals (2.3.2.1) - (2.3.2.6) from polar regions become less than the unavoidable round-off error for sufficiently large zonal wavenumbers.

2.4 VERTICAL DISCRETIZATION

2.4.1 The hybrid vertical representation

To represent the vertical variation of the dependent variables ξ, D, T, q_v and q_w the atmosphere is divided into $NLEV$ layers as illustrated in Figure 3. These layers are defined by the pressures of the interfaces between them (the"half levels"), and these pressures are given by

$$p_{k+1/2} = A_{k+1/2} + B_{k+1/2}\ p_S \tag{2.4.1.1}$$

for $k = 0, 1, 2 \ldots NLEV$. The $A_{k+1/2}$ and $B_{k+1/2}$ are constants whose values effectively define the vertical coordinate. Necessary values are

$$A_{1/2} = B_{1/2} = A_{NLEV+1/2} = 0 \quad \text{and} \quad B_{NLEV+1/2} = 1 \tag{2.4.1.2}$$

The usual sigma coordinate is obtained as the special case

$$A_{k+1/2} = 0 \ , \ k = 0, 1, 2, \ldots\ldots, \text{NLEV} \tag{2.4.1.3}$$

This form of hybrid coordinate has been chosen because it is particularly efficient from a computational viewpoint. It also allows a simple direct control over the "flattening" of coordinate surfaces as pressure decreases, since the $A's$ and $B's$ may be determined by specifying the distribution of half-level pressures for a typical sea-level surface pressure and for a surface pressure typical of the lowest expected to be attained in the model. Coordinate surfaces are surfaces of constant pressure at levels where $B_{k+1/2} = 0$.

The prognostic variables ξ, D, T, q_v and q_w are represented by their values at intermediate ("full-level") pressures, p_k. Values for p_k are not explicitly required by the model's vertical finite-difference scheme, which is described in the following section, but they are required by parameterization schemes, in the creation of initial data, and in the interpolation to pressure levels that forms part of the post-processing. Alternative forms for p_k have been discussed by Simmons and Burridge (1981) and Simmons and Strüfing (1981). Little sensitivity has been found, and the simple form

$$p_k = \frac{1}{2}(p_{k+1/2} + p_{k-1/2}) \tag{2.4.1.4}$$

has been adopted, where half-level values are as given by (2.4.1.1).

The explicit relationship between p and p_S defined for model half levels implicitly determines a vertical coordinate η. The model formulation is in fact such that this coordinate need not be known explicitly, as demonstrated in the following section. However, it is computationally convenient to define η for the radiative parametrization and for the vertical interpolation used in the post-processing. The half-level values are given by

$$\eta_{k+1/2} = \frac{A_{k+1/2}}{p_0} + B_{k+1/2} \tag{2.4.1.5}$$

where p_0 is constant pressure. From (2.4.1.1) it is seen that this coordinate is identical to the usual σ when $A_{k+1/2} = 0$, and in general equals σ when $p_0 = p_S$. $\eta = p/p_0$ at levels where coordinate surfaces are surfaces of constant pressure. Values of η between half-levels are given by linear interpolation :

$$\eta = \eta_{k+1/2} + \frac{(p - p_{k+1/2})\,(\eta_{k+1/2} - \eta_{k-1/2})}{(p_{k+1/2} - p_{k-1/2})} \quad \text{for} \quad (p_{k-1/2} \le p \le p_{k+1/2}) \qquad (2.4.1.6)$$

A 19-layer version is used operationally, and the corresponding values of the $A_{k+1/2}$ and $B_{k+1/2}$ are given in Table 1. The distribution of full-level pressures is shown in Figure 4. The top two layers are at constant pressures, and the lowest two layers are pure sigma layers. The value of p_0 used for the definition of η is the reference sea-level pressure of 101325 Pa.

Table 1 Vertical-coordinate parameters of the 19-layer ECHAM3 model

k	$A_{k+1/2}\,(Pa)$	$B_{k+1/2}$
0	0.000000	0.0000000000
1	2000.000000	0.0000000000
2	4000.000000	0.0000000000
3	6046.110595	0.0003389933
4	8267.927560	0.0033571866
5	10609.513232	0.0130700434
6	12851.100169	0.0340771467
7	14698.498086	0.0706498323
8	15861.125180	0.1259166826
9	16116.236610	0.2011954093
10	15356.924115	0.2955196487
11	13621.460403	0.4054091989
12	11101.561987	0.5249322235
13	8127.144155	0.6461079479
14	5125.141747	0.7596983769
15	2549.969411	0.8564375573
16	783.195032	0.9287469142
17	0.000000	0.9729851852
18	0.000000	0.9922814815
19	0.000000	1.0000000000

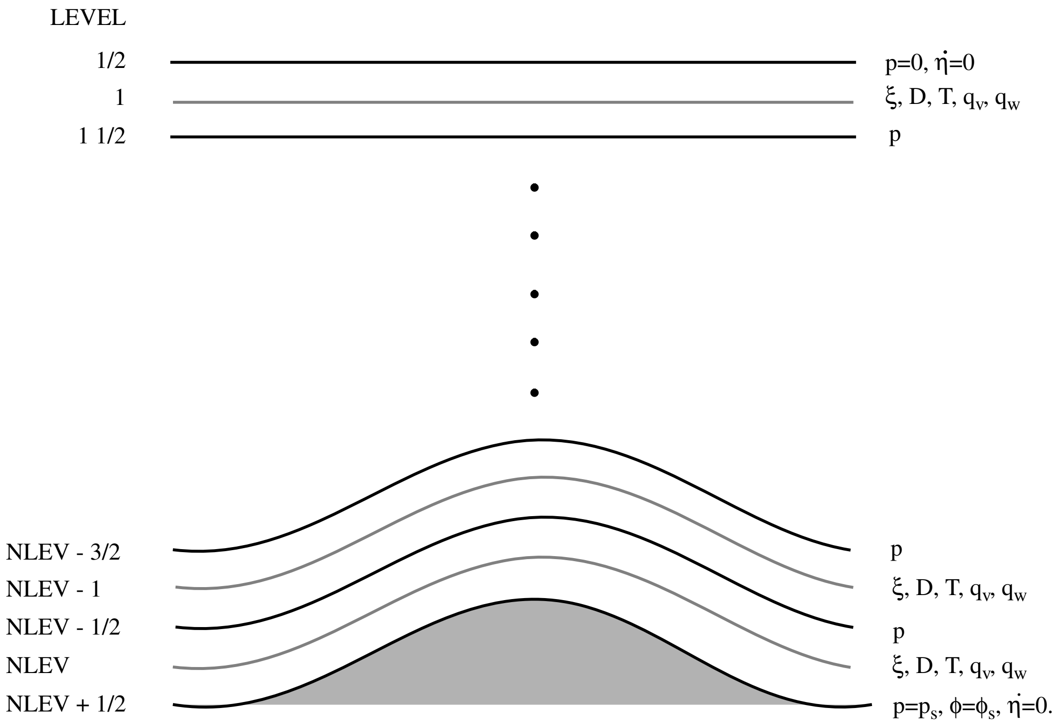


Figure 3 Vertical distribution of variables

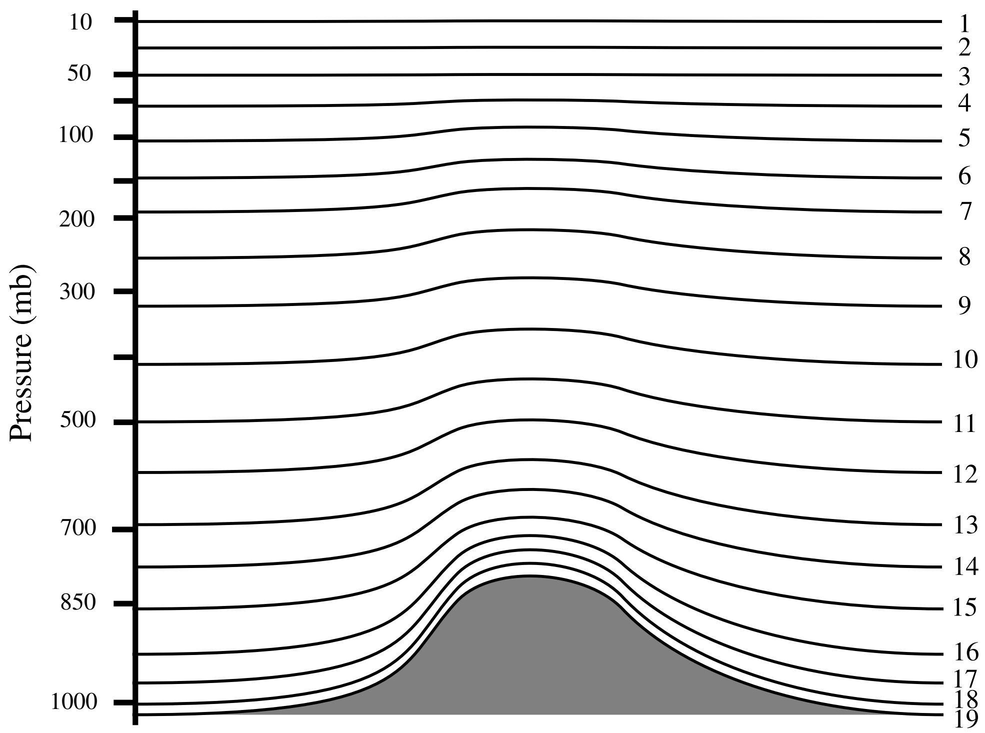


Figure 4 Distribution of full-level pressures in the 19-level ECHAM3 model

2.4.2 The vertical finite-difference scheme

The vertical finite-difference scheme is a generalization to the hybrid coordinate with form (2.4.1.1) of the scheme adopted in the first operational ECMWF model (Burridge and Haseler, 1977), apart from a small modification concerned with the conservation of angular momentum. The generalized scheme has been discussed by Simmons and Burridge (1981) and by Simmons and Strüfing (1981), and the presentation here is restricted to a prescription of the finite-difference forms of the various terms of the continuous equations that involve η.

a) The surface-pressure tendency

The finite-difference analogue of (2.2.11) is

$$\frac{\partial}{\partial t}lnp_S = -\frac{1}{p_S}\sum_{k=1}^{NLEV}\nabla\cdot(\underset{\sim}{v}_k\Delta p_k) \tag{2.4.2.1}$$

where the subscript "k" denotes a value for the k-th layer, and

$$\Delta p_k = p_{k+1/2} - p_{k-1/2} \tag{2.4.2.2}$$

From (2.4.1.1) we obtain

$$\frac{\partial}{\partial t}lnp_S = -\sum_{k=1}^{NLEV}\{\frac{1}{p_S}D_k\Delta p_k + (\underset{\sim}{v}_k\nabla lnp_S)\,\Delta B_k\} \tag{2.4.2.3}$$

$$\text{where}\quad \Delta B_k = B_{k+1/2} - B_{k-1/2} \tag{2.4.2.4}$$

b) The continuity equation

(2.2.10) gives

$$\left(\dot{\eta}\frac{\partial p}{\partial\eta}\right)_{k+1/2} = -\frac{\partial p_{k+1/2}}{\partial t} - \sum_{j=1}^{k}\nabla\,(\underset{\sim}{v}_j\Delta p_j) \tag{2.4.2.5}$$

and from (2.4.1.1)

$$\left(\dot{\eta}\frac{\partial p}{\partial\eta}\right)_{k+1/2} = -p_S\left[B_{k+1/2}\frac{\partial}{\partial t}lnp_S + \sum_{j=1}^{k}\{\frac{1}{p_S}D_j\Delta p_j + (\underset{\sim}{v}_j\nabla lnp_S)\,\Delta B_j\}\right] \tag{2.4.2.6}$$

where $\frac{\partial}{\partial t}lnp_S$ is given by (2.4.2.3).

c) Vertical advection

Given $\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right)_{k+1/2}$ computed from (2.4.2.6), vertical advection of a variable X is given by

$$\left(\dot{\eta}\frac{\partial X}{\partial \eta}\right)_k = \frac{1}{2\Delta p_k}\left\{ \left(\dot{\eta}\frac{\partial p}{\partial \eta}\right)_{k+1/2}(X_{k+1}-X_k) + \left(\dot{\eta}\frac{\partial p}{\partial \eta}\right)_{k-1/2} \cdot (X_k - X_{k-1}) \right\} \qquad (2.4.2.7)$$

This form ensures that there is no spurious source or sink of kinetic energy, potential energy, moisture or cloud water due to the finite-difference representation of vertical advection.

d) The hydrostatic equation

The form chosen for the finite-difference analogue of (2.2.7) is

$$\phi_{k+1/2} - \phi_{k-1/2} = -R_d \cdot (T_v)_k \cdot ln\left(\frac{p_{k+1/2}}{p_{k-1/2}}\right) \qquad (2.4.2.8)$$

which gives

$$\phi_{k+1/2} = \phi_S + \sum_{j=k+1}^{NLEV} R_d \cdot (T_v)_j \cdot ln\left(\frac{p_{j+1/2}}{p_{j-1/2}}\right) \qquad (2.4.2.9)$$

Full level values of geopotential are given by

$$\phi_k = \phi_{k+1/2} + \alpha_k \cdot R_d \cdot (T_v)_k \quad , \qquad (2.4.2.10)$$

$$\text{where } \alpha_1 = ln 2 \qquad (2.4.2.11)$$

and, for $k > 1$,

$$\alpha_k = 1 - \frac{p_{k-1/2}}{\Delta p_k} \cdot ln\left(\frac{p_{k+1/2}}{p_{k-1/2}}\right) \qquad (2.4.2.12)$$

Reasons for this particular choice of the α_k are given below.

e) The pressure gradient term

It is shown by Simmons and Strüfing (1981) that if the geopotential is given by (2.4.2.10), the form

$$R_d \cdot (T_v \cdot \nabla lnp)_k = \frac{R_d \cdot (T_v)_k}{\Delta p_k}\left\{ \left(ln\frac{p_{k+1/2}}{p_{k-1/2}}\right) \cdot \nabla p_{k-1/2} + \alpha_k \cdot \nabla(\Delta p_k) \right\} \qquad (2.4.2.13)$$

for the pressure-gradient term ensures no spurious source or sink of angular momentum due to the vertical differencing. This expression is adopted in the model, but with the α_k given by (2.4.2.12) for all k. This ensures that the pressure-gradient term reduces to the familiar form $R_d (T_v)_k \nabla ln\ p_S$ in the case of sigma coordinates, and the angular momentum conserving property of the scheme still holds in the case in which the first half-level below $p = 0$ is a surface of constant pressure. The choice $\alpha_1 = 1$ in the hydrostatic equation would have given angular momentum conservation in general, but a geopotential ϕ_1 inappropriate to the pressure-level $p = p_1 = \Delta p/2$. If, alternatively, ϕ_1 were to be interpreted not as a value for a particular level, but rather the mass-weighted layer-mean value, then the choice $\alpha_1 = 1$ would be appropriate.

It is shown by Simmons and Chen (1991) that the form (2.4.2.13) can be significantly improved, with benefit particularly in regions of steep terrain, if T_v is replaced by its deviation from a reference state,

$$\tilde{T}_v = T_v - T_{00}\left(\frac{p}{p_0}\right)^{\beta} \qquad (2.4.2.14)$$

where $\beta = \gamma \cdot R_d / g$, $p_0 = 1013.25\ \text{hPa}$, $T_{00} = 288\ \text{K}$ and $\gamma = 6.5\ \text{K/km}$. The reference temperature (2.4.2.14) is based on the tropospheric part of the ICAO (1964) standard atmosphere with a uniform lapse rate γ.

Using the form (2.4.1.1) for the half-level pressures (2.4.2.13) may be written

$$R_d \cdot (\tilde{T}_v \cdot \nabla lnp)_k = \frac{R_d \cdot (\tilde{T}_v)_k}{\Delta p_k}\left\{\Delta B_k + C_k \cdot \frac{1}{\Delta p_k} \cdot \left(ln\frac{p_{k+1/2}}{p_{k-1/2}}\right)\right\} \nabla p_S \qquad (2.4.2.15)$$

where

$$C_k = A_{k+1/2} \cdot B_{k-1/2} - A_{k-1/2} \cdot B_{k+1/2} \qquad (2.4.2.16)$$

The modified form (2.4.2.15) finally requires a reformulation of the surface geopotential according to

$$\phi_S = g \cdot z_S + \frac{R_d \cdot T_{00}}{\beta} \cdot \left(\frac{p_S}{p_0}\right)^{\beta} \qquad (2.4.2.17)$$

f) <u>Energy-conversion term</u>

To obtain a form for the term $\kappa \cdot T_v \cdot \omega / (1 + (\delta - 1)\, q_v)$ in (2.2.3) we use (2.2.8) to write

$$\left(\frac{\kappa \cdot T_v \cdot \omega}{(1 + (\delta - 1)\, q_v)\, p}\right)_k = \frac{\kappa \cdot (T_v)_k}{1 + (\delta - 1)\ (q_v)_k}\left(\frac{\omega}{p}\right)_k \qquad (2.4.2.18)$$

where

$$\left(\frac{\omega}{p}\right)_k = -\frac{1}{p}\int_0^{\eta_k} \nabla \cdot \left(\underset{\sim}{v} \cdot \frac{\partial p}{\partial \eta}\right) d\eta + (\underset{\sim}{v} \cdot \nabla lnp)_k \qquad (2.4.2.19)$$

An expression for $(\omega/p)_k$ is then determined by the requirement that the difference scheme conserves the total energy of the model atmosphere for adiabatic, frictionless motion. This is achieved by

(i) evaluating the first term on the right-hand side of (2.4.2.19) by

$$-\frac{1}{\Delta p_k}\left\{\left(ln\frac{p_{k+1/2}}{p_{k-1/2}}\right)\cdot\sum_{k=1}^{k-1}\nabla\cdot\left(\underset{\sim}{v}_j\cdot\Delta p_j\right)+\alpha_k\nabla\cdot\left(\underset{\sim}{v}_k\cdot\Delta p_k\right)\right\} \qquad (2.4.2.20)$$

where the α_k are as given by (2.4.2.11) and (2.4.2.12), and as in (2.4.2.3) and (2.4.2.5)

$$\nabla\cdot\left(\underset{\sim}{v}_k\cdot\Delta p_k\right) = D_k\cdot\Delta p_k+p_S\cdot\left(\underset{\sim}{v}_k\cdot\nabla lnp_s\right)\cdot\Delta B_k \qquad (2.4.2.21)$$

(ii) using the form of (2.4.2.15) to evaluate the second term on the right-hand side of (2.4.2.19)

$$\left(\underset{\sim}{v}\cdot\nabla lnp\right)_k = \frac{p_S}{\Delta p_k}\cdot\left\{\Delta B_k+C_k\cdot\frac{1}{\Delta p_k}\cdot\left(ln\frac{p_{k+1/2}}{p_{k-1/2}}\right)\right\}\cdot\underset{\sim}{v}_k\cdot\nabla lnp_S \qquad (2.4.2.22)$$

2.5 TIME INTEGRATION SCHEME

A semi-implicit time scheme is used for equations of divergence, temperature and surface pressure, based on the work of Robert et al. (1972). The growth of spurious computational modes is inhibited by a time filter (Asselin, 1972). In addition, a semi-implicit method for the zonal advection terms in the vorticity and moisture equations is used, following results obtained by Robert (1981, 1982). He showed that in a semi-implicit shallow water equation model the time-step criterion was determined by the explicit treatment of the vorticity equation. Facilities also exist for selective damping of short model scales to allow use of longer timesteps. These are incorporated within the horizontal diffusion routines of the model, and are described in section 2.6.
The semi-implicit schemes are formally given by:

$$\delta_t \xi = ZT \qquad - \frac{1}{2a} \beta_{ZQ} \frac{U_r(\mu)}{(1-\mu^2)} \frac{\partial}{\partial \lambda} \Delta_{tt} \xi \tag{2.5.1}$$

$$\delta_t q_v = Q_v T \qquad - \frac{1}{2a} \beta_{ZQ} \frac{U_r(\mu)}{(1-\mu^2)} \frac{\partial}{\partial \lambda} \Delta_{tt} q_v \tag{2.5.2}$$

$$\delta_t q_w = Q_w T \qquad - \frac{1}{2a} \beta_{ZQ} \frac{U_r(\mu)}{(1-\mu^2)} \frac{\partial}{\partial \lambda} \Delta_{tt} q_w \tag{2.5.3}$$

$$\delta_t D = DT - \nabla^2 G - \frac{1}{2} \beta_{DT} \nabla^2 \{ \gamma \Delta_{tt} T + R_d T_r \Delta_{tt} ln p_s \} \tag{2.5.4}$$

$$\delta_t T = TT \qquad - \frac{1}{2} \beta_{DT} \tau \Delta_{tt} D \tag{2.5.5}$$

$$\delta_t ln p_s = PT \qquad - \frac{1}{2} \beta_{DT} \nu \Delta_{tt} D \tag{2.5.6}$$

Here the terms ZT, $Q_v T$, $Q_w T$, G, TT, PT represent those on the right-hand sides of equations (2.2.17), (2.2.4), (2.2.5), (2.2.18), (2.2.3) and (2.2.11), apart from the diffusion terms, which are neglected here. Adiabatic components are evaluated at the current time, t, and parametrized components are generally evaluated using values of fields at the previous timestep, $t - \Delta t$. The treatment of diffusion terms is described in the following section.

The remaining terms on the right-hand sides of (2.5.1) - (2.5.6) are corrections associated with the semi-implicit time schemes, and are discussed more fully below. The operators δ_t and Δ_{tt} are given by

$$\delta_t X = (X^+ - X_f^-) / (2\Delta t) \tag{2.5.7}$$

$$\Delta_{tt} X = (X^+ + X_f^- - 2X) \tag{2.5.8}$$

where X represents the value of a variable at time t, X^+ the value at time $t + \Delta t$, and X_f^- a filtered value at time $t - \Delta t$. A further operator that will be used is

$$\tilde{\Delta}_{tt} X = X_f^- - 2X \tag{2.5.9}$$

The time filtering is defined by

$$X_f = X + \varepsilon_f (X_f^- - 2X + X^+) \tag{2.5.10}$$

and it is computationally convenient to split it into two parts;

$$\tilde{X}_f = X + \varepsilon_f (X_f^- - 2X) \tag{2.5.11}$$

$$X_f = \tilde{X}_f + \varepsilon_f X^+ \tag{2.5.12}$$

The timestep Δt depends on the resolution and is specified as follows:

T21	Δt = 40 minutes
T42	Δt = 24 minutes
T63	Δt = 15 minutes
T106	Δt = 12 minutes,

and $\varepsilon_f = 0.1$ independent of the resolution.

a) <u>The semi-implicit treatment of vorticity and moisture (water vapour and cloud water)</u>

Referring to equations (2.5.1), (2.5.2) and (2.5.3), an explicit treatment of the vorticity and moisture equations is obtained by setting $\beta_{ZQ} = 0$. Otherwise $\beta_{ZQ} = 1$ and $U_r(\mu)$ is a zonally-uniform reference zonal velocity, multiplied by $cos\theta$. Terms describing advection by this reference velocity are represented implicitly by the arithmetic mean of values at times $t + \Delta t$ and $t - \Delta t$, while the remainder of the tendencies are represented explicitly by values at time t. $U_r(\mu)$ may vary in the vertical.

Because of the use of integration by parts in the derivation of the prognostic equation (2.3.2.11) for the spectral coefficients of vorticity, it is necessary to treat the vorticity and moisture equations separately. Considering first the moisture equations, we obtain from (2.5.2), (2.5.3) and (2.5.7) - (2.5.9):

$$(1 + 2\Delta t\ \alpha(\mu)\frac{\partial}{\partial\lambda})\, q_v^+ = q_{vf}^- + 2\Delta t\ Q_v T - 2\Delta t\ \alpha(\mu)\frac{\partial}{\partial\lambda}\tilde{\Delta}_{tt} q_v \tag{2.5.13}$$

$$(1 + 2\Delta t\ \alpha(\mu)\frac{\partial}{\partial\lambda})\, q_w^+ = q_{wf}^- + 2\Delta t\ Q_w T - 2\Delta t\ \alpha(\mu)\frac{\partial}{\partial\lambda}\tilde{\Delta}_{tt} q_w \tag{2.5.14}$$

where

$$\alpha(\mu) = \frac{1}{2a}\beta_{ZQ}\frac{U_r(\mu)}{(1-\mu^2)} \tag{2.5.15}$$

Transforming to Fourier space then gives

$$q_{vm}^+ = b_m(\mu)\ \{ (q_{vf}^- + 2\Delta t\ Q_v T)_m - 2\ im\ \Delta t\ \alpha(\mu)\,\tilde{\Delta}_{tt} q_{vm} \} \tag{2.5.16}$$

$$q_{wm}^+ = b_m(\mu)\ \{ (q_{wf}^- + 2\Delta t\ Q_w T)_m - 2\ im\ \Delta t\ \alpha(\mu)\,\tilde{\Delta}_{tt} q_{wm} \} \tag{2.5.17}$$

where

$$b_m(\mu) = (1 + 2\ im\ \Delta t\ \alpha(\mu))^{-1} \tag{2.5.18}$$

New values $(q_n^m)_v^+$ and $(q_n^m)_w^+$ of the spectral coefficients of q_v and q_w are then computed by Gaussian integration as described in 2.3.2 .

For the vorticity equation, (2.2.17) is used to write

$$ZT = \frac{1}{a(1-\mu^2)} \frac{\partial}{\partial\lambda}(F_V + P_V) - \frac{1}{a}\frac{\partial}{\partial\mu}(F_U + P_U) \tag{2.5.19}$$

where the horizontal diffusion term has for convenience been neglected, since as specified in the following section it merely modifies the value of vorticity computed for time $t + \Delta t$. Proceeding as for the moisture equations, we obtain

$$\begin{aligned} \xi_m^+ = b_m(\mu) \Bigg[\left(\xi_f^- + \frac{2\ im\ \Delta t}{a(1-\mu^2)} (F_V + P_V) \right)_m & - 2\ im\ \Delta t\ \alpha(\mu)\tilde{\Delta}_{tt}\xi_m \\ & - \frac{2\ \Delta t}{a} \frac{\partial}{\partial\mu}(F_U + P_U)_m \Bigg] \end{aligned} \tag{2.5.20}$$

The factor $b_m(\mu)$ renders the right-hand side of this equation unsuitable for direct integration by parts, but a suitable form is found from the relation

$$b_m(\mu)\frac{\partial}{\partial\mu}(F_U + P_U) = \frac{\partial}{\partial\mu}\{b_m(\mu)\ (F_U + P_U)\} - c_m(\mu)\ (F_U + P_U) \tag{2.5.21}$$

where

$$c_m(\mu) = \frac{\partial}{\partial\mu} b_m(\mu) \tag{2.5.22}$$

This gives

$$\xi_m^+ = \tilde{Z}_{\lambda m}(\mu) + \frac{\partial}{\partial\mu}\tilde{Z}_{\mu m}(\mu) \tag{2.5.23}$$

where

$$\begin{aligned} \tilde{Z}_{\lambda m}(\mu) = {} & b_m(\mu)\ (\xi_f^-)_m \\ & + 2\Delta t \left(im b_m(\mu) \left[\frac{(F_V + P_V)_m}{a(1-\mu^2)} - \alpha(\mu)\tilde{\Delta}_{tt}\xi_m \right] + \frac{1}{a} c_m(\mu)\ (F_U + P_U)_m \right) \end{aligned} \tag{2.5.24}$$

and

$$\tilde{Z}_{\mu m}(\mu) = -\frac{2\Delta t}{a} b_m(\mu)\ (F_U + P_U)_m \tag{2.5.25}$$

New values $(\xi_n^m)^+$ are obtained from (2.5.23) by Gaussian quadrature, using integration by parts as illustrated by (2.3.2.1) and (2.3.2.11) for the continuous form of the equations.

In ECHAM3, $\beta_{ZQ} = 1$. $U_r(\mu)$ is the arithmetic mean of the maximum and minimum velocities multiplied by $cos\theta$, as computed for each latitude and model level at timestep $t-\Delta t$. Different values are thus used for different levels.

b) <u>The semi-implicit treatment of divergence, temperature and surface pressure</u>

Referring to equations (2.5.4) - (2.5.6), an explicit treatment of the divergence, temperature and surface pressure equations is obtained by setting $\beta_{DT} = 0$. For $\beta_{DT} = 1$, the nature of the semi-implicit correction is such that gravity wave terms for small amplitude motion about a basic state with isothermal temperatue T_r and surface pressure p_r are treated implicitly by the arithmetic mean of values at times $t+\Delta t$ and $t-\Delta t$, while the remainder of tendencies are represented explicitly by values at time t. The choice of an isothermal reference temperature is governed by considerations of the stability of the semi-implicit time scheme (Simmons et al, 1978), while the appropriate choice of p_r for the hybrid vertical coordinate is discussed by Simmons and Burridge (1981) and Simmons and Strüfing (1981).

γ, τ and ν in equations (2.5.4) - (2.5.6) are operators obtained from linearizing the finite-difference forms specified in 2.3.2 about the reference state (T_r, p_r). Their definitions are

$$(\gamma T)_k = \alpha_k^r R_d T_k + \sum_{j=k+1}^{NLEV} R_d T_j ln\left(\frac{p^r_{j+1/2}}{p^r_{j-1/2}}\right) \tag{2.5.26}$$

$$(\tau D)_k = \kappa T_r \left\{ \frac{1}{\Delta p_k^r}\left(ln\frac{p^r_{j+1/2}}{p^r_{j-1/2}}\right) S^r_{k-1/2} + \alpha_k^r D_k \right\} \tag{2.5.27}$$

and

$$\nu D = \frac{1}{p_r} S^r_{NLEV+1/2} \tag{2.5.28}$$

where

$$\begin{aligned} p^r_{k+1/2} &= A_{k+1/2} + p_r B_{k+1/2} \\ \Delta p_k^r &= p^r_{k+1/2} - p^r_{k-1/2} \\ S^r_{k+1/2} &= \sum_{j=1}^{k} D_j \Delta p_j^r \end{aligned} \tag{2.5.29}$$

and the α_k^r are defined by (2.4.2.11) and (2.4.2.12), but with half-level pressures replaced by reference values $p^r_{k+1/2}$.

Expanding (2.5.4) - (2.5.6) using (2.5.7) and (2.5.8), and writing l to denote $ln\ p'_S$, we obtain

$$D^+ = D_f^- + 2\Delta t(DT) - 2\Delta t \nabla^2 \{ G + \frac{1}{2}\beta_{DT}[\gamma(T^+ + T_f^- - 2T) + R_d T_r (l^+ + l_f^- - 2l)] \} \tag{2.5.30}$$

$$T^+ = T_1 - \Delta t \ \beta_{DT} \tau D^+ \tag{2.5.31}$$

and

$$l^+ = l_1 - \Delta t \ \beta_{DT} \nu D^+ \tag{2.5.32}$$

where

$$T_1 = T_f^- + 2\Delta t \, (TT) - \Delta t \ \beta_{DT} \tau \tilde{\Delta}_{tt} D \tag{2.5.33}$$

and

$$l_1 = l_f^- + 2\Delta t \, (PT) - \Delta t \ \beta_{DT} \nu \tilde{\Delta}_{tt} D \tag{2.5.34}$$

Substituting (2.5.31) and (2.5.32) into (2.5.30) then gives

$$(1 - \Gamma \nabla^2) \, D^+ = DT' \tag{2.5.35}$$

where

$$\Gamma = (\beta_{DT})^2 \, (\Delta t)^2 \, (\gamma \tau + R_d T_r \nu) \tag{2.5.36}$$

$$DT' = D_f^- + 2\Delta t \, (DT) + \nabla^2 R = \tilde{D}_\lambda + \tilde{D}_\mu + \nabla^2 R \tag{2.5.37}$$

with

$$\tilde{D}_\lambda = D_f^- + \frac{2\Delta t}{a \, (1 - \mu^2)} \frac{\partial}{\partial \lambda} (F_U + P_U) \tag{2.5.38}$$

$$\tilde{D}_\mu = \frac{2\Delta t}{a} \frac{\partial}{\partial \mu} (F_V + P_V) \tag{2.5.39}$$

and

$$R = -2\Delta t \, \{ G + \frac{B_{DT}}{2} \, (\gamma T_2 + R_d T_r l_2) \, \} \tag{2.5.40}$$

Here

$$T_2 = T_1 + T_f^- - 2T \tag{2.5.41}$$

$$l_2 = l_1 + l_f^- - 2l \tag{2.5.42}$$

The sequence of these semi-implicit calculations in the model is thus as follows. The expressions (2.5.33), (2.5.34) and (2.5.40) - (2.5.42) are computed on the Gaussian grid to form the gridpoint values of R. The spectral expansion of DT' is then derived by Gaussian quadrature, using integration by parts as illustrated by (2.3.2.2) and (2.3.2.12) for the continuous form of the equations. Since

$$\{ \, (1 - \Gamma \nabla^2) \, D^+ \}_n^m = (1 + \frac{n \, (n+1)}{a^2} \Gamma) \ (D^+)_n^m \quad , \tag{2.5.43}$$

the spectral coefficients of divergence at time $t + \Delta t$ are given from (2.3.32) by

$$(D^+)_n^m = \left(1 + \frac{n(n+1)}{a^2}\Gamma\right)^{-1} (DT')_n^m \quad , \tag{2.5.44}$$

where this operation involves, for each (m, n), multiplication of the vector of $NLEV$ values of $(DT')_n^m$ by a pre-computed $NLEV \times NLEV$ matrix whose elements are independent of time and determined by writing the operators γ, τ and ν in matrix and vector form. Finally, (2.5.31) and (2.5.32) are applied in spectral space to compute spectral coefficients of T and $ln\ p'_S$ at time $t + \Delta t$ in terms of the spectral coefficients of T_1 and l_1 (again determined by Gaussian quadrature) and those of D^+. In ECHAM3, $\beta_{DT} = 0.75$, $T_r = 300\ \mathrm{K}$ and $p_r = 800\ \mathrm{hPa}$.

2.6 HORIZONTAL DIFFUSION

2.6.1 Basic scheme

While the original ECMWF model uses a ∇^4 horizontal diffusion, ECHAM3 uses a diffusion parameterization based on the ideas of Laursen and Eliasen (1989). The contribution to the spectral tendency of any prognostic variable X is

$$-k_X\ L_n\ X_n^m \tag{2.6.1.1}$$

With this term, calculated at the new time $t + \Delta t$, the diffusive contribution to the semi-implicit time stepping scheme is

$$X_n^m(t+\Delta t) = X_n^m(t-\Delta t) \cdot \{1 + 2\Delta t\ k_X\ L_n\}^{-1} \tag{2.6.1.2}$$

L_n is chosen such that large scales are not damped, while short waves can be damped selectively,

$$L_n = \begin{cases} (n-n_*)^{\alpha} & \text{for} \quad (n > n_*) \\ 0 & \text{for} \quad (n \le n_*) \end{cases} \tag{2.6.1.3}$$

with $\alpha = 2$ for T21, and $\alpha = 4$ for T42 truncation. For both resolutions, we chose a cut-off wave number $n_* = 15$, i.e. only modes with $n \ge 16$ are damped.

The diffusion coefficient k_X varies for different variables and levels. While the lower levels (6 to 19) use the same k_X, the upper most levels are stronger damped,

$$k_X(l) = k_X \cdot fac(l) \tag{2.6.1.4}$$

where l denotes the vertical index. For $fac = 1$ and T21 resolution the damping time $1/(k_X \cdot L_n)$ of the shortest scale is 1.12 days for the vorticity, 0.22 days for the divergence and 5.59 days for temperature, humidity and cloud water, respectively. For T42 resolution the respective times are 0.30, 0.06 and 0.76 days. Figure 5 shows the damping times of the vorticity due to the horizontal diffusion as a function of the total wave number n and level l. Table 2 shows the amplification factors for the six uppermost levels.

Table 2 Vertical amplification factor for the horizontal diffusion

level l	factor fac
1	16
2	16
3	8
4	4
5	2
6	1

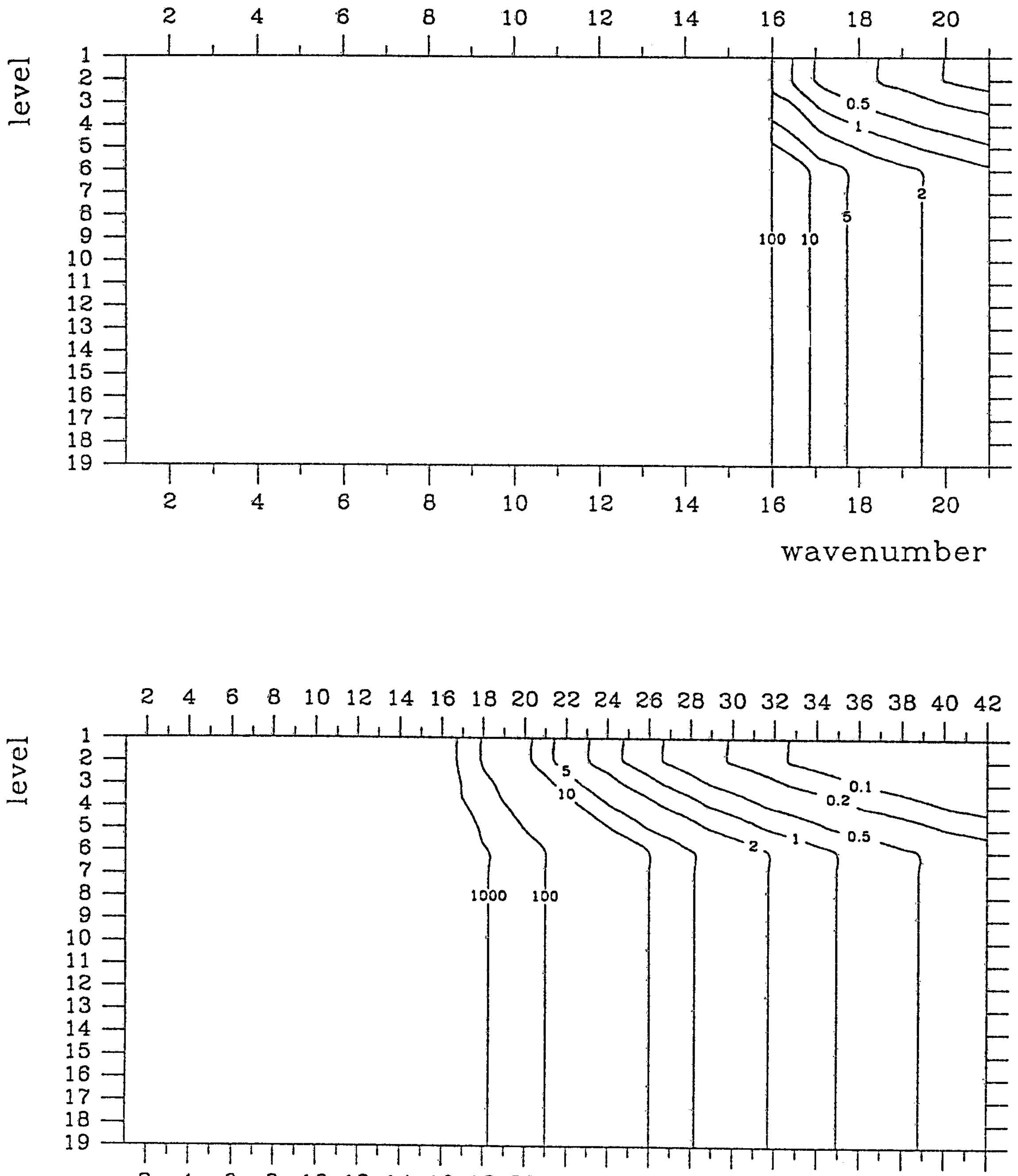


Figure 5 Damping times for the vorticity due to horizontal diffusion for T21 (upper panel) and T42 (lower panel) horizontal resolution. Contours for 0.1, 0.2, 0.5, 1, 2, 5, 10, 100, 1000 days.

A modified diffusion scheme is used for the temperature to avoid unrealistic warming of mountain tops and excessive summer precipitation associated with substantial mixing in the vicinity of steep mountain slopes. Only the deviation of the temperature field from the ICAO standard atmosphere is inserted into the diffusion procedure.

The diffusion coefficients were chosen such that the slope of the spectral kinetic energy is close to observations (see Figure 6). Moreover, as a result of the diffusion scheme together with a timestep of 40 /24/ 15/12 minutes for T21/T42/T63/T106 resolution, additional damping due to a violation of the CFL-criterion is considerably reduced as compared to the ECMWF model.

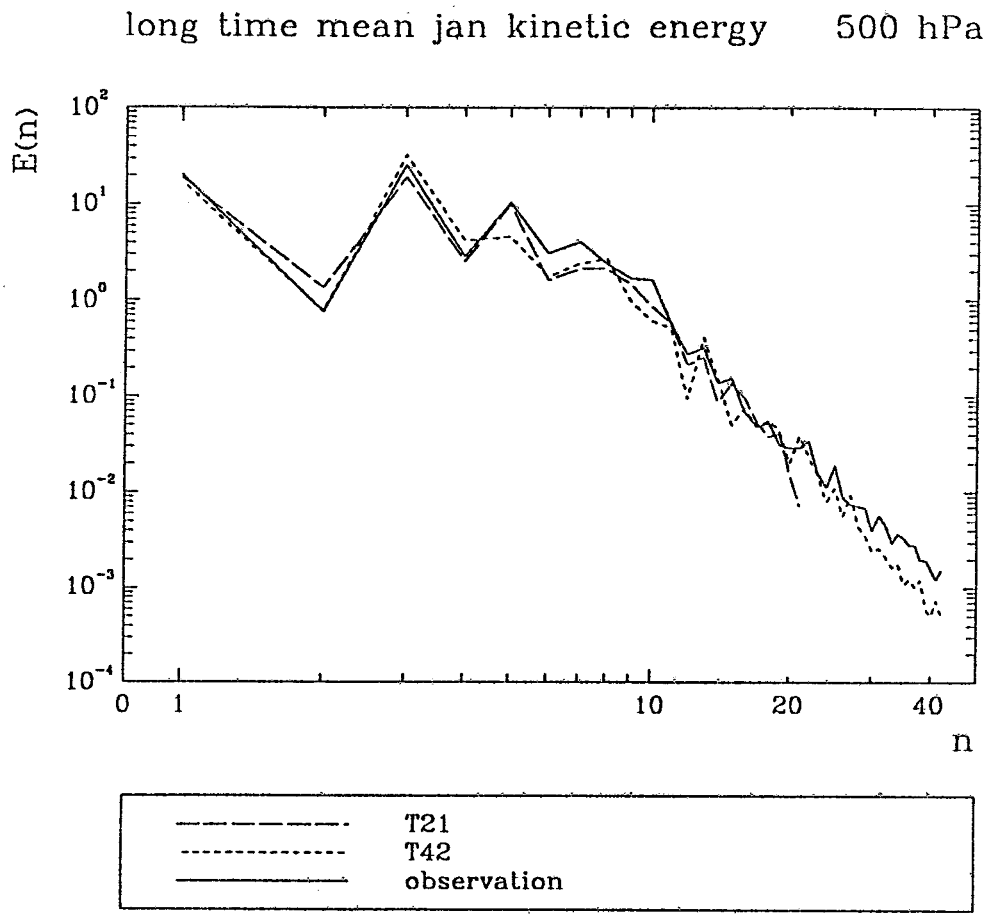


Figure 6 Mean kinetic energy spectra at 500 hPa during January for observations (ECMWF, mean of 1980 - 1989), for T21 resolution (mean of a 20 year control run with prescribed climatological sea surface temperature) and for T42 resolution (mean of a 10 year simulation with the observed sea surface temperatures of the years 1979 - 1988).

3. MODEL PHYSICS

3.1 INTRODUCTION

Processes associated with turbulent and convective transfer, condensation and radiation are important for the development of the atmospheric large-scale flow. Since these processes are related to spatially small or even molecular phenomena, they cannot be explicitly included in numerical models which only resolve scales larger than grid size in grid point models or the truncation wave number in spectral models.

Figure 7 shows the processes which are thought to be important for long term integrations. It shows how the different processes (printed within ellipses) depend on the variables given by the model (printed in rectangles).The thickness of the arrows represents qualitatively the importance of the interactions. For example, the effect of radiation on the ground temperature is very fast: It only takes a few hours for the sun to heat the ground. The relationship between radiation and the temperature of the air is much weaker: Condensation, diffusion and the flux of sensible heat from the ground change the air temperature more effectively. Any closed loop in the diagram (e.g. Temperature → Evaporation → Ground temperature → Sensible heat flux → Temperature) indicates a feed-back process. The reader can amuse himself by counting the number of closed loops in order to get a feeling for the complexity of the atmospheric system.

The effect of the sub-grid scale processes on the large-scale flow which we want to predict can only be considered by means of parameterisation, i.e. formulating the ensemble effect in terms of the resolved grid-scale variables.

Large efforts have been made in the recent past to design reliable parameterisation schemes for different processes. Based on a large amount of observational data the parametearisation of the turbulent fluxes in the planetary boundary layer has been considerably improved. A significant effort has been directed toward acccounting for vegetation effects on the surface fluxes. Progress has also been made in calculating the radiative fluxes, and a hierarchy of schemes differing with respect to the degree of approximation of the equations of radiative transfer were developed.

However, the parameterisation of convection is still problematic since the interaction of cumulus clouds with the large scale circulation is not sufficiently understood to decide on one of the existing schemes. Finally, there are also processes which have only occasionally been the subject of parameterisation studies as, mesoscale circulations, topographically induced circulations, thin layer clouds, coastal effects.

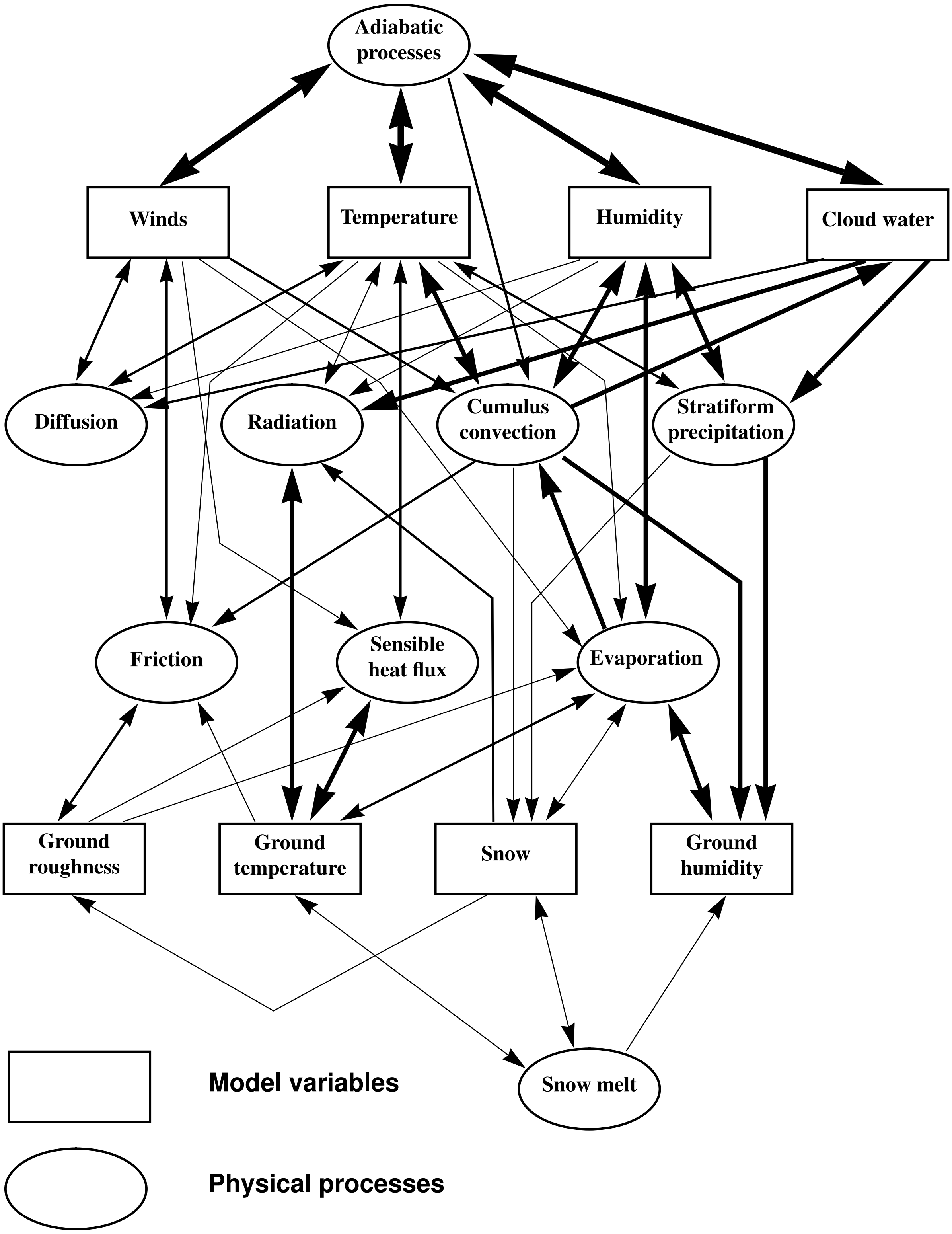


Figure 7 Schematic representation of the processes included in the ECHAM3 model.

The radiation scheme (section 3.2) is designed to take the cloud-radiation interaction into account in considerable detail. It allows partial cloud cover in any layer of the model, and considers multiple scattering. The effect of gases is subsequently taken into account. The reader can also refer to Hense et al. (1982), Rockel et al. (1991), and Eickerling (1989).

The dissipation processes are described in section 3.3 . The vertical eddy fluxes are simulated as a diffusive process. The dependence of the surface fluxes and diffusion coefficients on the stability of the atmosphere is based on the Monin-Obukhov similarity theory. The papers by Louis (1979) and Louis et al. (1982) may provide additional information.

An additional process by which momentum may be removed from the atmosphere is through the excitation of gravity waves when stabaly stratified flow interacts with the orography. This so-called "gravity-wave" drag (section 3.4) is parameterized using a modifed version of a scheme developed by Boer et al. (1984) and Palmer et al. (1986), and which uses directionally dependent sub-grid scale orographic variances (Miller et al., 1989).

The moist processes are separated into convective precipitation and stratiform precipitation. The convective processes are simulated by a mass flux scheme, in which the convection is driven by the large-scale convergence of moisture (Tiedtke, 1989). The stratiform cloud scheme is based upon a transport equation for cloud water including the formation of sub-grid scale clouds and a simplified treatment of the microphysical processes such as condensation, evaporation and precipitation formation by coalescence of cloud droplets and sedimentation of ice crystals (Roeckner et al., 1991).

Based mainly on Seller's ideas, (Sellers et al., 1986), the effect of vegetation on surface evapotranspiration has been accounted for. The dependence of sensible heat fluxes on snow coverage has been incorporated as well. The papers by Blondin and Böttger (1987) and Blondin (1989) give an overview of the surface parameterisation scheme.

Finally the ground processes include a prediction of the ground temperature, using a 5-level, finite-difference approximation of the diffusion equation. A prediction of soil moisture and snow cover is also included (Dümenil and Todini, 1992).

The horizontal diffusion, which is used in the operational model to keep the forecast from developing small scale noise, is included in the documentation of the adiabatic part (section 2.6).

3.2 RADIATION

3.2.1 Radiative heating

The radiative heating rate is computed as the divergence of net radiative fluxes F :

$$\left(\frac{\partial T}{\partial t}\right)_{rad} = -\frac{g}{C_{pd}} \cdot \frac{\partial F}{\partial p} \tag{3.2.1.1}$$

Section 3.2.2 describes the computation of the radiative fluxes. The solution of the radiative transfer equation to obtain the fluxes is unfortunately very expensive and is therefore performed only every two hours. To take into account the change in temperature and solar zenith angle between the times when the full radiation computation is performed, we define an effective emissivity ε_e and transmissivity τ_e of the model levels such that

$$F_T = \varepsilon_e \cdot \sigma \cdot T^4 \tag{3.2.1.2}$$

and

$$F_S = \tau_e \cdot S_0 \tag{3.2.1.3}$$

where F_T and F_S are the net thermal (long wave) and solar (short wave) fluxes respectively. σ is the Stefan-Boltzmann constant and S_0 is the solar flux at the top of the atmosphere. The values ε_e and τ_e are kept constant between the full radiation time steps and the net fluxes are recomputed at every time step, using (3.2.1.2) and (3.2.1.3) with the correct temperature and solar zenith angle.

3.2.2 The radiative transfer model

3.2.2.1 Introduction

The radiative transfer model is based on a two-stream approximation described by Kerschgens et al. (1978) and Zdunkowski et al. (1980). From the radiative transfer equations for the solar part of the spectrum, eqs. (3.2.2.1) and (3.2.2.2), and for the terrestrial part, eqs. (3.2.2.3) and (3.2.2.4), Hense et al. (1982) derived a broad-band formulation with spectral intervals listed in Table 3.

$$\frac{\partial}{\partial \delta} M^+ = [1 - \tilde{\omega}(1-\beta)] \frac{M^+}{\bar{\mu}} - \tilde{\omega}\beta \frac{M^-}{\delta} - \beta_0 \tilde{\omega} S_0 e^{-\delta/\mu_0} \tag{3.2.2.1}$$

$$\frac{\partial}{\partial \delta} M^- = \tilde{\omega}\beta \frac{M^+}{\delta} - [1 - \tilde{\omega}(1-\beta)] \frac{M^-}{\bar{\mu}} + (1-\beta_0)\tilde{\omega} S_0 e^{-\delta/\mu_0} \tag{3.2.2.2}$$

$$\frac{d}{d\delta} M^+ = \frac{1}{\bar{\bar{\mu}}} (M^+ - B) \tag{3.2.2.3}$$

$$\frac{d}{d\delta} M^- = \frac{1}{\bar{\bar{\mu}}} (M^- - B) \tag{3.2.2.4}$$

where M^+ and M^- are the upward and downward flux densities, δ is the optical depth, $\tilde{\omega}$ is the single scattering albedo, β and β_0 are the backscattering parameters for diffuse and direct radiation respectively, $1/\bar{\mu}$ is the diffusivity factor (2 for solar and 1.66 for terrestrial radiation), S_0 is the solar irradiance, μ_0 is the cosine of the solar zenith angle, and B is the Planck function.

Table 3 Spectral intervals and absorbers of the radiative transfer model

Terrestrial		**Solar**	
Spectral Interval(μm)	Absorber	Spectral Interval(μm)	Absorber
3.96 - 7.98	H_2O	0.215 - 0.685	O_3
7.98 - 8.89	H_2O, dimer	0.685 - 0.891	H_2O
8.89 - 10.15	O_3, dimer	0.891 - 1.273	H_2O
10.15 - 11.76	H_2O, dimer	1.273 - 3.580	H_2O, CO_2
11.76 - 20.10	H_2O, CO_2		
20.10 - 100.	H_2O		

The optical parameters in a cloud-free atmosphere are taken from Hense et al. (1982) for the solar spectrum and from Eickerling (1989) for the terrestrial spectrum. In the terrestrial part scattering is neglected.

3.2.2.2 Terrestrial radiation

The optical depth δ is defined as

$$\delta = \exp\left[\sum_{j=0}^{n} a_j (\log u_i)^j\right] \tag{3.2.2.5}$$

with the effective absorber mass u_i of the ith component,

$$u_i = \rho_i \left(\frac{p}{p_0}\right)^{\alpha_p} \left(\frac{T_0}{T}\right)^{\alpha_T} d \tag{3.2.2.6}$$

where a_j, α_p, α_T are coefficients derived by inverting the results of an "exact" reference radiation model, ρ_i, p and T denote concentration of the ith component, pressure and temperature, respectively, averaged over an atmospheric layer with geometrical thickness d, p_0 and T_0 are reference values set to 1013 hPa and 273.15 ^{o} K, respectively.

With these definitions and assuming that the Planck function is a linear function of δ, the radiative transfer equations can be solved as described by Hense et al. (1982).

3.2.2.3 Solar radiation

For the transfer of solar radiation, absorption (subscript a) and multiple scattering (subscript s) are taken into account. The relevant parameters are the optical thickness δ, the single scattering albedo $\tilde{\omega}$ and the backscattering coefficients β, β_0. The backscattering coefficients β and β_0 are derived from theory (Kerschgens et al., 1978). Scattering and absorption coefficients of stratospheric, urban and maritime aerosols are taken from a data set provided by Shettle and Fenn (1975). Multiplication by d gives the appropriate optical thicknesses $\delta_a^{aerosol}$, $\delta_s^{aerosol}$.

The optical thickness δ_a^{gas} for gas absorption is a function of the effective absorber amount and is defined similarly to the terrestrial radiation (see above). The optical thickness δ_s^{gas} for Rayleigh scattering of gas molecules is given by

$$\delta_s^{gas} = \bar{N} \cdot A_R \cdot d \tag{3.2.2.7}$$

where $\bar{N}$ is the mean number density and A_R is the scattering cross-section of molecules.

For diffuse radiation, the optical thickness δ^{dif} is calculated from

$$\delta^{dif} = a_a (\delta_a^{gas} + \delta_a^{aerosol})^{b_a} + a_s (\delta_s^{gas} + \delta_s^{aerosol})^{b_s} \tag{3.2.2.8}$$

where a_a, b_a, a_s and b_s are coefficients derived by inverting results of an "exact" reference model.

The single scattering albedo is defined as

$$\tilde{\omega} = 1 - \frac{\delta_a^{dif}}{\delta_s^{dif}} \tag{3.2.2.9}$$

For direct radiation, the optical thickness δ^{dir} at a level k is calculated by adding the optical thicknesses above this level

$$\delta^{dir} = \sum_{j=1}^{k} (\delta_{a_j}^{gas} + \delta_{a_j}^{aerosol} + \delta_{s_j}^{gas} + \delta_{s_j}^{aerosol}) \tag{3.2.2.10}$$

where δ_a^{gas} depends on the zenith angle θ. δ_a^{gas} is parameterized for $\theta = 35^{o}$ and $\theta = 75^{o}$ only. Optical thicknesses for other zenith angles are calculated by linear interpolation.

3.2.2.4 Cloud optical properties

The emissivity ε of clouds in the terrestrial region is described by

$$\varepsilon = 1 - e^{-\left[\frac{k \cdot m \cdot d}{\bar{\mu}}\right]} \qquad (3.2.2.11)$$

where $k = 0.084\ m^2g^{-1}$ is the mass absorption coefficient (Stephens, 1978), d is the geometrical cloud thickness and m is the water or ice content in gm^{-3}.

Optical thickness δ, single scattering albedo $\tilde{\omega}$, and back-scattering coefficient β are the optical parameters considered in transfer of solar radiation. For both cloud phases, the parameterization of Stephens (1978) is adopted:

$$\log(\delta) = \begin{cases} 0.2633 + 1.7095 \cdot ln\,[\log(m \cdot d)] & (\lambda < 0.685\ \mu m) \\ 0.3492 + 1.6518 \cdot ln\,[\log(m \cdot d)] & \text{otherwise} \end{cases} \qquad (3.2.2.12)$$

Single scattering albedo and back-scattering coefficients are taken from Kerschgens et al. (1978).

3.2.3 Input to the radiation scheme

3.2.3.1 Temperature

Temperature values are needed at the boundaries of the layers, where the fluxes are computed. They are derived from the full level temperatures using a pressure weighted interpolation:

$$T_{k+1/2} = T_k \cdot \frac{p_k(p_{k+1} - p_{k+1/2})}{p_{k+1/2}(p_{k+1} - p_k)} + T_{k+1} \cdot \frac{p_k(p_{k+1/2} - p_k)}{p_{k+1/2}(p_{k+1} - p_k)} \tag{3.2.3.1}$$

At the bottom of the atmosphere, the ground temperature T_S is used, while at the top the model's first full level temperature is used.

3.2.3.2 Clouds

Cloud liquid water and cloud cover are provided by subroutine COND, which performs the stratiform cloud calculation. Cloud overlap is treated such that if two cloudy layers are contiguous, they are assumed to have maximum overlap of the cloudy parts. Otherwise random overlap is assumed.

3.2.3.3 Aerosols

Provision is made to have various types of aerosols:

- oceanic
- desert
- urban

3.2.3.4 Carbon dioxide and ozone

Carbon dioxide has a constant mass mixing ratio over the whole globe corresponding to a volume concentration of 345 ppm (CO2FAC = 1.045). For ECHAM1 and ECHAM2 the concentration is 330 ppm (CO2FAC = 1.0).

The ozone mixing ratio R_{03} depends on height, latitude, longitude and season. Its vertical distribution is assumed to be such that its integral from 0 to the pressure p is

$$\int_0^p R_{03}\, dp = \frac{a}{1 + (b/p)^{3/2}} \tag{3.2.3.2}$$

The constants a and b are related to the total amount of ozone and the height of its maximum mixing ratio. They are imposed in terms of a limited series of spherical harmonics for the geographical distribution and a Fourier series for the seasonal variation. The total amount of ozone was taken from London et al. (1976) and the altitude of the maximum concentration was derived from Wilcox and Belmont (1977).

3.2.3.5 Surface albedo

Over snow-free land areas an annual mean background albedo is specified from satellite data (Geleyn and Preuss, 1983). According to more recent analyses (e.g., Dorman and Sellers, 1989) the satellite-based estimates are too low in the tropics and at middle latitudes (North America, Eurasia). This bias has been removed by applying an empirical correction in those areas where the satellite data suggests values of less than 15 %. The minimum values thus obtained are around 12 % for tropical rain forests.

In snow-covered areas the surface albedo is modified according to

$$\alpha_{Surf} = \alpha_{Sb} + (\alpha_S - \alpha_{Sb}) \cdot \frac{Sn}{Sn + Sn^*} \tag{3.2.3.3}$$

where α_S is the snow albedo (see below), α_{Sb} is the background albedo, Sn is the simulated snow depth (in water equivalent) and $Sn^* = 0.01\ \mathrm{m}$ is a critical snow depth. For $Sn \gg Sn^*$ the surface albedo approaches the albedo of snow.

The albedo of snow and ice surfaces (α_S) is a function of surface type (t_S), surface temperature (T_S) and fractional forest area (a_f) over land. For $T_S \geq T_m = 273.15\ \mathrm{K}$ (i.e., for melting of snow or ice), α_S is fixed at a relatively small value, $\alpha_S = \alpha_{Smin}(t_S, a_f)$, whereas α_S is larger, $\alpha_S = \alpha_{Smax}(t_S, a_f)$, for cold surfaces ($T_S \leq T_0 = 263.15\ \mathrm{K}$) according to Robock (1980). Over land, the respective snow albedos are assumed to depend on the fractional forest area ($0 \leq a_f \leq 1$) according to

$$\begin{aligned} \alpha_{Smin}(a_f) &= a_f \cdot \alpha_{Smin}(a_f = 1) + (1 - a_f) \cdot \alpha_{Smin}(a_f = 0) \\ \alpha_{Smax}(a_f) &= a_f \cdot \alpha_{Smax}(a_f = 1) + (1 - a_f) \cdot \alpha_{Smax}(a_f = 0) \end{aligned} \tag{3.2.3.4}$$

In the temperature range $T_0 < T_S < T_m$, $\alpha_S = \alpha_S(T_S, t_S, a_f)$ is obtained by linear interpolation

$$\alpha_S = \alpha_{Smax} - (\alpha_{Smax} - \alpha_{Smin}) \cdot \frac{T_S - T_0}{T_m - T_0} \tag{3.2.3.5}$$

with α_{Smax} and α_{Smin} as specified in Table 4 according to the estimate of Robock (1980) for sea ice and snow (without the solar zenith angle correction, however) and according to Kukla and Robinson (1980) for land ice.

Over sea, the albedo is constant (0.065) for diffuse radiation but depends on the solar zenith angle θ for the direct beam

$$\alpha_{S0} = \frac{0.05}{\cos\theta + 0.15} \tag{3.2.3.6}$$

with the limitation $\alpha_{S0} \leq 0.15$.

Table 4 Minimum and maximum surface albedos (Robock, 1980; Kukla and Robinson, 1980) referring to the temperature ranges $T_S \geq T_m = 273.15$ K and $T_S \leq T_0 = 263.15$ K respectively. The fractional forest area (a_f) is taken from a data set compiled by Matthews (1983).

surface type		α_{Smin}	α_{Smax}
sea ice		0.5	0.75
land ice		0.6	0.8
snow on land	for $a_f = 0$	0.4	0.8
	for $a_f = 1$	0.3	0.4

3.2.3.6 Solar zenith angle

Equations to compute the annual variation of the solar constant I_0, the solar declination δ_s and the difference between solar time and official time can be found in Paltridge and Platt (1976). These equations are used to give the cosine of the solar zenith angle.

The mean value of the solar constant I_0 is 1367 Wm^{-2} for ECHAM1 and ECHAM2 and 1365 Wm^{-2} for ECHAM3.

3.3 VERTICAL DIFFUSION

3.3.1 Basic equations

The parameterization scheme represents the turbulent exchanges of heat, momentum, moisture and cloud water at the surface and the turbulent transports of the same quantities in the lowest levels of the model. The top of the turbulent layer is computed using a combination of convective and dynamic criteria. Above this boundary layer the scheme only operates when the air is statically unstable. The equation for the vertical diffusion of any conservative quantity X is:

$$\frac{\partial X}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}\left(\rho K_X \frac{\partial X}{\partial z}\right) = \frac{1}{\rho}\frac{\partial J_X}{\partial z} \tag{3.3.1.1}$$

K_X is the exchange coefficient and J_X (positive downwards) is the vertical turbulent flux of X. The following boundary conditions have been assumed:

$$K_X \frac{\partial X}{\partial z} = 0 \qquad \text{for} \quad p = p_T \tag{3.3.1.2.a}$$

and

$$K_X \frac{\partial X}{\partial z} \rightarrow C_X |\underset{\sim}{v}_h(z)| \, (X(z) - X_S) \quad \text{for} \quad (z \rightarrow 0) \tag{3.3.1.2.b}$$

where p_T is the pressure at the top of the boundary layer and $\underset{\sim}{v}_h = (u, v)$ is the horizontal windvector.

The definition of the drag coefficient C_X depends on the height z above the ground at which $\underset{\sim}{v}_h$ and X are taken (the natural choice is the model´s lowest level) and on the stability of the layer. X_S represents a value of X at the surface.

X may be identified with each of the five variables u, v, q_v, q_w and s (dry static energy). s is defined as:

$$s = C_{pd}(1 + (\delta - 1)\, q_v) \cdot T + g \cdot z = C_p \cdot T + \phi \tag{3.3.1.3}$$

For the latter the problem is simplifed by assuming that ϕ remains constant with respect to time during the evolution process (even if in reality T variations would modify $z(p)$). A coupling with the vertical diffusion of q_v is avoided by assuming a constant value of $C_p(z)$ in time.

Since the thickness of the model layer Δz is small near the ground, when $(K \cdot 2\Delta t) / (\Delta z)^2 > 1$, the time-stepping procedure must be implicit in order to avoid numerical instability. The whole interaction with the adiabatic and other processes, however, cannot be treated implicitly. The following procedure has been chosen:

Starting from X values at $t-\Delta t$ an implicit treatment of equations (3.3.1.1), (3.3.1.2.a) and (3.3.1.2.b) results in new values $\tilde{X}(t+1)$. Dividing the difference $\tilde{X}(t+1) - X(t-1)$ by the timestep ($2\Delta t$, or Δt at the first timestep) the result is added to the adiabatic and radiative tendencies. Thus:

$$\frac{\tilde{X}(t+1) - X(t-1)}{2\Delta t} = \frac{1}{\rho}\frac{\partial}{\partial z}\left(\rho K_{X(t-1)}\frac{\partial}{\partial z}X^{*}\right) \tag{3.3.1.4}$$

and

$$K_{X(t-1)}\frac{\partial}{\partial z}X^{*} \rightarrow C_{X(t-1)}\left|\underline{v}_h(t-1,z)\right|\left(X^{*}(z) - X_S(t-1)\right) \quad \text{for} \quad (z \rightarrow 0) \tag{3.3.1.5}$$

replace equations (3.3.1.1), (3.3.1.2.a) and (3.3.1.2.b), where

$$X^{*} = \alpha \cdot \tilde{X}(t+1) + (1-\alpha) \cdot X(t-\Delta t)$$

In the ECHAM, $\alpha = 1.5$.

X_S does not vary in time for the solution of one system. If nothing else was done, this would mean that the surface sensible and latent heat flux and the surface moisture flux might be inconsistent on land with the forecast values for the soil parameter T_S for $t+\Delta t$. This would lead to a numerical instability in T_S for a too long timestep Δt. One can prevent this instability by storing not only the fluxes but also their derivatives with respect to T_S and W_S. The surface routines can then treat the fluxes as linear functions of the soil parameters and a stable (semi-implicit) algorithm can be implemented.

The coefficients K_X and C_X are assumed to be the same for u, v, q_v, q_w and s. Using m and h as subscripts for momentum and heat, the problem is reduced to the determination of K_m and K_h (at all atmospheric layers) and C_m and C_h at the surface. This has to be done solely from atmospheric values (at $t-\Delta t$) of u, v, T, q_v, q_w and from the surface conditions.

3.3.2 Surface fluxes

The surface fluxes for momentum, dry static energy and cloud water are parameterized as follows:

$$
\begin{aligned}
J_u &= \rho \cdot C_m \cdot |\underset{\sim}{v}_h| \cdot u \\
J_v &= \rho \cdot C_m \cdot |\underset{\sim}{v}_h| \cdot v \\
J_s &= \rho \cdot C_h \cdot |\underset{\sim}{v}_h| \cdot (s - s_S) \\
J_{q_w} &= \rho \cdot C_h \cdot |\underset{\sim}{v}_h| \cdot (q_w - q_{wS})
\end{aligned}
\tag{3.3.2.1}
$$

where C_m is the drag coefficient, C_h the transfer coefficient for heat and $|\underset{\sim}{v}_h|$ the absolute value of the horizontal velocity. C_m, C_h and $|\underset{\sim}{v}_h|$ apply to the lowest model level at $t - \Delta t$. u, v and s are the velocity components and the dry static energy to be computed implicitly for the lowest model level. s_S is the dry static energy at the surface at $t - \Delta t$, and q_{wS} is the cloud water content at the surface (set to zero).

The moisture flux is evaluated distinguishing between sea and land. Over sea

$$J_{q_v} = \rho \cdot C_h \cdot |\underset{\sim}{v}_h| \cdot \{ q_v - q_s (T_S, p_S) \} \tag{3.3.2.2}$$

where q_s is the saturation specific humidity at surface temperature T_S and pressure p_S.

Over land, each grid square is divided into 4 fractions:

i. fraction C_{Sn} covered with snow
ii. fraction $(1 - C_{Sn}) \cdot C_l$ covered with water in skin reservoir
iii. fraction $(1 - C_{Sn}) \cdot (1 - C_l) \cdot (1 - C_v)$ covered with bare soil
iv. fraction $(1 - C_{Sn}) \cdot (1 - C_l) \cdot C_v$ covered with vegetation

The snow cover fraction C_{Sn} depends on the snow cover Sn :

$$C_{Sn} = min\,(1, \frac{Sn}{Sn_{cr}}) \tag{3.3.2.3}$$

where Sn_{cr} is the critical snow cover (= 0.015 m equivalent water depth).

The wet skin fraction C_l is derived from the skin reservoir water content:

$$C_l = min\left(1, \frac{W_l}{W_{lmx}}\right) \tag{3.3.2.4}$$

with

$$W_{lmx} = W_{lmax}\,[\,(1 - C_v) + C_v \cdot L_t] \tag{3.3.2.5}$$

W_l is the prognostic variable for the skin reservoir content, W_{lmx} is the maximum skin reservoir content, C_v is the vegetation ratio, L_t is the leaf area index, and W_{lmax} is the maximum amount of water that can be held on one layer of leaf or bare ground. It is taken to be $2. \cdot 10^{-4}$ m.

The grid fraction C_v occupied by vegetation is equal to the climatological field C_{vcl} except in dry conditions when the vegetation is reduced according to the following empirical expression:

$$C_v = min\left(C_{vcl}, C_{vcl} \cdot \frac{W_S}{0.4 \cdot W_{Smax}}\right) \tag{3.3.2.6}$$

The quantity W_S represents the total amount of water available in the root zone and W_{Smax} is the field capacity (0.2 m water).

Evaporation from snow and the skin reservoir is at the potential rate

$$J_{q_v}(i), (ii) = \rho \cdot C_h \cdot |\underset{\sim}{v}_h| \cdot \{q_v - q_s(T_S, p_S)\} \tag{3.3.2.7}$$

For the evaporation from bare soil (no water in skin reservoir) it is assumed that the relative humidity h at the surface is related to the water content W_S of the soil:

$$J_{q_v}(iii) = \rho \cdot C_h \cdot |\underset{\sim}{v}_h| \cdot \{q_v - h \cdot q_s(T_S, p_S)\} \tag{3.3.2.8}$$

$$h = max\left[0.5 \cdot \left(1 - \cos\frac{\pi \cdot W_S}{W_{Smax}}\right), min\left(1, \frac{q_v}{q_s(T_S, p_S)}\right)\right] \tag{3.3.2.9}$$

The evaporation from dry (no water in skin reservoir) vegetated areas is proportional to the evaporation efficiency E:

$$J_{q_v}(iv) = \rho \cdot C_h \cdot |\underset{\sim}{v}_h| \cdot E \cdot \{q_v - q_s(T_S, p_S)\} \tag{3.3.2.10}$$

Based on Sellers et al. (1986), E is expressed as:

$$E = \left[1 + \frac{C_h \cdot |\underset{\sim}{v}_h| \cdot R_{co}(PAR)}{F(W_S)}\right]^{-1} \tag{3.3.2.11}$$

where $R_{co}/F(W_S)$ is the stomatal resistance of the canopy, with a minimum value R_{co} dependent on the **P**hotosynthetically **A**ctive **R**adiation (PAR), and an empirical function of the available water in the root zone $F(W_S)$.

$$\frac{1}{R_{co}} = \frac{1}{k \cdot c}\left[\frac{b}{d \cdot PAR} \cdot ln\left(\frac{d \cdot e^{k \cdot L_t} + 1}{d + 1}\right) - ln\left(\frac{d + e^{k \cdot L_t}}{d + 1}\right)\right] \tag{3.3.2.12}$$

where

$$d = \frac{a + b \cdot c}{c \cdot PAR}, \; k = 0.9, \; L_t = 4, \; a = 5000 \; Jm^{-3}, \; b = 10 \; Wm^{-2}, \; c = 100 \; sm^{-1}$$

and PAR is 55 % of the net short wave radiation at the surface. In case of dew deposition $(q_v > q_s)$ we set

$$E = h = 1 \tag{3.3.2.13}$$

The water stress factor $F(W_S)$ is

$$F(W_S) = \begin{cases} 1 & \text{if} \quad W_S \geq W_{cr} \\ \dfrac{W_S - W_{pwp}}{W_{cr} - W_{pwp}} & \text{if} \quad W_{pwp} < W_S < W_{cr} \\ 0 & \text{if} \quad W_S \leq W_{pwp} \end{cases} \tag{3.3.2.14}$$

W_{cr} is a critical value taken as 50% of the field capacity W_{Smax}, while W_{pwp} is the permanent wilting point taken as 20% of W_{Smax}.

The total evaporation in a grid square combines the four fractions:

$$\begin{aligned} J_{q_v} = \rho \cdot C_h \cdot |\underset{\sim}{v}_h| \cdot [\, & \{C_{Sn} + (1 - C_{Sn}) \cdot C_l\} \cdot \{q_v - q_s\} \\ & + (1 - C_{Sn}) \cdot (1 - C_l) \cdot (1 - C_v) \cdot \{q_v - h \cdot q_s\} \\ & + (1 - C_{Sn}) \cdot (1 - C_l) \cdot C_v \cdot E \cdot \{q_v - q_s\} \,] \end{aligned} \tag{3.3.2.15}$$

3.3.3 Definition of the drag coefficients

The method has been described by Louis (1979) and updated subsequently in Louis et al. (1982). Only the main points are mentioned here.

Starting from the Monin-Obukhov similarity theory (using dry air for simplicity), the gradients of wind (u) and internal energy ($s = C_p \cdot T + g \cdot z$) are assumed to be universal functions of a stability parameter.

$$\frac{k\ z}{u_*} \cdot \frac{\partial u}{\partial z} = \phi_m(z/L) \qquad (3.3.3.1.a)$$

$$\frac{k\ z}{s_*} \cdot \frac{\partial s}{\partial z} = \phi_h(z/L) \qquad (3.3.3.1.b)$$

The stability parameter L is the Obukhov length:

$$L = \frac{s \cdot u_*^2}{k \cdot g \cdot s_*} \qquad (3.3.3.2.a)$$

where k is von Karman's constant and u_* and s_* are scaling parameters derived from the fluxes:

$$\rho \cdot u_*^2 = (J_u)_{z \to 0} = \rho \cdot C_m \cdot |u|^2 \qquad (3.3.3.2.b)$$

$$\rho \cdot u_* \cdot s_* = (J_s)_{z \to 0} = \rho \cdot C_h \cdot |u| \cdot (s - s_S) \qquad (3.3.3.2.c)$$

Equations (3.3.3.1.a) and (3.3.3.1.b) can be integrated over the lowest model layer, and L eliminated using (3.3.3.2.a) in order to derive C_m and C_h. However, such expression cannot be obtained analytically because of the complicated form of ϕ_m and ϕ_h. C_m and C_h are approximated by following analytical expressions (Louis, 1979):

$$C_m = \left(\frac{k}{ln\ (z/z_0)}\right)^2 \cdot f_m(Ri, z/z_0) \qquad (3.3.3.3.a)$$

and

$$C_h = \left(\frac{k}{ln\ (z/z_0)}\right)^2 \cdot f_h(Ri, z/z_0) \qquad (3.3.3.3.b)$$

here the Richardson number Ri is defined as:

$$Ri = \frac{g \cdot \Delta z \cdot \Delta s}{C_p \cdot T \cdot |\Delta u|^2} \qquad (3.3.3.4)$$

The empirical functions f_m and f_h must have the correct behaviour near neutrality and in the asymptotic cases of high stability or instability.

a) Near neutrality one obtains $Ri \to 0, z/L \to 0, \phi_m = 1 + b_m \cdot z/L$ and $\phi_h = 1 + b_h \cdot z/L$.

We obtain then

$$\begin{aligned} f_m &\cong 1 - 2 \cdot b_m \cdot Ri \\ f_h &\cong 1 - (b_m + b_h) \cdot Ri \end{aligned} \tag{3.3.3.5}$$

Furthermore , there is some evidence (Pandolfo ,1967), that $Ri \sim z/L$ which implies $b_h = 2 \cdot b_m$. Thus we have

$$f_m = 1 - 2 \cdot b \cdot Ri \quad \text{and} \quad f_h = 1 - 3 \cdot b \cdot Ri \tag{3.3.3.6}$$

In ECHAM3 b = 5.

b) In highly unstable cases only the problem of f_h in the case of vanishing u (free convection case) has been considered. To have a non-zero heat flux, u must vanish from $(u \cdot f_h)$ for large negative $(s - s_S)$ (that is for high negative value of Ri). So f_h must behave like $\sqrt{-Ri}$.

The analytical expression chosen for the whole unstable case $(Ri < 0)$ is:

$$f_h = 1 - \frac{3 \cdot b \cdot Ri}{1 + C_h(z/z_0) \cdot \sqrt{-Ri}} \tag{3.3.3.7}$$

For this free convection case one can write an equation for a new similarity theory, independent of J_u,

$$\frac{z}{s_*} \cdot \frac{\partial s}{\partial z} = C \tag{3.3.3.8}$$

with

$$s_* = \left[\frac{(J_s/\rho)^2}{(g \cdot z)/s} \right]^{1/3}$$

Using Eq. (3.3.3.7) for the limit $Ri \to \infty$,

$$C_h(z/z_0) = C \cdot 3 \cdot b \left[\frac{k}{ln(z/z_0)} \right]^2 [(z/z_0)^{1/3} - 1]^{3/2} \tag{3.3.3.9}$$

One can furthermore replace $[(z/z_0)^{1/3} - 1]^{3/2}$ by $\sqrt{(z/z_0)}$ if z is much greater than z_0, so that

$$C_h(z/z_0) = C \cdot 3 \cdot b\left[\frac{k}{ln\,(z/z_0)}\right]^2 \sqrt{(z/z_0)} \tag{3.3.3.10}$$

In order to avoid numerical problems for high z_0 values that might suppress z one replaces (z/z_0) by $(z/z_0) + 1$ in (3.3.3.10) and (3.3.3.3.a), (3.3.3.3.b).

In ECHAM3 $C = 5$.

In the highly unstable case f_m is not very important since there is little wind shear on which to act. Therefore a similar expression to (3.3.3.7) has been chosen with the same denominator to save computing time

$$f_m = 1 - \frac{2 \cdot b \cdot Ri}{1 + 3 \cdot b \cdot C\left[\frac{k}{ln\,(z/z_0 + 1)}\right]^2 \sqrt{(z/z_0 + 1)\,(-Ri)}} \tag{3.3.3.11}$$

c) Finally in the highly stable case we follow Ellison (1957) by combining two equations

$$\phi_m^4 - \frac{z}{L}\phi_m^3 - 1 = 0 \qquad \text{KEYPS equation} \tag{3.3.3.12}$$

and

$$\frac{\phi_m}{\phi_h} = \frac{1 - (\frac{z}{L})/\phi_m/R_{crit}}{(1 - \left[(\frac{z}{L})/\phi_m\right])^2} \tag{3.3.3.13}$$

The flux Richardson number $Ri_f = \frac{z}{L}/\phi_m$ has its critical value R_{crit} for $(Ri \to \infty)$.

For $\frac{z}{L} \to +\infty$ this gives

$$f_m \sim 1/(\sqrt{\mathrm{Ri}}) \text{ and } f_h \sim 1/(Ri^{3/2}) \tag{3.3.3.14}$$

For lack of better information and to minimise the computing time for the whole stable range $(Ri > 0)$ the following expressions have been chosen

$$f_m = \frac{1}{1 + (2 \cdot b \cdot Ri)/(\sqrt{1 + d \cdot Ri})} \tag{3.3.3.15}$$

$$f_h = \frac{1}{1 + (3 \cdot b \cdot Ri) \cdot (\sqrt{1 + d \cdot Ri})} \tag{3.3.3.16}$$

d is related to the critical flux Richardson number R_{crit} by

$$R_{crit} = \frac{2}{3 \cdot d} \tag{3.3.3.17}$$

In ECHAM3 $d = 5$.

A plot of the drag coefficients for heat and momentum, normalized by their value at neutrality, is shown in Figure 8.

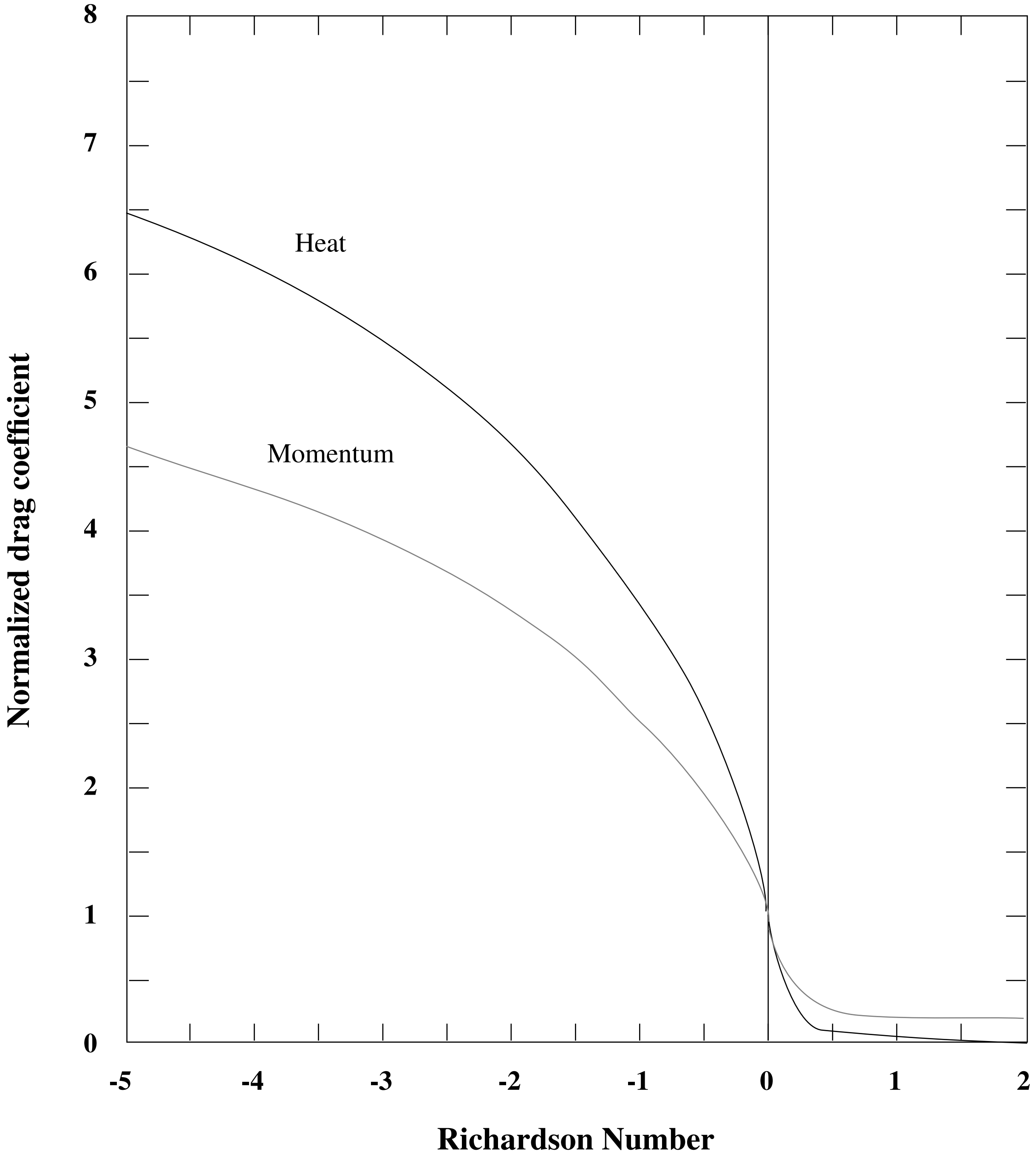


Figure 8 Drag coefficients for heat (solid) and momentum (dashed) in terms of the Richardson number, normalized by their value at neutrality.

3.3.4 Definition of the exchange coefficients

The logical extension of the similarity theory for surface fluxes to the atmosphere is the mixing length approach.

$$K_m = l_m^2 \cdot \left|\frac{\partial \underset{\sim}{v}_h}{\partial z}\right| \cdot f_m(Ri) \tag{3.3.4.1}$$

$$K_h = l_h^2 \cdot \left|\frac{\partial \underset{\sim}{v}_h}{\partial z}\right| \cdot f_h(Ri) \tag{3.3.4.2}$$

The functional dependencies of f_m and f_h on Ri are the same as for C_m and C_h except that the term

$$\left[\frac{k}{ln(z/z_0)}\right]^2 [(z/z_0)^{1/3} - 1]^{3/2}$$

in (3.3.3.9) is replaced by its equivalent

$$\frac{l^2}{(\Delta z)^{3/2} \cdot z^{1/2}} \left[\left(\frac{z+\Delta z}{z}\right)^{1/3} - 1 \right]^{3/2} \tag{3.3.4.3}$$

The intensity of the vertical diffusion in the atmosphere is dependent on the choice of l_h and l_m. Here, the solution of Blackadar (1962) is used, that goes smoothly from the asymptotic value $k \cdot z$ for $z \rightarrow 0$ to a constant value in the high atmosphere:

$$\frac{1}{l_m} = \frac{1}{k \cdot z} + \frac{1}{\lambda_m} \quad , \quad \frac{1}{l_h} = \frac{1}{k \cdot z} + \frac{1}{\lambda_h} \tag{3.3.4.4}$$

One also has to have a relationship between λ_m, λ_h and d so that the critical flux Richardson number in the atmosphere is 1 (e.g. a balance between shear generation and buoyancy destruction).

$$R_{crit}(z \rightarrow \infty) = \frac{2}{3d} \cdot \left(\frac{\lambda_h}{\lambda_m}\right) = 1 \tag{3.3.4.5}$$

gives $\lambda_h = \lambda_m \sqrt{(3 \cdot d)/2}$.

In ECHAM3 $\lambda_m = 160 \text{ m}$.

3.3.5 Moisture and cloud effects

In the ECMWF model and also in ECHAM1 and ECHAM2, moisture effects on stability have been included by replacing the temperature T in the definition in the Richardson number (3.3.3.4) by the virtual temperature T_v since it is T_v that determines the buoyancy in moist (but cloud-free) air. In ECHAM3 cloud effects are considered by reformulating the Richardson number to include the impact of cloud processes on the buoyancy term (Brinkop, 1992). This so-called moist Richardson number can be written according to

$$Ri_m = \frac{g}{\Theta_v} \cdot \frac{A \cdot \Delta\Theta_l + \Theta \cdot B \cdot \Delta q_t}{(\Delta u)^2 + (\Delta v)^2} \cdot \Delta z \tag{3.3.5.1}$$

where $q_t = q_l + q_i + q_v$ is the total water content, Θ_v is the virtual potential temperature, $\Theta_l = \Theta - (\Theta \cdot L \cdot q_l / T \cdot C_{pd})$ is the liquid water potential temperature which is a conservative quantity in the absence of freezing, precipitation and radiative effects (Betts, 1973), and Θ is the potential temperature.

The parameters A and B are defined as follows:

Unsaturated case

$$A_{unsat} = 1 + 0.61 \cdot q_t \tag{3.3.5.2}$$

$$B_{unsat} = 0.61 \tag{3.3.5.3}$$

Saturated case

$$A_{sat} = 1 + 0.61 \cdot q_t - \frac{0.622 \cdot \dfrac{L}{R_d \cdot T} \cdot q_s}{1 + 0.622 \cdot \dfrac{L^2}{R_d \cdot C_{pd} \cdot T^2}} \cdot \left(\frac{L}{C_{pd} \cdot T} (1 + 0.61 \cdot q_t) \right) \tag{3.3.5.4}$$

$$B_{sat} = \frac{L}{C_{pd} \cdot T} - 1 \tag{3.3.5.5}$$

The coefficients A and B in (3.3.5.1) are assumed to depend linearly on cloud cover b:

$$A = b \cdot A_{sat} + (1 - b) \cdot A_{unsat} \tag{3.3.5.6}$$

$$B = b \cdot B_{sat} + (1 - b) \cdot B_{unsat} \tag{3.3.5.7}$$

In the clear-sky case ($b = 0$ and $q_t = q_v$) Ri_m as defined by (3.3.5.1) is identical to the more familiar form used in the ECMWF model, for example.

3.3.6 Definition of the top of the boundary layer

Two levels are computed.

a) First level above the dynamical height (Ekman layer height).

$$h_{dyn} = 0.5 \cdot (u_* / f)$$

Equatorwards of 20 ° f is set to $5 \cdot 10^{-5} s^{-1}$.

b) A dry convective level, h_{cnv}, is defined as the lowest level for which the static stability $s > s_{NLEV}$.

The top of the planetary boundary layer is then given by

$$h_{pbl} = max\,(h_{dyn}, h_{cnv}) \qquad (3.3.6.1)$$

The above formulation takes into account early morning cases (where the dry convective boundary layer starts to develop), where considering $h_{cnv} = h_{dyn}$ would give too strong a vertical constraint for the turbulent diffusion. p_T is then calculated as the pressure of the first model level above h_{pbl}.

3.3.7 Roughness length

Over land the roughness length z_0 is geographically prescribed. Over ice-free sea z_0 is calculated from the Charnock (1955) formula:

$$z_0 = C_{char} \cdot \frac{u_*^2}{g} \qquad (3.3.7.1)$$

with a minimum value for z_0 of $1.5 \cdot 10^{-5} m$. In ECHAM3 the Charnock constant C_{char} is set to 0.018 for T21 resolution and to 0.032 for higher resolutions. Over sea ice a constant value of $z_0 = 0.001$ m is used.

In unstable conditions over sea an empirical interpolation for heat and moisture is used between the free convection limit and the neutral approximation (Miller et al., 1992):

$$C_h = C_{mn} \cdot (1 + C_R^{\gamma})^{1/\gamma} \qquad (3.3.7.2)$$

where

$$C_R = \frac{0.0016 \cdot (\Delta\Theta_v)^{1/3}}{C_{mn} \cdot |\underset{\sim}{v}_h|}$$

$$C_{mn} = \left(\frac{k}{ln\,(z/z_0)}\right)^2$$

and $\gamma = 1.25$.

In stable situations the same formulation is used as over land, with z_0 according to (3.3.7.1).

3.3.8 Vegetation

Over land, a vegetation ratio C_v is assigned to each grid box.

3.3.9 Kinetic energy dissipation

The kinetic energy lost by the mean flow through the diffusion process is assumed to be directly transformed into internal energy:

$$\delta T_{\text{dissipation}} = \frac{1}{C_p(q)} \cdot \varepsilon = \frac{1}{C_p(q)} \cdot \frac{u^2 + v^2 - (\tilde{u}^2 + \tilde{v}^2)}{2} \qquad (3.3.9.1)$$

This procedure by-passes the sub-grid scale energy cascade, but it allows a closed energy cycle in the model.

3.3.10 Vertical diffusion at higher levels

Above the planetary boundary layer (cf., section 3.3.6) the vertical diffusion scheme only operates when the air is statically unstable.

3.4 GRAVITY WAVE DRAG

3.4.1 Theory

The parameterisation scheme represents the momentum transports due to sub-grid scale gravity waves excited by stably stratified flow over irregular terrain (Miller et al., 1989). The scheme is a modified form of that proposed by Boer et al. (1984) and by Palmer et al. (1986) in which a low-level wave stress is defined together with criteria for the reduction in stress with height as the vertically propagating waves are absorbed and/or reflected.

The influence of these wave stresses in regions of wave momentum flux divergence is as follows:

$$\left(\frac{\partial \underset{\sim}{v}_h}{\partial t}\right)_{\text{gravity waves}} = -g\left(\frac{\partial \underset{\sim}{\tau}}{\partial p}\right)_{\text{gravity waves}}$$

where $\underset{\sim}{v}_h$ is the horizontal wind vector, and $\underset{\sim}{\tau}$ is the wave stress.

3.4.2 The formulation of the scheme

The formulation of the scheme consists of two parts

(i) The parametric form for $\underset{\sim}{\tau}(x, y, t)$

(ii) The modelling of the dynamical processes which determine the vertical distribution.

(i) The following quantities are defined for use in the scheme:

$\underset{\sim}{V}_L$ is the wind vector for a suitably defined low-level flow and ρ_L, N_L are the low-level density and the Brunt-Väisälä frequency respectively. The function $f_1(\underset{\sim}{x})$ describes the orographic forcing of gravity waves and in the scheme described here $f_1(\underset{\sim}{x})$ is prescribed to be a directionally dependent sub-grid scale orographic variance computed from the US Navy dataset containing mean orographic heights for ($1/6^{o}$ lat * $1/6^{o}$ lon) areas. Fr is a form of Froude number for the low-level flow defined as

$$Fr = \frac{N_L\sqrt{f_1(\underset{\sim}{x})}}{|\underset{\sim}{V}_L|}$$

where $\underset{\sim}{V}_L$ represents a low-level wind vector averaged over the lowest three levels of the model.

$f_2(\underset{\sim}{x})$ is an orographic anisotropy function measuring the two-dimensionality of the sub-grid scale orography

$$f_2(\underset{\sim}{x}) = 1 - e^{(1-\alpha)}\text{, and } \alpha = max(\alpha_1, \alpha_2) \qquad (3.4.2.1)$$

where α_1 and α_2 are computed from the sub-grid scale variances of the four directional components (N/S, E/W, NE/SW, NW/SE)

$$\begin{aligned}\alpha_1^2 &= max\left(\frac{var(N/S)}{var(E/W)}, \frac{var(E/W)}{var(N/S)}\right)\\ \alpha_2^2 &= max\left(\frac{var(NE/SW)}{var(NW/SE)}, \frac{var(NW/SE)}{var(NE/SW)}\right)\end{aligned} \tag{3.4.2.2}$$

z_c is the atmospheric depth corresponding to three-quarters of a hydrostatic vertical gravity wavelength and is computed by the solution of the equation

$$\int_0^{z_c} \frac{N(z)}{U(z)} dz = \frac{3\pi}{2} \tag{3.4.2.3}$$

The scheme can then be written as $\underset{\sim}{\tau}_{gw}(p) = \underset{\sim}{\tau}_w(p) + \underset{\sim}{\tau}_{Fr}(p)$.

The first term $\underset{\sim}{\tau}_w(p)$ describes a part of the low-level drag and the upper part, dependent on the pressure p

$$\underset{\sim}{\tau}_w(p) = \begin{cases} \underset{\sim}{\tau}_w(p_S) \cdot (1-\beta) \cdot \dfrac{p-p'}{p_S-p'} & \text{for} \quad (p \ge p') \\ \beta \cdot \underset{\sim}{\tau}_w(p_S) \cdot f(p) & \text{for} \quad (p < p') \end{cases} \tag{3.4.2.4}$$

where $f(p)$ describes the vertical stress profile, computed as shown in (ii) below.

The choice of p' determines the depth for this part of the low-level drag and is currently chosen as $p' = 0.8 \cdot p_S$, and β controls the ratio of low to high-level drag (currently equal to 0.3).

$\underset{\sim}{\tau}_w(p_S)$ is determined by

$$\underset{\sim}{\tau}_w(p_S) = K \cdot \underset{\sim}{V}_L \cdot N_L \cdot var^* \quad \text{, where} \quad var^* = min\left(var, \left(Fr_c \cdot |\underset{\sim}{V}_L| / N_L\right)^2\right)$$

and Fr_c is the critical value for the low-level Froude number (currently equal to 2).

Analytical results of an isolated bell-shaped mountain give a value for $K = \pi / (16 \cdot a)$ where a is the mountain half-width. Hence $K \sim 2.5 \cdot 10^{-5} m^{-1}$ for typical sub-grid scale orography.

The second term $\underset{\sim}{\tau}_{Fr}(p)$ describes the additional drag which occurs when the low-level flow is supercritical and the dynamical mechanism of resonant trapping of waves occurs leading to high-drag situations (see, for example, Peltier and Clark, 1986). It takes the form

$$\underset{\sim}{\tau}_{Fr}(p) = \underset{\sim}{\tau}_{Fr}(p_S) \cdot \frac{p - p_{z_c}}{p_S - p_{z_c}} \quad \text{if} \quad (Fr > Fr_c) \tag{3.4.2.5}$$

where

$$\underset{\sim}{\tau}_{Fr}(p_S) = \begin{cases} K_L \cdot \rho \cdot \dfrac{\underset{\sim}{V}_L^3}{N_L} \cdot (Fr - Fr_c)^2 \cdot f_2(\underset{\sim}{x}) & \text{if} \quad (Fr > Fr_c) \\ 0 & \text{otherwise} \end{cases}$$

p_{z_c} is the pressure corresponding to the height z_c and K_L is currently 4 K. Typical values of z_c are around 3 to 5 km but much larger values do occur.

(ii) The vertical structure of $\underset{\sim}{\tau}_{gw}$ is calculated by constructing a local wave Richardson number which attempts to describe the onset of turbulence due to the gravity waves becoming convectively unstable or encountering critical layers.

This wave Richardson number can be written in the form $\tilde{R} = \bar{R} \cdot (1 - \alpha) / (1 + \alpha \cdot \bar{R}^{1/2})^2$ where $\bar{R}$ is the Richardson number of the basic flow. The parameter $\alpha = N \cdot |\delta z| / u$ is a form of inverse Froude number in which $|\delta z|$ represents the amplitude of the wave and u is the wind speed resolved in the direction of $\underset{\sim}{\tau}_{gw}$.

By requiring that $\tilde{R}$ never falls below a critical value $\tilde{R}_{crit}$ (currently equal to 0.25), values of wave stress are defined progressively from the surface upwards.

3.5 CUMULUS CONVECTION

Cumulus convection is parameterized by a mass flux scheme which is described in detail (including numerical aspects) in Tiedtke (1989). The scheme considers penetrative convection, shallow convection and midlevel convection. Clouds are represented by a bulk model and include updraft and downdraft mass fluxes. Momentum transport by convective circulations is also included following the proposal by Schneider and Lindzen (1976).

3.5.1 Large-scale budget equations

The contributions from cumulus convection to the large-scale budget equations of heat, moisture and momentum are:

$$
\begin{aligned}
\left(\frac{\partial \bar{s}}{\partial t}\right)_{cu} &= -\frac{1}{\bar{\rho}}\frac{\partial}{\partial z}[M_u s_u + M_d s_d - (M_u + M_d)\,\bar{s}] \\
&\quad + L\,(c_u - e_d - \tilde{e}_l - \tilde{e}_p) - (L_s - L_v)\,m \\
\left(\frac{\partial \bar{q}_v}{\partial t}\right)_{cu} &= -\frac{1}{\bar{\rho}}\frac{\partial}{\partial z}[M_u q_u + M_d q_d - (M_u + M_d)\,\bar{q}_v] \\
&\quad + (c_u - e_d - \tilde{e}_l - \tilde{e}_p) \\
\left(\frac{\partial \bar{u}}{\partial t}\right)_{cu} &= -\frac{1}{\bar{\rho}}\frac{\partial}{\partial z}[M_u u_u + M_d u_d - (M_u + M_d)\,\bar{u}] \\
\left(\frac{\partial \bar{v}}{\partial t}\right)_{cu} &= -\frac{1}{\bar{\rho}}\frac{\partial}{\partial z}[M_u v_u + M_d v_d - (M_u + M_d)\,\bar{v}]
\end{aligned}
\tag{3.5.1.1}
$$

where M_u, M_d, c_u and e_d are the net contributions from all clouds to the upward mass flux, downward mass flux, condensation/sublimation and evaporation respectively, and s_u, s_d, q_u, q_d, u_u, u_d, v_u and v_d are the weighted averages of s, q_v, u and v from all updrafts and downdrafts. Here $\tilde{e}_l$ is the evaporation of cloud water that has been detrained into the environment, $\tilde{e}_p$ is the evaporation of precipitation in the unsaturated subcloud layer and m is the melting of snow. In addition to (3.5.1.1) we consider the equations for precipitation

$$
\begin{aligned}
P^w(z) &= \int (G_P^w - e_d^w - \tilde{e}_P^w + m)\,\bar{\rho}\,dz \\
P^l(z) &= \int (G_P^l - e_d^l - \tilde{e}_P^l + m)\,\bar{\rho}\,dz
\end{aligned}
\tag{3.5.1.2}
$$

where $P^w(z)$, $P^l(z)$ are the fluxes of rain water and snow at height z and G_P^w and G_P^l are the conversion rates from cloud ice and cloud water into precipitation, respectively.

3.5.2 Cloud model equations

The updraft of the cloud ensemble is assumed to be in a steady state. Then the bulk equations for mass, heat, moisture, cloud water content and momentum are

$$\begin{aligned}
\frac{\partial M_u}{\partial z} &= E_u - D_u \\
\frac{\partial}{\partial z}(M_u s_u) &= E_u \bar{s} - D_u s_u + L\bar{\rho} c_u \\
\frac{\partial}{\partial z}(M_u q_u) &= E_u \bar{q}_v - D_u q_u - \bar{\rho} c_u \\
\frac{\partial}{\partial z}(M_u l) &= D_u l + \bar{\rho} c_u - \bar{\rho} G_P \\
\frac{\partial}{\partial z}(M_u u_u) &= E_u \bar{u} - D_u u_u \\
\frac{\partial}{\partial z}(M_u v_u) &= E_u \bar{v} - D_u v_u
\end{aligned} \tag{3.5.2.1}$$

where E and D are the rates of mass entrainment and detrainment per unit length, l is the cloud water content (water/ice) and c_u is the net condensation/sublimation in the updrafts.

Cloud air is assumed to be saturated and cloud processes are crudely represented. Freezing and melting processes are not considered and the conversion from cloud droplets to rain/snow is assumed to be proportional to the cloud water content as

$$\bar{\rho} G_P = K(z) \cdot l \cdot M_u$$

where $K(z)$ is an empirical function that varies with height. This simple parameterization yields rather reasonable vertical distributions of the generation of raindrops (Yanai et al., 1973). Here K is assumed to be zero near cloud base and constant at higher levels

$$K(z) = \begin{cases} 0, & \text{if } (z \le Z_B + \Delta z) \\ 2 \cdot 10^{-3} m^{-1}, & \text{if } (z > Z_B + \Delta z) \end{cases} \tag{3.5.2.2}$$

where Δz is 1500 m over sea and 3000 m over land. The choice of $K = 0$ at lower levels ensures that shallow cumuli do not produce precipitation, noting that a sizeable portion of the liquid water content in nonprecipitating cumuli is of precipitation-sized drops.

We further note that the cloud water detrained into the environmental air is assumed to evaporate there instantaneously. Then

$$\bar{e}_l = \frac{1}{\bar{\rho}} \cdot D_u \cdot l \tag{3.5.2.3}$$

The vertical integration of (3.5.2.1) requires the knowledge of cloud base mass flux and mass entrainment and detrainment. Cloud-base mass flux is determined for the various types of convection from the parameterization assumptions discussed below. Entrainment of mass into convective plumes is assumed to occur through turbulent exchange of mass through the cloud edges and through organized inflow associated with large-scale convergence, detrainment through turbulent exchange and as organized outflow at the top:

$$E_u = E_u^{(1)} + E_u^{(2)} , \qquad D_u = D_u^{(1)} + D_u^{(2)} \tag{3.5.2.4}$$

Turbulent entrainment and detrainment are parameterized as

$$E_u^{(1)} = \varepsilon_u M_u , \qquad D_u^{(1)} = \delta_u M_u \tag{3.5.2.5}$$

where the fractional entrainment/detrainment rates depend inversely on cloud radii (Simpson and Wiggert, 1969; Simpson, 1971):

$$\varepsilon_u = \frac{0.2}{R_u} , \qquad \delta_u = \frac{0.2}{R_u} \tag{3.5.2.6}$$

By assuming typical cloud sizes for the various types of convection, average values of entrainment/detrainment rates are defined. In the presence of synoptic scale flow convergence, large clouds which contribute most to the convective heating and moistening are assumed to exist and consequently small values for entrainment/detrainment rates are imposed whereas in the absence of flow convergence clouds of smaller sizes with larger entrainment rates prevail. In order to keep the scheme simple we use fixed values of turbulent entrainment/detrainment rates for each of the various types of convection:

$$\varepsilon_u = \delta_u = \begin{cases} 1 \times 10^{-4} m^{-1} & \text{for penetrative and midlevel convection in the presence of large-scale flow convergence} \\ 3 \times 10^{-4} m^{-1} & \text{for shallow convection in suppressed conditions} \end{cases} \tag{3.5.2.7}$$

For penetrative convection and midlevel convection we deliberately impose a very small value typical for tropical deep convective clouds (Simpson, 1971) so as not to inhibit the penetration of clouds to large heights. For shallow convection we use a value typical for the larger trade wind cumuli (Nitta, 1975), noting that small clouds with much larger entrainment/detrainment rates which detrain immediately above cloud base are not represented in our parameterization.

The parameterization of organized entrainment and detrainment is discussed below.

Below cloud base, the net convective fluxes of heat and moisture due to updraft and compensating subsidence in the environment are assumed to decrease linearly from their values at cloud base towards zero at the surface so as not to alter the vertical structure of the subcloud layer and in particular its well-mixed character.

Downdrafts are considered to be associated with convective precipitation from the updrafts and originate from cloud air influenced by the injection of environmental air. Following Fritsch and Chappell (1980) and Foster (1958), the Level of Free Sinking (LFS) is assumed to be the highest model level where a mixture of equal parts of cloud air and saturated environmental air at wet-bulb temperature becomes negatively buoyant with respect to the environmental air. The downward mass flux is assumed to be directly proportional to the upward mass flux. Following Johnson (1976, 1980) the mass flux at the LFS is specified from the updraft mass flux at cloud base as

$$(M_d)_{LFS} = \gamma (M_u)_{base} \quad \text{with} \quad \gamma = -0.3 \tag{3.5.2.8}$$

The coefficient γ is a disposable parameter.

The vertical distribution of the downdraft mass flux, dry static energy and moisture below the LFS is determined by the equations for mass, dry static energy and moisture content as

$$\begin{aligned}
-\frac{\partial M_d}{\partial z} &= E_d - D_d \\
-\frac{\partial}{\partial z}(M_d s_d) &= E_d \bar{s} - D_d s_d + L\bar{\rho} e_d \\
-\frac{\partial}{\partial z}(M_d q_d) &= E_d \bar{q}_v - D_d q_d + \bar{\rho} e_d \\
-\frac{\partial}{\partial z}(M_d u_d) &= E_d \bar{u} - D_d u_d \\
-\frac{\partial}{\partial z}(M_d v_d) &= E_d \bar{v} - D_d v_d
\end{aligned} \tag{3.5.2.9}$$

We note that cumulus downdrafts can be viewed as the reverse of updrafts. Entrainment and detrainment in downdrafts are highly uncertain as relevant data are not available. Numerical experiments show, however, that the results are rather insensitive to changes in the entrainment and detrainment rates. We use

$$\varepsilon_d = \delta_d = 2 \times 10^{-4} m^{-1} \tag{3.5.2.10}$$

This gives a mass flux which is independent of height and which effectively detrains into the environment in the subcloud layer. We also note that e_d is the evaporation of convective rain to maintain a saturated descent and that the moistening and cooling of the environmental air injected at LFS is also due to evaporating rain. As the downdrafts are determined from the updrafts the remaining parameterization task is to specify the updraft. This is done by means of closures defined below for the various types of convection.

a. Penetrative convection

Many diagnostic studies show that penetrative convection predominantly occurs in disturbed situations and strongly depends on low-level synoptic scale convergence. Various parameterization schemes are based on this relationship one way or another. Here, we apply a moisture convergence hypothesis: Following Kuo (1965, 1974) and Lindzen (1981), we postulate that when there is a deep layer of conditional instability and large-scale moisture convergence, cumulus clouds exist that entrain environmental air through their base and through their sides directly proportional to the supply of moisture and detrain cloud air at higher levels.

The injection of mass into the clouds through their base is determined by imposing a moisture balance for the subcloud layer such that the moisture content is maintained in the presence of large-scale transports, turbulent transports and convective transports. This balance may be written as

$$[M_u(q_u-\bar{q}_v)+M_d(q_d-\bar{q}_v)]_B = \int_o^B \left[\overline{\underset{\sim}{v}_h}\cdot\nabla\bar{q}_v+\bar{w}\frac{\partial\bar{q}_v}{\partial z}+\frac{1}{\bar{\rho}}\frac{\partial(\bar{\rho}\overline{w'q'_v})_{tu}}{\partial z}\right]\bar{\rho}dz \qquad (3.5.2.11)$$

where B denotes the cloud base height defined as the condensation level for surface air. The vertical distribution of the updraft mass flux above cloud base is determined using similar arguments as for the subcloud layer, that is, we postulate that there is organized entrainment which is directly proportional to the large-scale moisture convergence as

$$E_u^{(2)} = -\frac{\bar{\rho}}{\bar{q}_v}\left[\overline{\underset{\sim}{v}_h}\cdot\nabla\bar{q}_v+\bar{w}\frac{\partial\bar{q}_v}{\partial z}\right] \qquad (3.5.2.12)$$

Organized entrainment is only considered in the lower part of the cloud layer where large-scale convergence is encountered, that is, below the level of strongest vertical ascent. The idea to link the cloud mass flux directly to the large-scale moisture convergence has first been advocated by Lindzen (1981) who indicated that it may provide vertical profiles of mass flux and convective heating in good agreement with observations. The assumption (3.5.2.12) ensures that the vertical distribution of the convective mass flux follows that of the large-scale ascent which is partly supported by diagnostic studies for tropical convection (e.g. Cheng et al., 1980; Johnson, 1980). Equation (3.5.2.12) forms, together with the assumption (3.5.2.11) for the cloud base mass flux, the basic closure and as such is crucial for the performance of the parameterization of penetrative convection. The verification of the scheme during long periods of tropical convection confirms that the closure provides realistic profiles of convective mass fluxes and convective heating (see Tiedtke, 1989).

In addition to organized entrainment we consider turbulent entrainment and detrainment by equations (3.5.2.5) and (3.5.2.6).

Cumulus clouds detrain effectively at levels near to their zero-buoyancy level by means of organized outflow. Therefore, the vertical distribution of the total detrainment from all clouds depends on the spectral cloud distribution. Since spectral cloud distribution is not available, however, organized outflow is assumed to occur only in the model layer which contains the zero-buoyancy level of the deepest clouds. Our detrainment assumption implies an unimodal cloud distribution with large detrainment from the deepest clouds and little detrainment from shallow clouds and medium deep clouds.

The effect from shallow cumuli in the presence of penetrative convection has been neglected because their parameterization is still an unsolved problem. This is because the role of shallow cumuli in connection with penetrative convection is not well understood, particularly when cumulus downdraft occur simultaneously as these compete with shallow convection having similar effects on the environment as shallow cumuli (Johnson, 1976). The results obtained with this scheme indicate, however, that neglecting the contributions from shallow cumuli when penetrative convection is encountered does not introduce large errors in the convective heating and drying.

b. Shallow convection

Here we consider cumulus convection, which predominantly occurs in undisturbed flow, that is in the absence of large-scale convergent flow. Typical examples are trade wind cumuli under a subsidence inversion, convection occurring in the ridge region of tropical easterly waves and daytime convection over land. This type of convection seems to be effectively controlled by subcloud layer turbulence. In fact, most of the diagnostic studies carried out for tradewind cumuli show that the net upward moisture flux at cloud base level is nearly equal to the turbulent moisture flux at the surface (Le Mone and Pennell, 1976). As this implies a quasi-steady moisture balance, we shall apply the same moisture budget equation (3.5.2.11) as for penetrative convection. The difference, however, is that the moisture supply to cumulus clouds is now largely through surface evaporation as the contributions from large-scale convergence are either small or even negative, such as in the undisturbed trades where dry air is transported downward to lower levels.

Under typical tradewind conditions the vertical distribution of the total convective fluxes above cloud base is dominated by two types of clouds: very small cumuli, which in large numbers detrain immediately above cloud base, and deeper clouds, which detrain just beneath and above the trade inversion. The intrusion of cumulus clouds into the stable layer above the inversion is through overshooting of cumuli above their level of zero-buoyancy (Nitta, 1975). Because of the coarse resolution employed in large-scale models, where the vertical gridlength is typically 50 to 100 hPa, it is difficult to represent these two types. The scheme presented here ignores the effects of very small cumuli but tentatively accounts for the effects of overshooting cumuli, as we assume that a given fraction of the cloud ensemble penetrates into the inversion layer and detrains there into the environment. Thus, cloud air shall only partly detrain into the environment within the model layer that contains the zero-buoyancy level; the remaining fraction shall intrude into the next layer above and detrain there:

$$D_u^{(2)} = \frac{(1-\beta)\ (M_u)_{k+1/2}}{\Delta z} \qquad \text{for top layer } k$$

$$D_u^{(2)} = \frac{\beta\,(M_u)_{k+1/2}}{\Delta z} \qquad \text{for layer } k - 1 \tag{3.5.2.13}$$

where β is a tunable parameter (currently 0.15)

Although this parameterization is very crude, it clearly reproduces more realistic trade inversions than when the effect from overshooting cumuli is ignored $(\beta = 0)$, as then the inversion becomes too strong in the simulation and the cloud layer below the inversion too moist (see Tiedtke, 1989).

c. Midlevel convection

Midlevel convection, that is, convective cells which have their origin not in the boundary layer but start at levels above the boundary layer, often occur at rainbands at warm fronts and in the warm sector of tropical cyclones (Browning et al., 1973; Houze et al., 1976; Herzegh and Hobbs, 1980). These cells are probably formed by the lifting of low level air until it becomes saturated (Wexler and Atlas, 1959) and the primary moisture source for the clouds is from low-level large-scale convergence (Houze et al., 1976). Often a low-level temperature inversion exists that inhibits convection from starting freely from the surface; therefore convection seems to be initiated by lifting low-level air dynamically to the level of free convection. This occurs often in connection with mesoscale circulations which might be related to conditionally symmetric instability (Bennetts and Hoskins, 1979; Bennets and Sharp, 1982) or a wave-CISK mechanism (Emanuel, 1982).

Although it is not clear how significant the organization of convection in mesoscale rainbands is for the large-scale flow a parameterization should ideally account for both convective and mesoscale circulations. Such a parameterization, however, is presently not available and we must therefore rely on simplified schemes. Here we use a parameterization which in a simple way considers the finding of the diagnostic studies mentioned above. We assume that convection is activated when there is a large-scale ascent at lower levels, the environmental air is sufficiently moist, i.e., of relative humidity in excess of 90 %, and convectively unstable layer exists above. The free convection level is determined by lifting a parcel of environmental temperature and moisture content

$$T_u = \bar{T}, \qquad q_u = \bar{q}_v \tag{3.5.2.14}$$

adiabatically, allowing for condensational heating, and then checking for buoyancy. The upward mass flux is set equal to the vertical mass transport by the large-scale flow at that level:

$$(M_u)_B = \bar{\rho}_B \bar{w}_B \tag{3.5.2.15}$$

which ensures that the amount of moisture which is vertically advected through cloud base by the large-scale ascent is fully available for the generation of convective cells.

In addition to the injection of mass through cloud base, we assume again that cloud air is produced by moisture convergence above cloud base through lateral entrainment in the same way as for penetrative convection given by equation (3.5.2.12).

3.5.3 Discretization of the model equations

The flux divergence in the large-scale budget equations (3.5.1.1) and in the cloud equations (3.5.2.1) and (3.5.2.9) are approximated by centered finite differences as

$$\frac{1}{\bar{\rho}}\frac{\partial}{\partial z}(M \cdot a) = g \cdot \frac{M_{k+1/2} \cdot a_{k+1/2} - M_{k-1/2} \cdot a_{k-1/2}}{p_{k+1/2} - p_{k-1/2}} \tag{3.5.3.1}$$

The definition of the large-scale variables at half levels pose a problem, when the half-level values defined by linear interpolation of full-level values (e.g. $\bar{s}_{k+1/2} = 0.5\,(\bar{s}_k + \bar{s}_{k+1})$). In this case very noisy profiles evolve in time particularly with regard to humidity. Much smoother profiles are obtained when the half-level values are determined by downward extrapolation from the next full level above along a cloud-ascent through that level:

$$\begin{aligned} \bar{T}_{k+1/2} &= \bar{T}_k + \left(\frac{\partial T}{\partial p}\right)_{h_s}(p_{k+1/2} - p_k) \\ \bar{q}_{k+1/2} &= \bar{q}_k + \left(\frac{\partial q}{\partial p}\right)_{h_s}(p_{k+1/2} - p_k) \end{aligned} \tag{3.5.3.2}$$

where $h_s = s + L \cdot q_s$ is the saturation moist static energy. Using an extrapolation like (3.5.3.2) for calculating the downward transports is also more consistent with the calculations of the updrafts where cloud air is transported upwards through level $k + 1/2$ with the thermal state below that level and equally with the downdrafts which depend only on values of s and q_v above that level. Similarly, because of (3.5.3.2) the downward transport of environmental air through the same level accounts now only for thermal properties above that level. The choice of a moist adiabate for extrapolation is dictated by the property of the moist static energy which is, by convection in the absence of downdrafts, only changed through the fluxes of moist static energy

$$\left(\frac{\partial \bar{h}}{\partial t}\right)_{cu} = -\frac{1}{\bar{\rho}}\frac{\partial}{\partial z}[M_u\,(h_u - \bar{h})] \tag{3.5.3.3}$$

As the lines of the saturation moist static energy h_s through point $(p_{k+1/2}, \bar{T}_{k-1/2})$ and the updraft moist static energy are almost parallel, apart from entrainment effects, the difference $h_u - \bar{h}$ is little affected by the vertical discretization.

The ascent in the updrafts is obtained by vertical integration of (3.5.2.1). Starting at the surface the condensation level (= lowest half-level which is saturated or supersaturated and where buoyancy is met) is determined from an adiabatic ascent. The cloud profile above cloud base is determined layer by layer by first doing a dry adiabatic ascent with entrainment and detrainment included and then adjusting temperature and moisture towards a saturated state. The cloud parcel is finally checked for positive buoyancy, that is, for excess of the virtual static energy in the cloud over the environmental value

$$(s_v)_u \geq \bar{s}_v = s + C_p \cdot T \cdot \delta \cdot 0.608 \cdot q_v \tag{3.5.3.4}$$

and cloud top is defined as the level where the parcel loses buoyancy.

Finally, we mention that for numerical reasons the environmental air must not be convectively unstably stratified:

$$\bar{s}_{k-1/2} \geq \bar{s}_{k+1/2} \tag{3.5.3.5}$$

3.5.4 Melting of snow

Melting of snow is parameterized as stratiform precipitation (see section 3.6)

3.5.5 Evaporation of rain

The evaporation of convective rain is parameterized following a proposal of Kessler (1969), where the evaporation is assumed to be proportional to the saturation deficit $q_s - q_v$ and to be dependent on the density of rain $M_R\ (g/m^3)$

$$E = \alpha_1 \cdot (q_s - q_v) \cdot M_R^{13/20} \tag{3.5.5.1}$$

where α_1 is a constant being zero for $q_v > q_s$.

As the density of rain M_R is not given by the model it is convenient to express it in terms of the rain intensity R [g / (m^2 sec)] as

$$R = M_R\,(V_0 + w) \cong M_R \cdot V_0 \tag{3.5.5.2}$$

where V_0 is the mean fall speed of rain drops which again is parameterized following Kessler (1969).

$$V_0 = \frac{\alpha_2 \cdot M_R^{1/8}}{\sqrt{p/p_S}} \tag{3.5.5.3}$$

Thus we have

$$E = \alpha_1 \cdot (q_s - q_v)\left[\left(\frac{\sqrt{p/p_S}}{\alpha_2} \cdot R\right)^{13/20}\right]^{8/9} \tag{3.5.5.4}$$

Since the convective rain takes place only over a fraction C_c of the grid area the evaporation rate at level k becomes

$$E = C_c \cdot \alpha_1 \cdot (q_s - q_v) \left[\left(\frac{\sqrt{p/p_S}}{\alpha_2} \cdot \frac{R}{C_c} \right) \right]^{\alpha_3} \tag{3.5.5.5}$$

where the constants have the following values (Kessler, 1969)

$$\alpha_1 = 5.44 \cdot 10^{-4}, \quad \alpha_2 = 5.09 \cdot 10^{-3}, \quad \alpha_3 = 0.5777$$

In order to save computing time (with $\alpha_3 = 1/2$) we use slightly different values

$$\alpha_1 = 6.94 \cdot 10^{-4}, \quad \alpha_2 = 7.35 \cdot 10^{-3}, \quad \alpha_3 = 0.5$$

In view of the uncertainty of the fractional area of precipitating clouds a constant value of

$$C_c = 0.05$$

is assumed.

The evaporation rate is calculated implicitly in the model by means of

$$2g \frac{\partial}{\partial p} R^{1/2} = -A \tag{3.5.5.6}$$

which follows from

$$E = A \cdot R^{1/2} \qquad\qquad A = \alpha_1 (q_s - q_v) \left[\frac{\sqrt{p/p_S}}{\alpha_2} \cdot \frac{1}{C_c} \right]^{1/2} \tag{3.5.5.7}$$

and

$$E = \frac{1}{\rho} \cdot \frac{\partial R}{\partial z} = -g \cdot \frac{\partial R}{\partial p} \tag{3.5.5.8}$$

3.5.6 Stratocumulus

Stratocumulus convection is parameterized by means of a vertical diffusion scheme (Tiedtke et al., 1988). It is only applied in the boundary layer where it provides a net upward transport of moisture and cloud water to avoid the generation of saturated layers within a convective boundary layer.

The net effect of stratocumulus convection is given by the turbulent fluxes of heat, moisture and cloud water, which are parameterized on the basis of the eddy diffusion theory (cf., section 3.3):

$$\left(\frac{\partial s}{\partial t}\right)_{Sc} = \frac{1}{\rho} \cdot \frac{\partial}{\partial z}\left(\rho \cdot K \cdot \frac{\partial s}{\partial z}\right) \tag{3.5.6.1}$$

$$\left(\frac{\partial q_v}{\partial t}\right)_{Sc} = \frac{1}{\rho} \cdot \frac{\partial}{\partial z}\left(\rho \cdot K \cdot \frac{\partial q_v}{\partial z}\right) \tag{3.5.6.2}$$

$$\left(\frac{\partial q_w}{\partial t}\right)_{Sc} = \frac{1}{\rho} \cdot \frac{\partial}{\partial z}\left(\rho \cdot K \cdot \frac{\partial q_w}{\partial z}\right) \tag{3.5.6.3}$$

Equations (3.5.6.1), (3.5.6.2) and (3.5.6.3) are only applied below 900 hPa and when neither the criterion for deep convection nor for shallow convection is fulfilled. Cloud base is the condensation level for surface air. Then (3.5.6.1), (3.5.6.2) and (3.5.6.3) are applied considering the fluxes through cloud base and cloud top using the following parameterization of the eddy diffusion coefficient

$$K\,[m^2/s] = \begin{cases} K_{max} \cdot min\left(b \cdot \dfrac{q_w}{q_{wr}}, 1\right) & \text{cloud layers and cloud base} \\ K_{max} \cdot b \cdot \delta r & \text{cloud top entrainment} \\ 0 & \text{elsewhere} \end{cases}$$

where b is cloud cover (cf., equation (3.6.1.4)), q_{wr} is a cloud water threshold (currently $3 \cdot 10^{-5}$), δr is the relative humidity jump at cloud top and K_{max} is a specified upper limit of K (currently $10\ m^2/s$).

3.6 STRATIFORM CLOUDS

The equations relevant for the discussion of the stratiform cloud scheme are the budget equations for the mass mixing ratio of water vapour (q_v) and cloud water ($q_w = q_l + q_i$), respectively, where q_l is the liquid fraction and q_i is the solid fraction. For convenience, the governing equations are given in a compact form according to

$$\frac{\partial q_v}{\partial t} = R(q_v) - C + E \tag{3.6.1}$$

$$\frac{\partial q_w}{\partial t} = R(q_w) + C - P \tag{3.6.2}$$

where $R(q_v)$ and $R(q_w)$ denote the sum over all advective and sub-grid scale transports of q_v and q_w, respectively. The cloud microphysical terms are the condensation of water vapour $(C > 0)$, the evaporation of cloud water $C < 0$, the formation of precipitation by coalescence of cloud droplets and sedimentation of ice crystals $(P > 0)$, and the evaporation of precipitation in unsaturated air $(E > 0)$.

3.6.1 Sub-grid scale cloud formation

Since real clouds are often smaller than the size of a grid box, sub-grid scale cloud formation is taken into account. The formalism has been developed by Sundquist (1978). Assuming that a fractional horizontal area (b) of a grid box is covered by clouds while the remaining part $(1-b)$ is cloud-free, the equations (3.6.1) and (3.6.2) are modified according to

$$\frac{\partial q_v}{\partial t} = R(q_v) - b \cdot C_c - (1-b) \cdot C_o + (1-b) \cdot E_o \tag{3.6.1.1}$$

$$\frac{\partial q_w}{\partial t} = R(q_w) + b \cdot C_c + (1-b) \cdot C_o - b \cdot P_c \tag{3.6.1.2}$$

where the subscript (c) denotes the respective process in the cloudy part and the subscript (o) the process in the cloud-free part of the grid box. The transport terms are assumed to be identical in both parts as well as temperature and wind. Evaporation of precipitation is not allowed in the cloudy part, i.e. $E_c = 0$, and there is no precipitation formation outside the cloud, i.e. $P_o = 0$. The significance of C_o, which is formally retained in eqs. (3.6.1.1) and (3.6.1.2) will be discussed below. In the cloudy part, saturation with respect to the grid averaged temperature T is assumed, $q_c = q_s(T)$, while the cloud-free part is unsaturated by definition, i.e. $q_o < q_s(T)$. Hence, fractional cloud cover (b) and grid-mean water vapour mixing ratio (q_v) are related according to

$$q_v = b \cdot q_s + (1-b) \cdot q_o \tag{3.6.1.3}$$

or, in terms of the relative humidity ($r = q_v / q_s$) and solving for (b)

$$b = \frac{r - r_o}{1 - r_o} \tag{3.6.1.4}$$

for $r > r_o$ and $b = 0$ otherwise. The relative humidity in the cloud-free part (r_o) has to be specified or parameterized. The choice of r_o should depend on factors such as grid resolution and sub-grid scale variance of vertical velocity (Sasamori, 1975). Presently, we specify r_o as a function of height and stability only: In stable stratification, r_o decreases linearly from the surface layer ($r_o = 0.99$) to the top of the PBL ($r_o = 0.85$) above which r_o remains constant. If penetrative convection occurs, r_o decreases further to a minimum value of 0.5 near the tropopause , as suggested by Xu and Krüger (1991) who showed in a modelling study of a cumulus ensemble that convectively driven stratiform clouds such as cirrus anvils may form already in a relatively dry environment.

3.6.2 Condensation and evaporation

Condensational growth of cloud droplets is assumed if the grid-mean relative humidity r exceeds the threshold r_o. Oppositely, an existing cloud will be diluted by evaporation if $r < r_o$. In the case of cloud formation by moisture convergence or adiabatic cooling, for example, any supersaturation in the cloudy part of the grid box will result in a condensational growth of cloud droplets.

A basic problem is to specify the partitioning of the net moisture convergence between the b-part and the $(1-b)$-part of the grid box.

According to the definition (3.6.1.3), $\frac{\partial q_v}{\partial t}$ may be decomposed into three parts,

$$\frac{\partial q_v}{\partial t} = b \cdot \frac{\partial q_s}{\partial t} + (1-b) \cdot \frac{\partial q_o}{\partial t} + (q_s - q_o) \cdot \frac{\partial b}{\partial t} \tag{3.6.2.1}$$

which represent the changes of q_s, q_o and b due to temperature changes and moisture convergence. Inserting (3.6.2.1) into (3.6.1.1) and separating the b-terms, provides the condensation (evaporation) rate C_c as a function of moisture convergence (divergence) and adiabatic cooling (heating),

$$b \cdot \frac{\partial q_s}{\partial t} = b \cdot R(q_v) - b \cdot C_c \tag{3.6.2.2}$$

According to (3.6.2.2), the b-fraction of the moisture convergence will be used for condensation while the remaining part, $(1-b) \cdot R(q_v)$, increases the relative humidity in the cloud-free part. The condensation rate is calculated from the moisture convergence into pre-existing clouds only.

Furthermore, we assume that there is always an abundance of cloud condensation nuclei and ice nuclei so that condensational growth is allowed to start as soon as the 100 % relative humidity threshold in the b-part of the grid box is exceeded. The saturation water vapour pressure is calculated from Tetens formula (Lowe, 1977) with suitable coefficients for the liquid phase and ice phase, respectively.

It remains to calculate the cloud water evaporation rate (C_o) in the cloud-free part of the grid box. Cloud water outside the b-part may be generated by advective or diffusive transports across the boundaries of the grid box while internal mixing by molecular or turbulent diffusion is neglected. Moreover, numerical effects do contribute also as, for example, the representation of cloud water by spherical harmonics in the host model. In these cases we assume that cloud water in the $(1-b)$ -part will be evaporated instantaneously,

$$R(q_w) + C_o = 0 \tag{3.6.2.3}$$

Note that all numerical effects are included in the transport term $R(q_w)$ which may also become negative. In this case, C_o is positive and actually represents a condensation rate.

3.6.3 Precipitation formation

Analogous to equation (3.6.1.3), the grid-mean cloud water mixing ratio is given by

$$q_w = b \cdot q_c \tag{3.6.3.1}$$

assuming that all of the cloud water is confined to the cloudy part of the grid box which implies that the in-cloud mixing ratio is defined as $q_c = q_w / b$.

The mechanism of precipitation formation depends crucially on the cloud water phase. Since we apply only one budget equation (3.6.1.2) for the cloud water mixing ratio, the liquid and ice phases are diagnosed as a function of temperature T according to

$$q_c = q_{cl} + q_{ci} = q_c \cdot f_l(T) + q_c \cdot f_i(T) \tag{3.6.3.2}$$

with $f_l + f_i = 1$. The fractions f_l and f_i have been obtained from a wealth of aircraft measurements, as compiled by Matveev (1984), by applying an exponential fit to these data (Rockel et al., 1991),

$$f_l(T) = 0.0059 + 0.9941 \cdot e^{-[0.003102\,(T-273.15)^2]} \tag{3.6.3.3}$$

The growth of cloud droplets to precipitating rain drops by autoconversion is parameterized in a convenient exponential form (Sundquist, 1978). In addition, the collision of cloud droplets with larger rain drops is taken into account (Smith, 1990) so that the total coalescence rate is given by

$$P_{cl} = q_{cl}\left[C_{r1}\left[1 - e^{-(q_{cl}/q_{cr})^2}\right] + C_{r2}\langle P\rangle\right] \tag{3.6.3.4}$$

where $\langle P\rangle$ is the rain flux density (stratiform and convective) at the top of the respective cloud layer and C_{r1}, C_{r2} and q_{cr} are microphysical constants which determine the efficiency of rain formation and, thus, the cloud lifetime (in ECHAM3/T42 $C_{r1} = 10^{-4} s^{-1}$, $C_{r2} = 1\ m^2 kg^{-1}$ and $q_{cr} = 3 \cdot 10^{-4}$).

Ice crystals settle at a rate which depends on the form and size of the crystals. Both parameters are not available in the model. However, according to an observational study by Heymsfield (1977), the terminal velocity of the crystals can be parameterized in terms of the ice water mixing ratio which is a model variable,

$$V_t = \alpha \cdot (\rho_a q_{ci})^{\beta} \tag{3.6.3.5}$$

where ρ_a is the air density, and the constants α and β are obtained from a best fit to the data with a slight adjustment (tuning) in the respective model version (in ECHAM3/T42, $\alpha = 1.77$ and $\beta = 0.16$). The loss of ice crystals due to sedimentation is given by the divergence of the ice flux density,

$$P_{ci} = g \frac{\partial}{\partial p}(V_t \rho_a q_{ci}) \tag{3.6.3.6}$$

where p is the pressure and g is the acceleration of gravity. The total rate of precipitation formation is given by

$$P_c = P_{cl} + P_{ci} \tag{3.6.3.7}$$

A crucial assumption in the parameterization (3.6.3.4) - (3.6.3.7) is that the precipitation formation in a mixed phase, i.e. in a temperature range between about 0 ^{o} C and -40 ^{o} C, can be treated independently for the ice phase and the liquid phase, respectively. A proper treatment of the interaction between both phases, such as the rapid condensational growth of ice crystals at the expense of cloud droplets (Bergeron-Findeisen process), will require a more elaborate scheme which should be based on the budget equations for each phase.

3.6.4 Evaporation of precipitation

Precipitation falling into the cloud-free part of a grid box is exposed to evaporation which is parameterized in terms of the saturation deficit according to

$$E_o = -\frac{1}{\Delta t} \cdot \frac{\gamma \cdot (q_o - q_s)}{1 + \dfrac{L \cdot dq_s/dT}{C_{pd} \cdot (1 + (\delta - 1)\, q_v)}} \tag{3.6.4.1}$$

where γ is a tunable parameter (currently $\gamma = 0.1$). The evaporation rate in a layer with thickness Δp is limited, however, by the precipitation flux density at the top of the layer, where $\langle \ \rangle$ denotes a vertical integral of the respective quantity,

$$\langle E_o (p + \Delta p) \rangle \le \langle P (p) \rangle \tag{3.6.4.2}$$

with

$$\langle P (p) \rangle = \frac{1}{g} \int_o^p [b \ P_c - (1 - b)\, E_o]\, dp' \tag{3.6.4.3}$$

and

$$\langle E_o (p + \Delta p) \rangle = \frac{1}{g} \int_p^{(p + \Delta p)} E_o \ dp' \tag{3.6.4.4}$$

3.6.5 Melting of snow

Melting of snow is parameterized considering observational data summarized by Mason (1971). Melting occurs in a thin layer of a few 100 m below the freezing level. We therefore assume that the snow can melt in each layer whenever the temperature exceeds $2\,^{o}\mathrm{C}$. The melting is limited not only by the snow amount but also by keeping the included cooling of the layer such that the temperature of the layer after melting is not less than the $2\,^{o}\mathrm{C}$ threshold.

3.7 SOIL PROCESSES

3.7.1 General

The surface parameterization scheme comprises the evolution of a temperature profile in the soil, the soil hydrology and the snow pack over land.

The distribution of gridpoint characteristics is given according to the land-sea mask (Figure 9). The characteristics of sea ice are assumed if the sea surface temperature is below -1.8 ^{o}C. An extra mask specifies glacier areas over land covered permanently with ice.

The temperature at the interface between the atmosphere and the surface is always called T_S. This variable either contains the sea surface temperature, the temperature at the top of the snow pack, or the temperature of the upper soil layer (Figure 10).

W_S is the soil moisture in [m] and W_l is the water content of the interception reservoir.

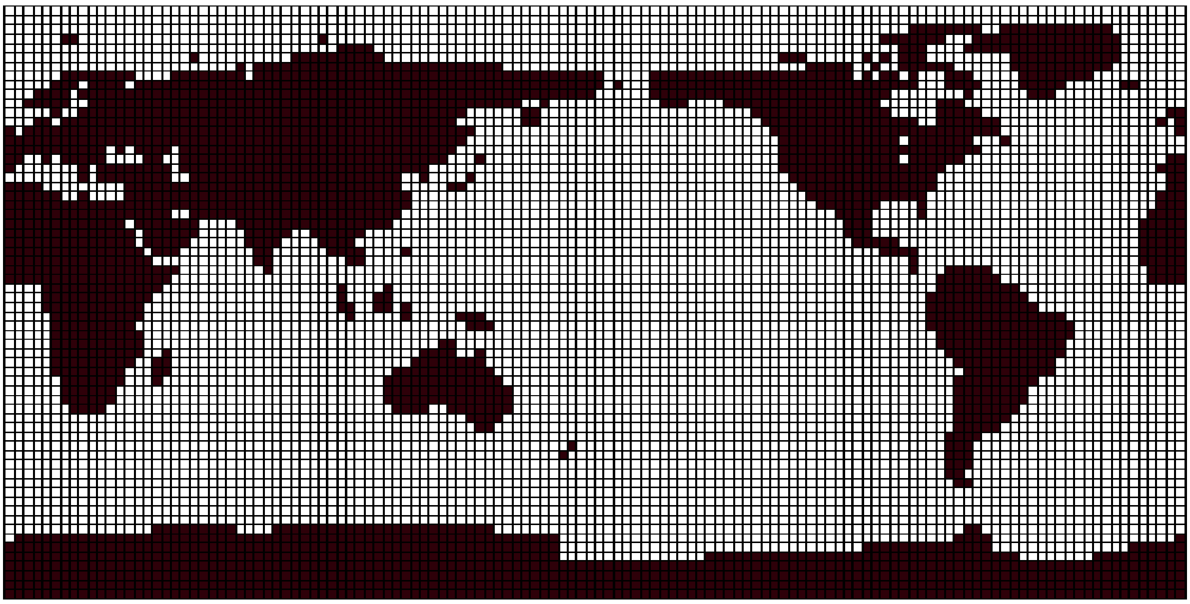


Figure 9 Land-sea mask of the ECHAM3 model for the T42-truncation

3.7.2 Soil temperature

The soil is divided into five layers according to Figure 10. The thicknesses of the individual soil layers increase with depth.

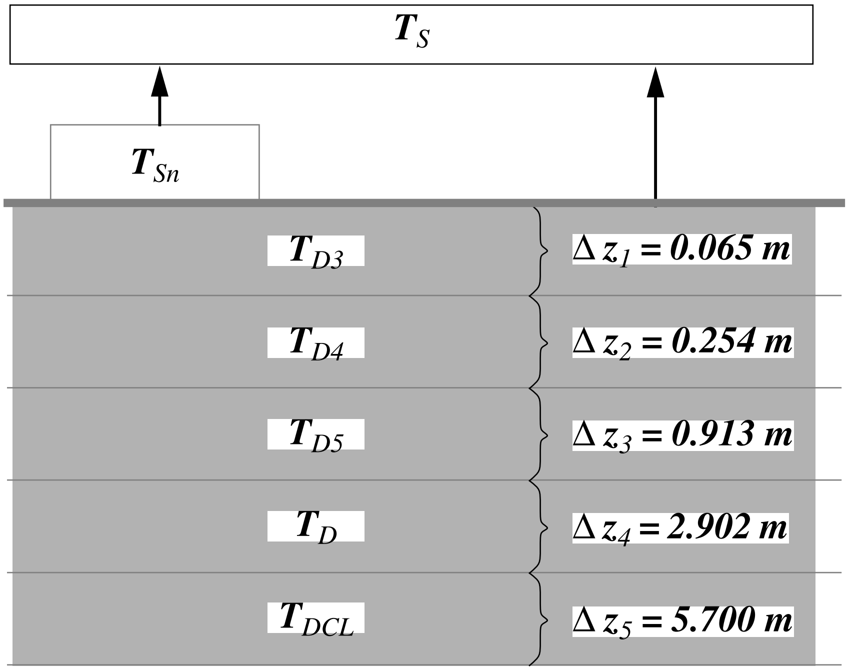


Figure 10 Description of soil temperatures in the ECHAM3 model.[1]

This is an extension of the scheme of Warrilow et al. (1986) to five layers. At the lowest layer boundary a zero heat flux condition is prescribed in order to ensure that no artificial heat sources and sinks may affect the energy balance of the earth-atmosphere system. The heat conduction equation follows from

$$\begin{aligned} \frac{\partial T_1}{\partial t} &= \frac{F_S}{\rho_g C_g \Delta z_1} + \frac{2\kappa (T_2 - T_1)}{\Delta z_1 (\Delta z_1 + \Delta z_2)} && \text{(layer 1)} \\ \frac{\partial T_i}{\partial t} &= -\frac{2\kappa (T_i - T_{i-1})}{\Delta z_i (\Delta z_{i-1} + \Delta z_i)} + \frac{2\kappa (T_{i+1} - T_i)}{\Delta z_i (\Delta z_i + \Delta z_{i+1})} && \text{(layers 2 to 5)} \end{aligned} \tag{3.7.2.1}$$

1. Note that for technical reasons the variable T_S is always the interface temperature to the atmosphere. In snow covered areas it contains the skin temperature of the snow pack extrapolated to the top of the snow pack from T_{Sn} at the middle of the snow pack.
If there is no snow it contains T_{D3} .

with

$T_1 = T_{D3}$	temperature for layer 1	[K]
$T_2 = T_{D4}$	temperature for layer 2	[K]
$T_3 = T_{D5}$	temperature for layer 3	[K]
$T_4 = T_D$	temperature for layer 4	[K]
$T_5 = T_{DCL}$	temperature for layer 5	[K]
κ	heat diffusivity in the soil	$7.5 \cdot 10^{-7}\ \mathrm{m^2 s^{-1}}$
$\rho_g \cdot C_g$	heat capacity of the soil per unit volume or	$2.4 \cdot 10^6\ \mathrm{J\ m^{-3}\ K^{-1}}$
	heat capacity of land ice per unit volume	$2.09 \cdot 10^6\ \mathrm{J\ m^{-3}\ K^{-1}}$
F_S	sum of the radiative and turbulent fluxes at the surface if there is no snow; heat flux from the snow to the deep soil if the snow depth exceeds 0.025 m.	

NOTE: All soil characteristics are assumed to be the same over all continents.

3.7.3 Snow pack temperature

The scheme accounts for the three different conditions at the surface boundary in the presence of a snow pack over land:

1) In the case of permanent ice cover at a grid cell, the soil heat equation Eq.(3.7.2.1) is solved assuming the characteristics of ice. These areas are defined by the glacier mask GLAC.
2) For a snow pack of less than 0.025 m water equivalent (3.7.2.1) is solved assuming the characteristics of the bare soil.
3) For snow depths deeper than 0.025 m, an extra heat conduction equation (Bauer et al., 1986) evolves according to

$$\frac{\partial}{\partial t} T_{Sn} = \frac{F_S}{\rho_{Sn} \cdot C_{Sn} \cdot Sn} \tag{3.7.3.1}$$

with

T_{Sn}	temperature in the middle of the snow pack	
F_S	sum of the radiative and turbulent fluxes at the surface	
$\rho_{Sn} C_{Sn}$	heat capacity of the snow per unit volume	$0.6345 \cdot 10^6\ \mathrm{J\ m^{-3}\ K^{-1}}$
	computed using a density of snow ρ_{Sn} of	$300\ \mathrm{kg\ m^{-3}}$
Sn	depth of the snow pack	

3.7.4 Snow melt

The temperature T_{Sn} is used to define a temperature T_S at the top of the snow pack by extrapolation. T_S serves as an interface to the atmosphere. This temperature may not exceed the snow melt temperature T_{melt}. If sufficient energy is available to raise the temperature T_{Sn} above T_{melt}, this energy is used to warm the soil underneath. Only if both T_{D3} and T_{Sn} reach the value of T_{melt}, further energy input will be used to melt the snow. Snow that is less deep than 0.025 m may melt if T_{D3} equals T_{melt}.

3.7.5 Sea-ice temperature

The ice-surface temperature T_{ice} is calculated from the energy balance at the ice surface. To avoid problems due to a large time step it is assumed that T_{ice} represents the uppermost 10 cm of the ice sheet. The energy balance equation for this layer can be written as

$$C_{pi}\left(\frac{dT_{ice}}{dt}\right) = F_{atm} + F_{oce} \tag{3.7.5.1}$$

where C_{pi} is the heat capacity of a 10 cm ice layer. F_{atm} is the atmospheric heat flux consisting of the solar and thermal radiation and the sensible and latent heat flux. The oceanic heat flux F_{oce} is parameterized as

$$F_{oce} = \frac{\alpha}{h_{ice}} \cdot (T_{oce} - T_{ice}) \tag{3.7.5.2}$$

where α is a heat transfer coefficient (2 W/mK), h_{ice} is the ice thickness and T_{oce} is the temperature of the underlying ocean.

To solve equation (3.7.5.1) the atmospheric heat flux is linearized with respect to the surface temperature at the previous time step T_S^{old}

$$F_{atm}(T_{ice}) = F_{atm}(T_S^{old}) + \frac{\partial}{\partial T}F_{atm}(T_{ice} - T_S^{old}) \tag{3.7.5.3}$$

This leads to a linear equation for the new ice-surface temperature T_S^{new}

$$T_S^{new} = \frac{F_{atm} - \frac{\partial}{\partial T}F_{atm}T_S^{old} + \frac{\alpha}{h_{ice}} \cdot T_{oce} + \frac{C_{pi}}{\Delta t} \cdot T_{ice}^{old}}{\frac{C_{pi}}{\Delta t} + \frac{\alpha}{h_{ice}} - \frac{\partial F_{atm}}{\partial T}} \tag{3.7.5.4}$$

3.7.6 Soil hydrology

The parameterisation of soil hydrology comprises three budget equations for

i. snow amount Sn (m water equivalent) accumulated at the surface,

ii. water amount W_l intercepted by the vegetation during rain or snow melt episodes (the so-called skin reservoir),

iii. soil water amount W_S.

The water equivalent of the snow layer is computed over land and glacier areas from

$$\frac{\partial}{\partial t}Sn = \frac{J_{q_{Sn}} + P_{Sn} - M_{Sn}}{\rho_w} \qquad (3.7.6.1)$$

with

$J_{q_{Sn}}$	evaporation rate per unit area over the snow pack
P_{Sn}	snow fall rate per unit area
M_{Sn}	snow melt rate per unit area
ρ_w	density of water

Rain water and melting snow on the leaves is intercepted by the vegetation until its water holding capacity W_{lmx} (cf. equation (3.3.2.5)) is exceeded. The corresponding budget equation is given by

$$\frac{\partial W_l}{\partial t} = \frac{J_{q_{vi}} + C_{ip} \cdot C_v \cdot (C_a \cdot P_R + M_{Sn})}{\rho_w} \qquad (3.7.6.2)$$

with

$J_{q_{vi}}$	evaporation rate from the skin reservoir (cf., section 3.3.2)
P_R	rainfall rate per unit area
C_v	fraction of the grid box covered with vegetation (cf., (3.3.2.6))
C_{ip}	coefficient of efficiency of rain and snow melt interception (currently 50 %)
C_a	fractional area wetted by rain during a time step (currently 100 % for large-scale rain and 50 % for convective rain

The amount of rain and snow melt which does not enter the skin reservoir is used to calculate the amount of soil infiltration and surface runoff. The soil water reservoir evolves according to

$$\frac{\partial W_S}{\partial t} = \frac{J_{q_v} - J_{q_{vi}} + P_R - P_{Ri} + M_{Sn} - M_{Sni} - R_R - R_D}{\rho_w} \qquad (3.7.6.3)$$

with

J_{q_v}	grid-mean evaporation rate per unit area according to (3.3.2.15)
P_{Ri}	rainfall rate per unit area intercepted by the skin reservoir
M_{Sni}	snow melt rate per unit area intercepted by the skin reservoir
R_R	surface runoff rate per unit area from precipitaton events and snow melt
R_D	runoff rate per unit area from drainage processes

The computation of R_D and R_R follows the scheme by Dümenil and Todini (1992) which is based on catchment considerations. The scheme accounts for the heterogeneity of a grid area by assuming that the total field capacity W_{Smax} for the grid area results from the integral of the local field capacities which are distributed over the grid area in a non-linear way. The resulting fractional saturated area s/S is a function of the degree of the grid-mean relative soil wetness W_S/W_{Smax} and a structure parameter b that defines the sub-grid scale characteristics of the basin or grid box:

$$\frac{s}{S} = 1 - \left(1 - \frac{W_S}{W_{Smax}}\right)^b \tag{3.7.6.4}$$

Equation (3.7.6.4) defines a fractional saturated area s/S of a grid box where runoff would occur for a certain rainfall (or snow melt) event, while in the area $1 - (s/S)$ the rainfall would infiltrate (cf., Figure 11). The surface runoff rate R_R is computed from the area integral of rain and snow melt that arrives in the saturated part s/S of the grid area. The amount of surface runoff resulting from a rainfall (or snow melt) event during a time interval Δt (e.g. the model time step) is computed from

$$\frac{1}{\rho_w}\int_{t_0}^{t_0+\Delta t} R_R dt = Q - (W_{Smax} - W_S) + W_{Smax}\left[\left(1 - \frac{W_S}{W_{Smax}}\right)^{\frac{1}{1+b}} - \frac{Q}{(1+b)\cdot W_{Smax}}\right]^{1+b} \tag{3.7.6.5}$$

if $[...] > 0$ or

$$\frac{1}{\rho_w}\int_{t_0}^{t_0+\Delta t} R_R dt = Q - (W_{Smax} - W_S) \tag{3.7.6.6}$$

if $[...] \le 0$ and $Q + W_S > W_{Smax}$

where

$$Q = \int_{t_0}^{t_0+\Delta t} \frac{(P_R - P_{Ri} + M_{Sn} - M_{Sni})}{\rho_w} dt \tag{3.7.6.7}$$

is the total water available for infiltration and runoff after possible interception in the skin reservoir (cf., equation (3.7.6.2)).

Equation (3.7.6.6) represents the well-known “bucket model” (Manabe, 1969) where runoff is computed if the precipitation event would cause an oversaturation of the whole grid box (i.e., $W_S + Q > W_{Smax}$). Note also, that the scheme (3.7.6.5) converges to the bucket model for $b \to 0$.

In the ECHAM model, the structure parameter b is parameterized in terms of the sub-grid scale height distribution which is taken as a measure of the typical steepness of the terrain in the respective grid box:

$$b = max\left[\frac{\sigma_h - \sigma_o}{\sigma_h + \sigma_{max}};0.01\right] \tag{3.7.6.8}$$

where σ_h is the standard deviation of the terrain height and σ_o and σ_{max} are prescribed minimum and maximum values of σ_h depending on the model resolution (currently, $\sigma_o = 100\ \mathrm{m}$ and $\sigma_{max} = 1000\,(1500)\ \mathrm{m}$ for T21 (T42) resolution).

According to the runoff parameterization (3.7.6.5) - (3.7.6.8), surface runoff is extremely efficient in steep terrain (b approaching 0.5 for $\sigma_h \to \sigma_{max}$) while most of the precipitation is allowed to infiltrate the soil if the terrain is relatively flat ($b \to 0.01$).

The infiltration rate per unit area I_R is defined as

$$\frac{1}{\rho_w}\int_{t_0}^{t_0+\Delta t} I_R dt = Q - \frac{1}{\rho_w}\int_{t_0}^{t_0+\Delta t} R_R dt \tag{3.7.6.9}$$

For frozen soil, however, we assume zero infiltration so that the surface runoff results as

$$\frac{1}{\rho_w}\int_{t_0}^{t_o+\Delta t} R_R dt = Q \tag{3.7.6.10}$$

Runoff due to drainage processes occurs independently of the water input Q if the soil wetness is between 5 % and 90 % of the field capacity (slow drainage) or larger than 90 % (fast drainage):

$$\frac{R_D}{\rho_w} = \begin{cases} d_{min}\cdot\dfrac{W_S}{W_{Smax}} \ \text{if}\ (W_{Smin} < W_S < W_{dr}) \\ d_{min}\cdot\dfrac{W_S}{W_{Smax}} + (d_{max} - d_{min})\left(\dfrac{W_S - W_{dr}}{W_{Smax} - W_{dr}}\right)^d \ \text{if}\ (W_S \ge W_{dr}) \end{cases} \tag{3.7.6.11}$$

with

$$d_{min} = 2.8 \cdot 10^{-10} \text{ m/s}$$
$$d_{max} = 2.8 \cdot 10^{-8} \text{ m/s}$$
$$d = 1.5$$
$$W_{Smin} = 0.05 \cdot W_{Smax}$$
$$W_{dr} = 0.9 \cdot W_{Smax}$$

The amount of evaporation is computed from the atmospheric demand (cf., section 3.3.2) but is limited by the soil moisture availability over bare soil and due to vegetation.

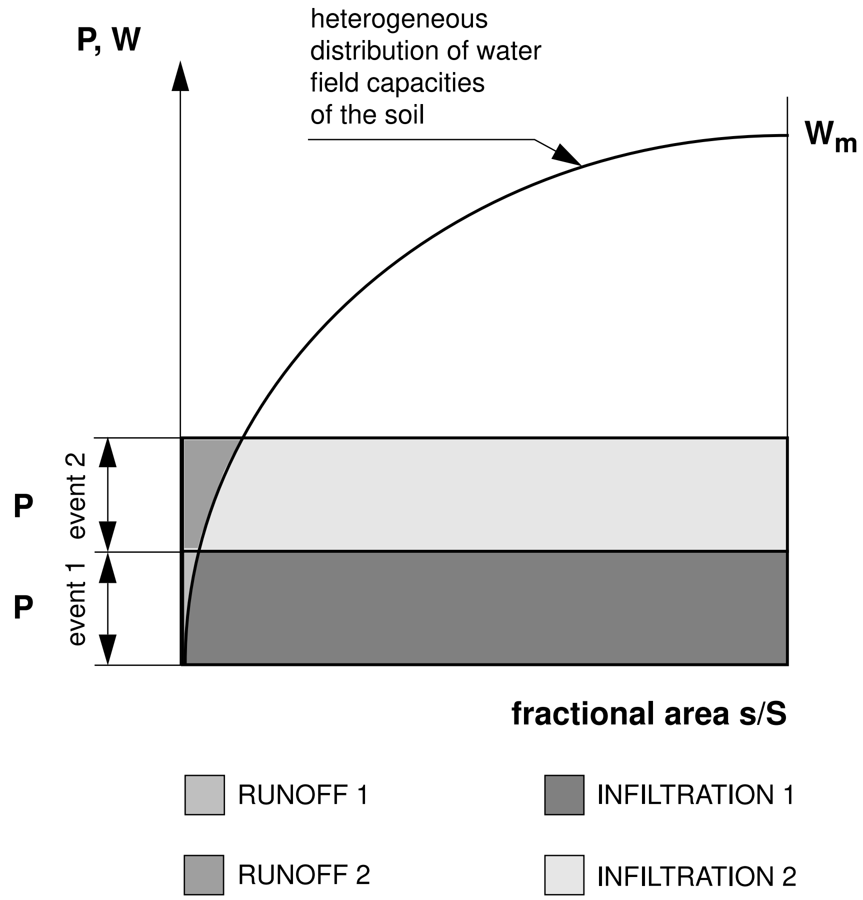


Figure 11 Surface runoff and infiltration from precipitation events.

4. SYSTEM DESCRIPTION

4.1 OVERVIEW AND THE CONTROL ROUTINES

Figure 12 is a diagram of the ECHAM3 climate model. It shows the main sections of the model as well as the name of the control routines. The main progam is called MASTER. Its function is to start the whole system by calling subroutine CONTROL.

Routine CONTROL first calls several subroutines to set up constants, default values and logical switches. The first subroutine called is INILOC, which initialises the memory manager. Then subroutine MAKESD is called to read and initialize namelist SDSCTL and store it in common block COMSDS. Subroutine INCOM which calls various subroutines to set up constants in most of the common blocks, and subroutine INILEG is called to compute Legendre coefficients needed for the Legendre transforms.

Depending on whether the run is the continuation of a previous run or a new run, CONTROL then calls either RESTART or START. The main function of these subroutines is to read the initial data and to set up the work files. The initial data exists as spherical harmonics for the atmosphere (prognostic variables) and as gridpoint data for the surface.

After labelling the run´s output (subroutine LABRUN), CONTROL finally calls subroutine STEPON which supervises the actual running of the model. Essentially, STEPON organizes the scanning through the data (Figure 13). STEPON also calls the subroutines which perform calculations in spectral space (part of the semi-implicit adjustment and horizontal diffusion), it increments the timestep and checks for the completion of the run.

The computation is divided into two scans through the latitude lines. The two scan structure is shown in Figure 13. During the first scan (shown in Figure 14), all the calculations in Fourier and grid-point space are done, then direct Legendre transforms produce spherical harmonics coefficients. After performing the end of the semi-implicit adjustment and the horizontal diffusion in spectral space, a second scan (shown in Figure 15) is used to go back to Fourier space by inverse Legendre transform.

All the physical parameterisation is done in grid point space. As shown in Figure 16, it is controlled by routine PHYSC, called by GPC. The routine which computes the radiative heating rates uses a forward timestep, from $t - \Delta t$ to $t + \Delta t$, using some input computed at less frequent interval, as described in section 2.5 . The vertical diffusion also uses a forward timestep, but it is performed implicitly in order to keep it numerically stable despite the fine vertical resolution near the ground. The moist convection modifies the tendencies previously computed, and the stratiform precipitation routine adjusts the $t + \Delta t$ moisture, cloud water and temperature fields.

The computations of the geopotential height and the pressure of the model levels are self-explanatory and not described here.

Overall the model computations may be divided into five main categories.

- Grid point computations.

  These include the computations of non linear dynamical terms, the parametrization of diabatic processes and the treatment of linear terms which could be calculated in spectral space but only at the expense of extra storage (time filtering, linear-combinatins of t and $t - \Delta t$ values of the variables associated with the semi-implicit schemes).

- Fourier transforms.

  These use a fast transform technique according to Temperton (1983) to convert from grid point to Fourier space and vice versa.

- Fourier space computations.

  These include computations of zonal derivatives and of some quantities related to the semi-implicit treatment of vorticity and humidity.

- Legendre transforms.

  These carry out conversion from Fourier to spectal space and vice versa. They include the computations of the meridional derivatives and of the Laplacian operator.

- Spectral computations.

  These comprise the final solution of equation (2.5.43) and the spectral forms of (2.5.31) and (2.5.32), that is the completion of the semi implicit scheme. Horizontal diffusion is also computed in spectral space.

4.2 FLOW DIAGRAM

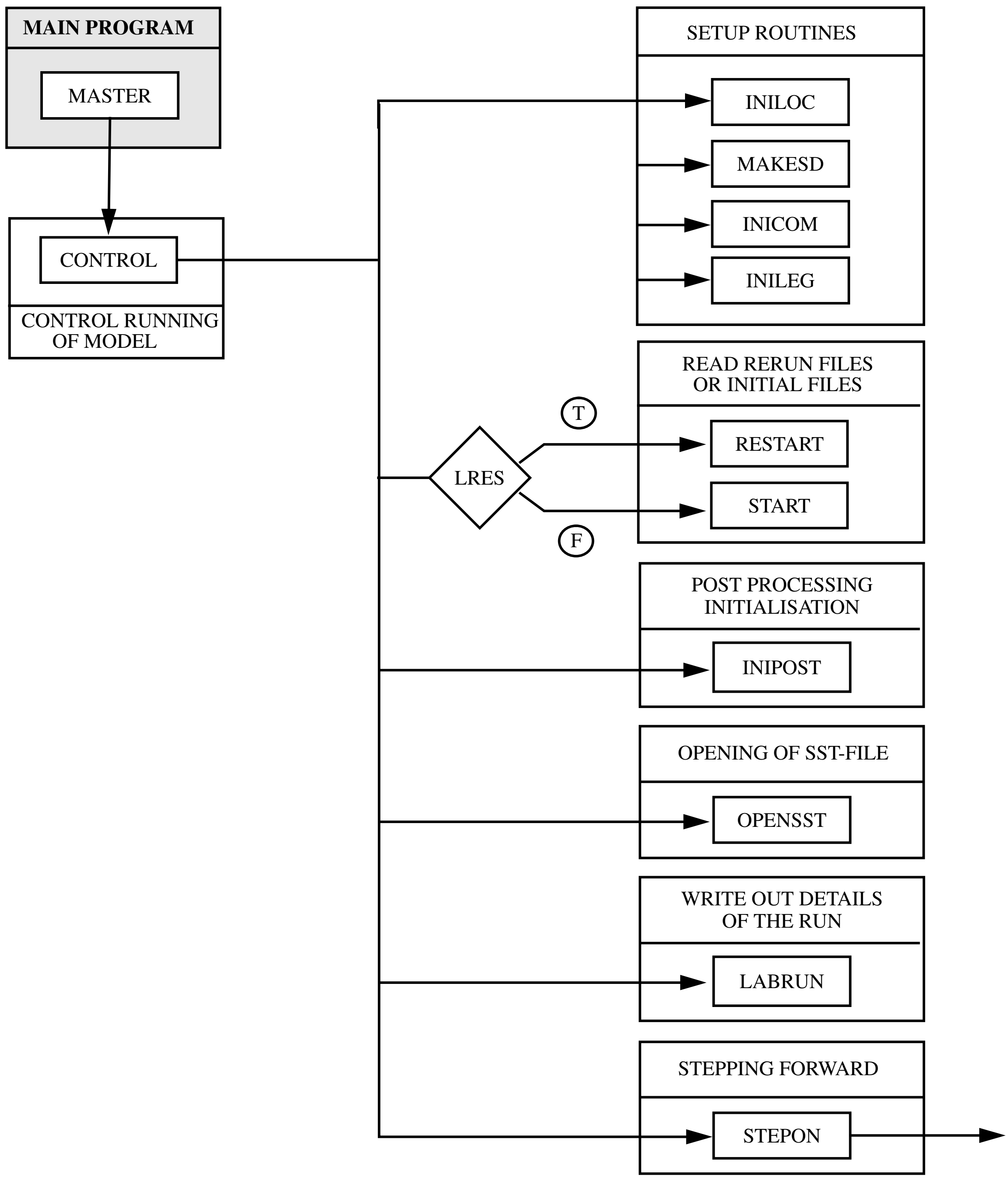


Figure 12 Flow diagram showing the initialization and control subroutines, which are managed by subrouitne CONTROL.

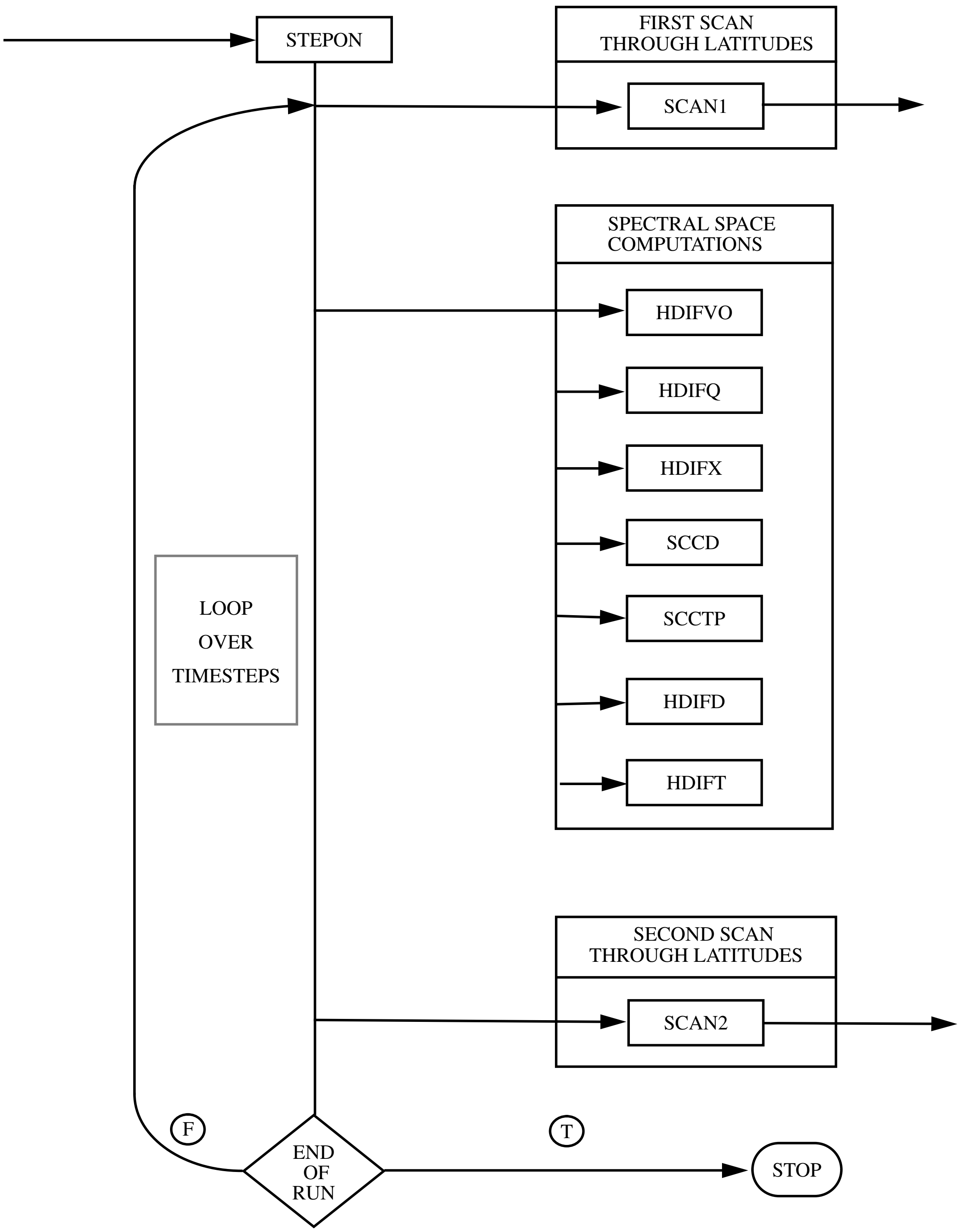


Figure 13 Flow diagram showing the two scan structure of the model, which is controlled by subroutine STEPON.

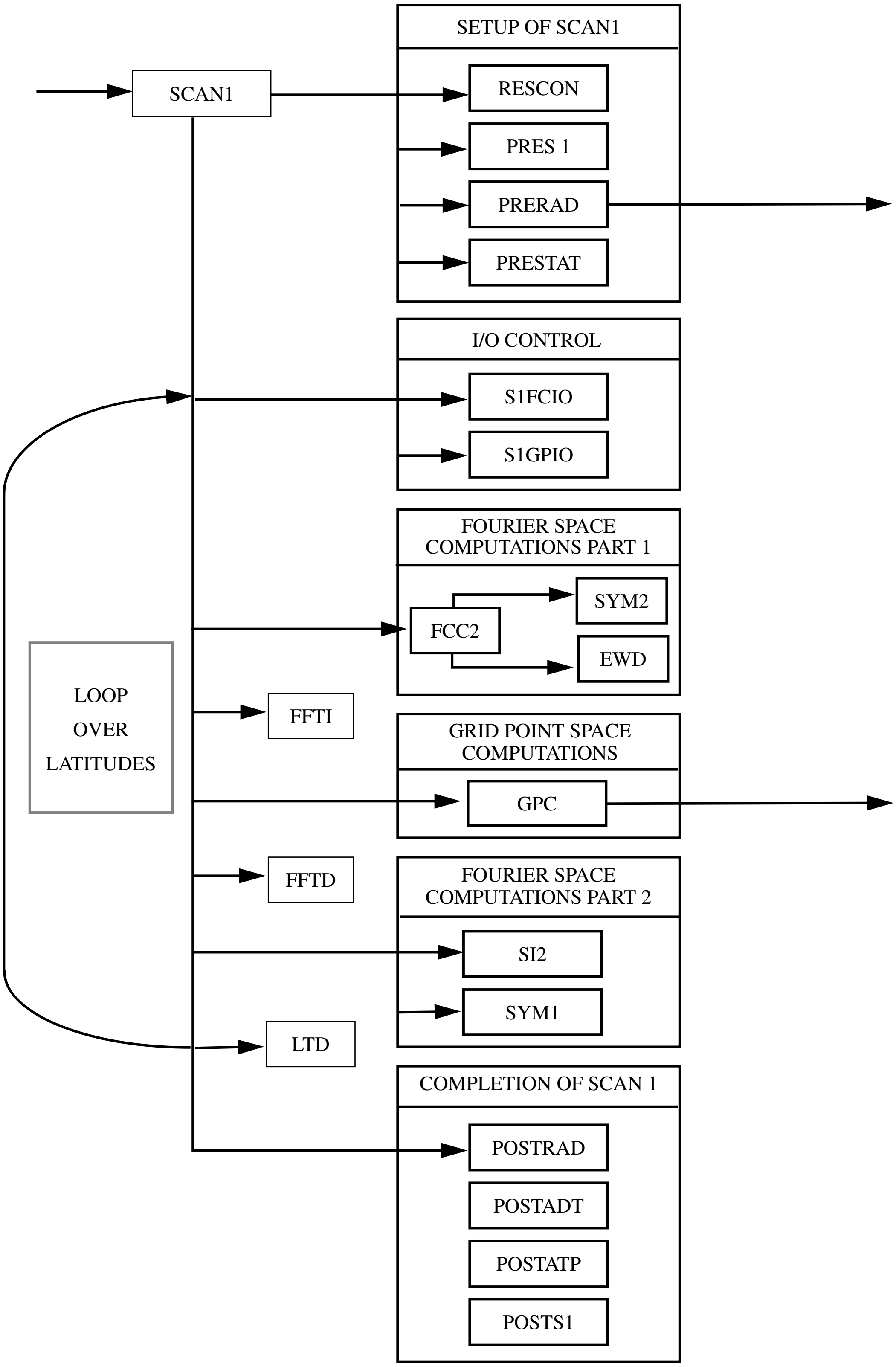


Figure 14 Flow diadram showing the detailed structure of the computations done in the first scan, which is controlled by subroutine SCAN1.

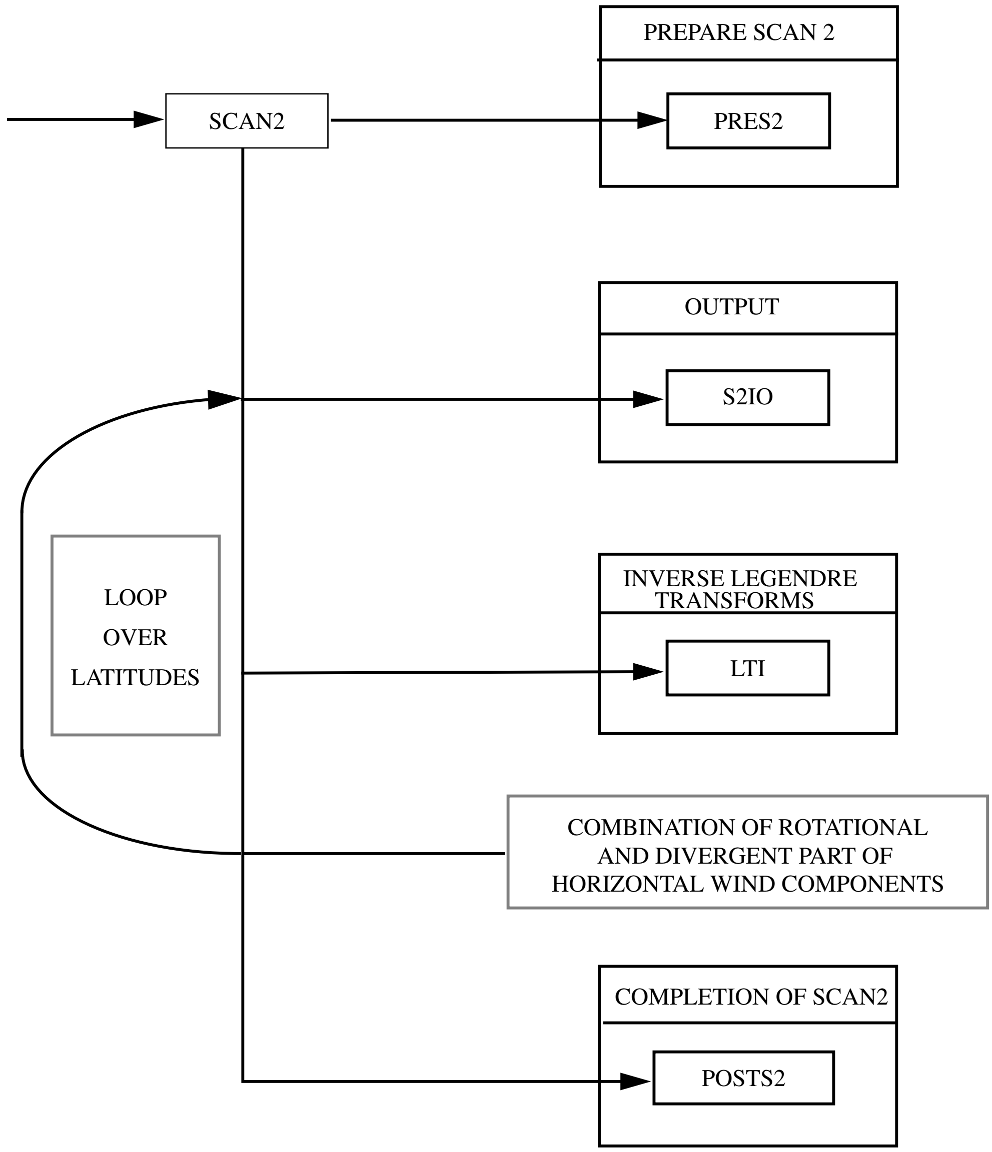


Figure 15 Simplified flow diagram showing the main computations done in the second scan, which is controlled by subroutine SCAN2.

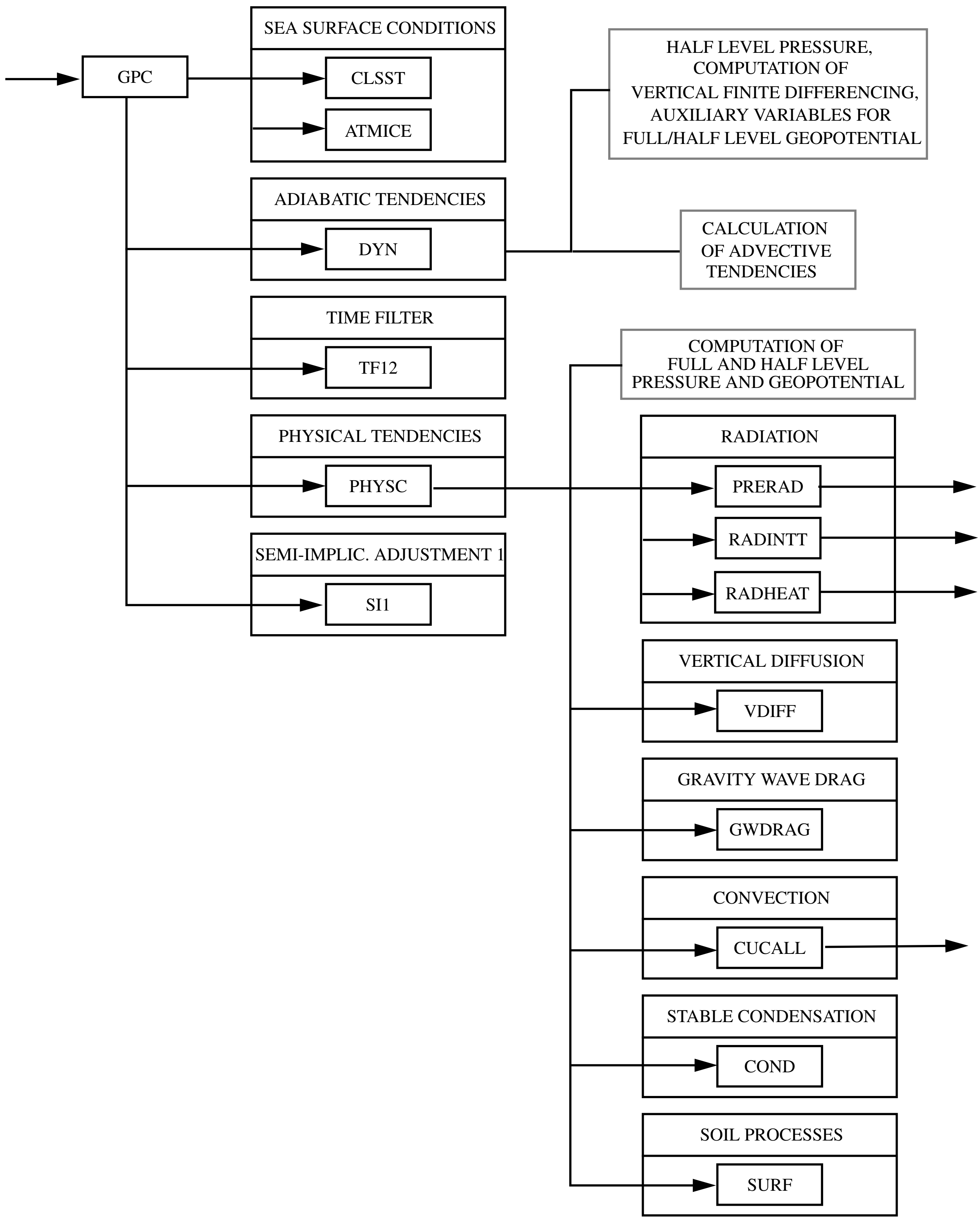


Figure 16 Flow diagram showing the grid point space computations done in the first scan controlled by subroutine GPC.

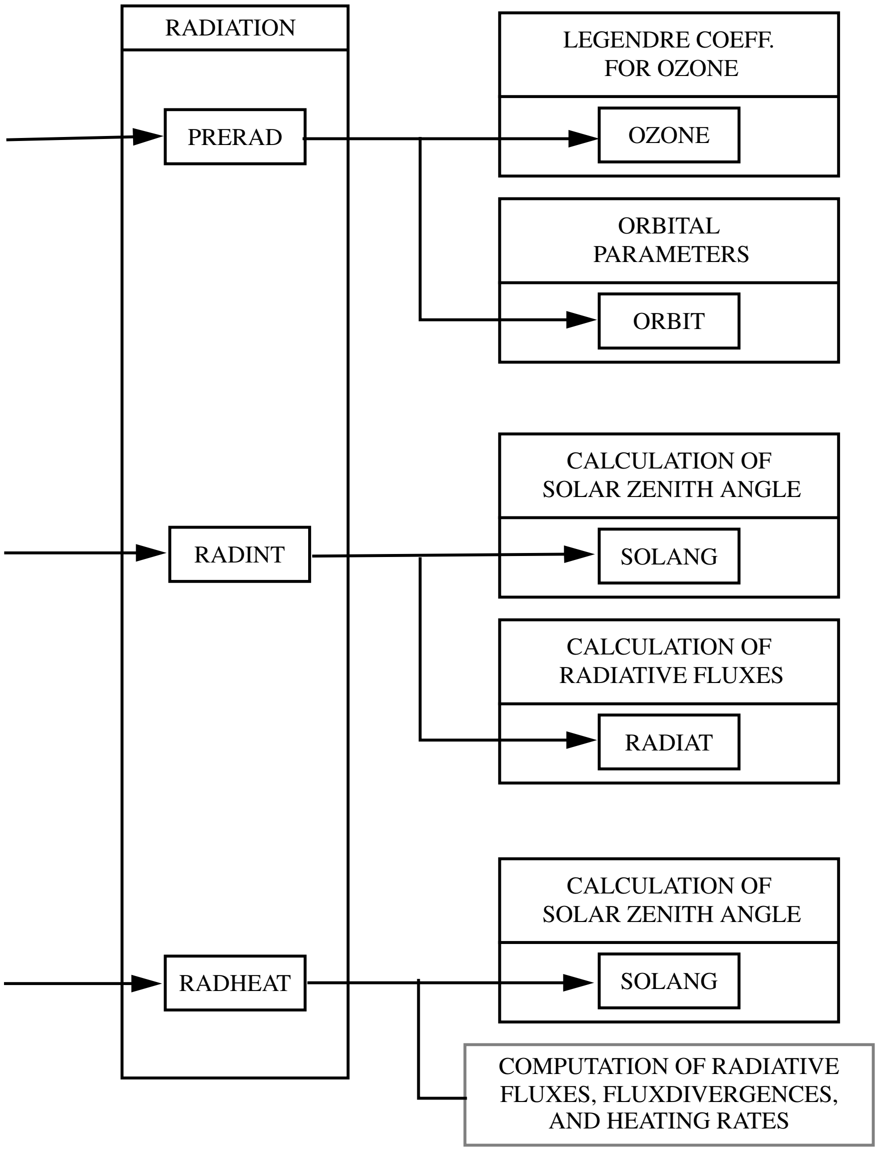


Figure 17 Simplified flow diagram showing the structure of the radiation scheme, which is called by subroutine GPC and controlled by subroutines PRERAD, RADINT and RADHEAT.

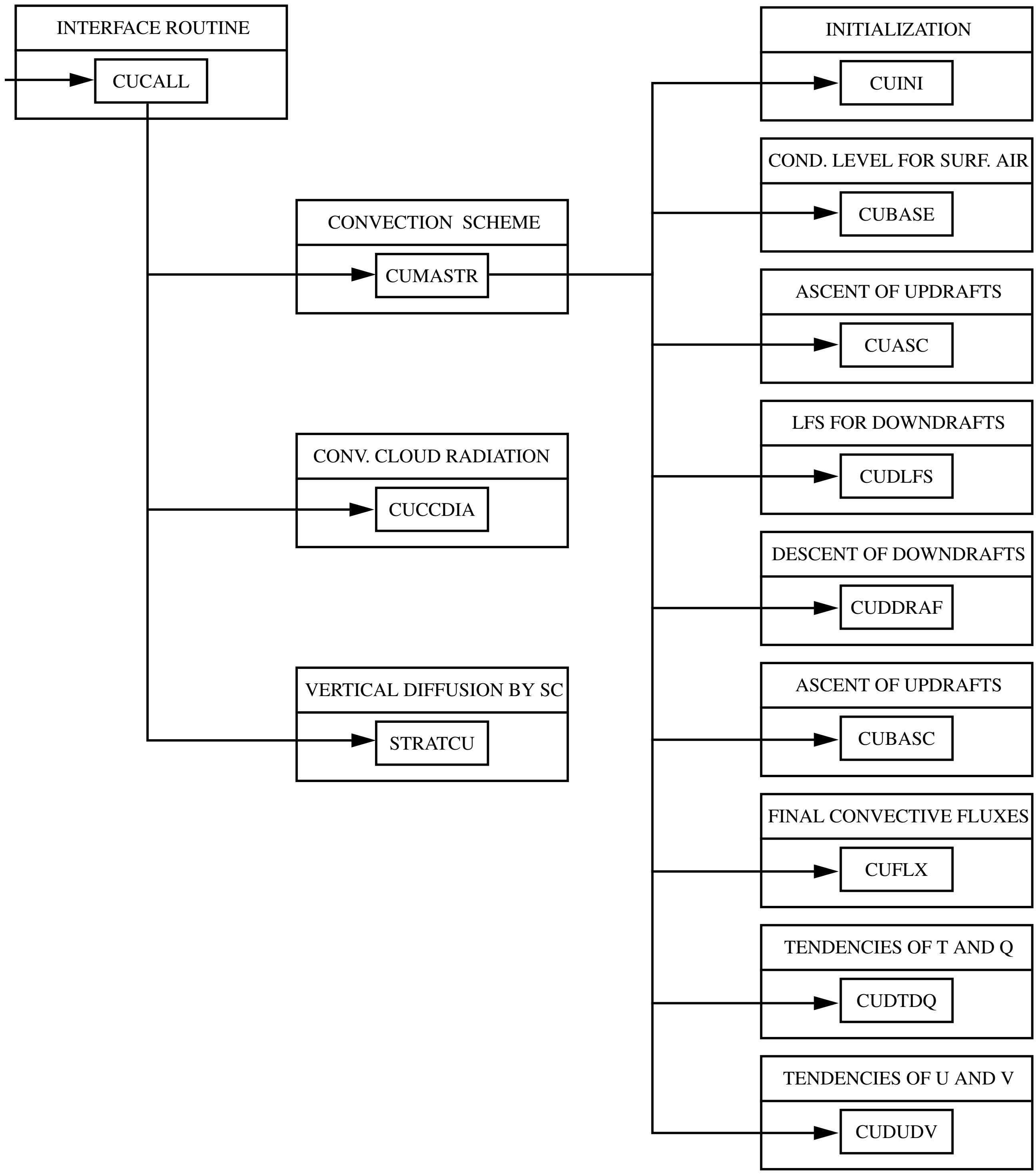


Figure 18 Flow diagram showing the structure of cumulus convection and stratocumulus schemes controlled by subroutine CUCALL.

4.3 COMPUTER CODE ORGANIZATION

4.3.1 The two scans

To save memory storage, it is necessary to store data on disk and scan through it in order only to keep in the central memory what is needed. For low resolution versions of the model (< T106) memory storage is not a problem. In this case an optional in-core version of the model can be used (LCORE=T in namelist RUNCTL). In this version the principle scanning structure of the model is the same but instead of writing to disk the data is copied into global memory buffers.

Grid point and Fourier space computations proceed one latitude row at a time. Therefore it is natural to store slices on disk or in the global memory buffer accordingly. However, when the data is represented in terms of spherical harmonics (what we also call the spectral space), the computations need a global set of spectral coefficients. These are kept in the central memory.

Subroutine SCAN1 reads through the data latitude row by latitude row. As shown in Figure 14, SCAN1 handles the computations in grid-point and Fourier space. Then it performs the direct Legendre transforms to accumulate the spectral coefficients. In order to perform the Legendre transforms efficiently it is necessary to compute the symmetric and antisymmetric parts of the Fourier components with respect to the equator. As a consequence the latitude lines are scanned alternately in the Northern and Southern hemispheres. Hence odd numbered lines go from the North pole to the equator, while even numbered lines go from the South pole to the equator.

In addition SCAN1 calls some diagnostic subroutines, as well as some subroutines to prepare the computation of radiation (PRERAD) and statistics (PRESTAT), and to print some output (POSTRAD, POSTATP, POSTATD, POSTS1).

After the computations in spectral space are finished, subroutine SCAN2 then performs the inverse Legendre transforms and stores the Fourier coefficients latitude row by latitude row, in the alternate fashion as described above. The part of the flow diagram, which is subject to the computations in subroutine SCAN2, is shown in Figure 15.

4.3.2 Dynamic subroutines

In this section we give some details about each subroutine involved in the adiabatic computation. The physical parametrization is controlled by subroutine PHYSC.

GPC

This subroutine controls all the grid-point computations, i.e. the adiabatic and physical tendencies (DYN and PHYSC), the time filtering (TF12) and the contribution of the current and previous time levels to the semi-implicit scheme (SI1). In addition, some global statistics for the dynamics and the physics are computed and printed (STATD and STATP). A schematic of the computations done in subroutine GPC is shown in Figure 16.

Some specialised subroutines are called by DYN and SI1: PRES computes the pressure at all half-levels, knowing the surface pressure. AUXHYB computes some auxiliary variables such as the pressure difference across the model layers and the logarithm of the ratio of the pressures at the half levels; GEOPOT integrates the hydrostatic equation to get the geopotential at all levels; PGRAD computes the geopotential and pressure gradient terms for the semi-implicit scheme; and CONTEQ calculates the temperature and surface pressure increments for the semi-implicit scheme.

FFTD and FFTI

Direct and inverse fast Fourier transforms respectively.

SI2

Computes the contributions to the semi-implicit scheme which can be performed using Fourier components. They are mainly related to the semi-implicit treatment of vorticity and humidity.

SYM1

Computes the symmetric and antisymmetric parts of the Fourier components in order to take full advantage of the symmetry properties of the Legendre polynomials in the Legendre transforms.

FCC2

This corresponds to the computations involving Fourier components done before the inverse Fourier transforms (FFTI). It consists of two blocks:

SYM2

Recomputes real Fourier components from their symmetric and antisymmetric parts.

EWD

Computes Fourier components of the zonal derivatives (since $\left(\frac{\partial F}{\partial \lambda}\right)_m = im\,(F_m)$) and also Fourier components of the winds from their rotational and divergent parts.

LTD

This subroutine performs direct Legendre transforms, of which there are three types:

a) The basic ones: $F_m \rightarrow F_n^m = \sum_{k=1}^{K} F_m(\mu_k) P_n^m(\mu_k) w(\mu_k)$ (4.3.2.1)

where

k scans over latitude lines,
m is the zonal wave number (refer to Figure 2 on page 12),
n is the meridional index (refer to Figure 2 on page 12),
F_m represents Fourier components,
$P_n^m(\mu_k)$ are values of the Legendre polynomials for the corresponding latitude line,
$w(\mu_k)$ are quadrature weights (Gaussian weights),
$\mu = cos\phi$, the cosine of the latitude.

Equation (4.3.2.1) can also be written

$$F_n^m = \sum_{k=1}^{K/2} \{F_m(\mu_k) P_n^m(\mu_k) w(\mu_k) + F_m(-\mu_k) P_n^m(-\mu_k) w(-\mu_k)\} \quad (4.3.2.2)$$

An important property of the Legendre polynomials is that

$$P_n^m(-\mu) = P_n^m(\mu), \quad \text{if (n+m) even}$$

$$P_n^m(-\mu) = -P_n^m(\mu), \quad \text{if (n+m) odd}$$

We have also $w(-\mu) = w(\mu)$, thus

$$F_n^m = \sum_{K=1}^{k/2} \{F_m(\mu_m) + F_m(-\mu_k)\} P_n^m(\mu_k) w(\mu_k), \text{ if (n+m) even}$$
$$F_n^m = \sum_{k=1}^{K/2} \{F_m(\mu_m) - F_m(-\mu_k)\} P_n^m(\mu_k) w(\mu_k), \text{ if (n+m) odd} \quad (4.3.2.3)$$

Using symmetric $(F_m(\mu) + F_m(-\mu))$ or antisymmetric $(F_m(\mu) - F_m(-\mu))$ parts of the Fourier components in this way results in a significant computational saving compared with direct use of form (4.3.2.1).

b) The meridional derivatives of the Legendre transform:

$$F_m \rightarrow \left(\frac{\partial F}{\partial \mu}\right)_n^m = -\sum_k F_m(\mu_k) A_n^m(\mu_k) w(\mu_k) \quad (4.3.2.4)$$

where

$$A_n^m(\mu_k) = \frac{\partial}{\partial \mu} P_n^m(\mu_k)$$

The negative sign corresponds to an integration by parts.

For these we also make use of symmetry properties i.e.

$$A_n^m(-\mu) = -A_n^m(\mu)\text{, for (n+m) even}$$

$$A_n^m(-\mu) = A_n^m(\mu)\quad\text{, for (n+m) odd}$$

c) Laplacian operator of the Legendre transform:

$$F_m \rightarrow (\nabla^2 F)_n^m = \sum_k F_m(\mu_k)\, R_n^m(\mu_k)\, w(\mu_k)$$

with

$$R_n^m(\mu_k) = -\frac{n(n+1)}{a^2} P_n^m(\mu) \tag{4.3.2.5}$$

using the fact that spherical harmonics are eigenfunctions of the Laplacian operator on the sphere. The $R_n^m(\mu)$ have the same symmetry properties as the $P_n^m(\mu)$.

LTI

This subroutine performs inverse Legendre transforms and makes use of the same symmetric properties as the direct transforms. It computes symmetric and antisymmetric parts of Fourier components for the main fields and their meridional derivatives. Specifically,

$$Q_m \text{ and } \left(\frac{1}{a}\frac{\partial Q}{\partial \mu}\right)_m$$

$$T_m \text{ and } \left(\frac{1}{a}\frac{\partial T}{\partial \mu}\right)_m$$

$$(ln\ p_s)_m \text{ and } \left(\frac{1}{a}\frac{\partial}{\partial \mu} ln\ p_s\right)_m\quad .$$

For vorticity and divergence LTI computes also the rotational and divergent symmetric and antisymmetric parts of the Fourier components of the wind, i.e. for ξ_m:

$$U_{\xi_m} = \sum_n \frac{a(1-\mu)^2}{n(n+1)} \xi_n^m \frac{\partial}{\partial \mu} P_n^m$$

$$iV_{\xi_m} = \sum_n \frac{am}{n(n+1)} \xi_n^m P_n^m$$

and for D_m

$$iU_{D_m} = \sum_n \frac{am}{n\,(n+1)} D_n^m P_n^m$$

$$V_{D_m} = -\sum_n \frac{a\,(1-\mu)^2}{n\,(n+1)} D_n^m \frac{\partial}{\partial\mu} P_n^m$$

These rotational and divergent parts are recombined in SYM2.

SCCx

These subroutines perform computations in spectral space to complete the semi-implicit scheme:

SCCD inverts the Helmholtz equation (cf equation (2.5.43)).

SCCTP computes the implicit contribution of divergence (as computed in SCCD) to the surface pressure and temperature (cf equations (2.5.31) and (2.5.32)).

4.4 SUBROUTINES IN ALPHABETICAL ORDER

The code of the ECHAM3 model includes much more routines, but they are not called.

The list of called routines consists of:

AERO	for the aerosol distibution.
ALLOCA	memory manager routine to allocate array space.
ALLOCB	memory manager routine to allocate buffer array space.
ATMICE	computes seaice cover and depth for uncoupled runs.
AUXHYB	calculates auxiliary variables connected with the vertical finite differencing scheme.
BSSLZR	routine to return zeros of the j0 Bessel function.
BUFF4A	routine to allocate buffers for the symmetric/antisymmetric Fourier-coefficients read in **SCAN1** and written in **SCAN2.**
BUFGRD	input/output routine to allocate gridpoint buffers.
BUFIN	reads a buffer from disk.
BUFNL1	input/output routine to allocate Legendre buffer **L1**
BUFNL2	input/output routine to allocate Legendre buffer **L2**
BUFNL3	input/output routine to allocate Legendre buffer **L3**
BUFOUT	writes a buffer to disk.
CLSST	reads climate sea-surface-temperatures to atmosphere.
CONTEQ	computes temperature and surface pressure increments for the semi-implicit scheme.
CONTROL	main control routine for the model.
COPYBFI	copies a buffer from global array to row array.
COPYBFO	copies a buffer from row array to global array.
COPYI	copies one integer array into another.
COPYRE	copies real values from 1. argument to 2. argument.
COTOEC	organizes the output from the cologne radiation scheme.
CUADJTQ	adjusts T and q due to condensation in cloud ascent.
CUASC	calculates profiles of T, q, l, u and v, corresponding fluxes and precipitation rates during cloud ascent.
CUBASE	calculates cloud base values for cumulus-parameterization.
CUBASMC	calculates cloud base values for midlevel convection.
CUCALL	receives updated tendencies, precipitation and convective cloud parameters for radiation.
CUCCDIA	updates precipitation, cloud base and top for diagnostic scheme of cumulus parameterization.
CUDDRAF	calculates vertical profiles for cumulus downdrafts.
CUDLFS	calculates level of free sinking for cumulus downdrafts.
CUDTDQ	updates precipitation rates and tendencies of T and q.
CUDUDV	updates tendencies of u and v.

CUENTR	calculates en-/detrainment rates for updrafts in cumuli.
CUFLX	final calculation of convective fluxes in the cloud and in the subcloud layer.
CUMASTR	master routine for massflux-scheme.
CUPARAM	defines disposable parameters for massflux-scheme.
DATEFM	calculates initial and forecast data time.
DATIM	returns date and time in ECMWF format.
DYN	computes adiabatic tendencies and auxilliary hybrid.
ECTOCO	organises input to the cologne radiation scheme.
EWD	computes east west derivatives.
EXPAND	expands compressed **G3**-fields.
FCC2	2nd part of the computations in Fourier space.
FFTD	direct Fourier transforms.
FFTI	inverse Fourier transforms.
FLUSS	calculates long-wave radiative flux densities.
FMDDR3	routine to create model grid descriptor record(ddr3).
FORMDDR	routine to format data description records.
FREEALL	memory manager routine to free all managed memory space.
GAUAW	computes abscissas and weights for gaussian integration.
GEOPOT	calculates full- or half-level geopotentials.
GETDDR	routine to read data description records.
GPC	controls grid point computations.
GWDRAG	does the gravity wave parameterisation.
GWPACK	packs directional orographic variances.
GWUNPK	unpacks directional orographic variances.
HARRAY	prints name and value of hollerith array.
HDIFD	horizontal diffusion on divergence.
HDIFQ	horizontal diffusion on humidity.
HDIFT	horizontal diffusion on temperature.
HDIFVO	horizontal diffusion on vorticity.
HDIFX	horizontal diffusion on extra variable.
HELMO	computes matrix needed to invert Helmholtz equation.
HVAR	prints name and value of character variable.
IARRAY	prints name and value of integer array.
INHYSI	initializes constants for the vertical part of the semi-implicit scheme.
INICOM	sets up constants in various common blocks.
INICON	presets constants in **COMCON**.
INICTL	presets constants in **COMCTL**.
INIDIA	presets constants in **COMDIA**.
INIDOC	presets constants in **COMDOC**.
INIFFT	presets constants in **COMFFT**.

INIGAU	presets constants in **COMGAU**.
INIHYB	initializes constants for vertical coordinate calculations.
INILEG	sets up polynomials needed for the Legendre transforms.
INILOC	initializes memory manager tables.
ININMI	presets variables in **COMNMI**.
INIPHY	initializes physical constants of uncertain value.
INISOIL	initiates soil temperatures at five levels.
INISU0	computes initial spectral components for the zonal mean wind used in linearization of vort. and hum. equation.
INITASK	sets up task control.
IOPOSI	routine to position data sets for an initial run.
IOPOSR	routine to position data sets for an resumed run.
IQTASK	retrieves the number of the current task.
IRRAD	organizes computation of long-wave radiative flux densities.
IVAR	print name and value of integer variable.
KUO0	computes T and q tendencies by deep convection (ECHAM1 and 2 only).
LABRUN	labels a forecast run.
LARRAY	prints name and value of logical array.
LEGINV	calculates modified Legendre polynomials for inverse transform using Legendre-polynomials.
LEGMOD	calculates modified Legendre polynomials for direct transform using Legendre-polynomials.
LEGTRI	Legendre functions for a triangular truncation.
LOCATE	memory manager routine to request data.
LTD	direct Legendre transforms needed for all except the mean wind.
LTI	inverse Legendre transforms for all except the mean wind.
LVAR	prints name and value of logical variable.
LWCOND	computes large-scale water phase changes and cloud cover.
MAKEDDR	routine to create data description records.
MAKESD	makes a start data set.
MASTER	main program.
MESAGE	prints 48 characters message on channel nout.
MMERROR	memory manager error reporting routine.
MMGETL	memory manager routine to retrieve the table length.
MMGETN	memory manager routine to retrieve from the name table.
MMLIST	memory manager routine to trace back subroutines.
MMPUTN	memory manager routine to enter into the name table.
MMSWAPN	memory manager routine to swap names in the name table.
MODDDR	routine to modify data description records.
OFFTR	memory manager routine to switch trace off.
ONLOCK	allows code to be protected in a multi-tasking environment.

ONTR	memory manager routine to switch trace on.
OPENR	opens a data set.
OPENSST	opens SST-file.
OPTI	calculates long-wave optical depths.
OPTS	computes optical depths for solar radiation.
ORBIT	compute solar orbital parameters.
OUTGP	controls postprocessing of gridpoint fields (disk version).
OUTGPI	controls postprocessing of gridpoint fields (incore version).
OUTSP	controls postprocessing of spectral fields.
OZONE	compute annual cycle of the ozone distibution.
PACKVEG	packs vegetation parameters.
PGRAD	computes sum of geopotential and pressure gradient terms for semi-implicit scheme.
PHCS	computes the values of the Legendre polynomials and their meridional derivations.
PHYSC	organises calculation of physical tendencies.
POSTATD	completes statistics for dynamics.
POSTATP	completes budgets for physics.
POSTRAD	reallocates space for the matrix for the Helmholtz equation and recompute it after a radiation step.
POSTS1	routine to complete **SCAN1** input/output.
POSTS2	routine to complete **SCAN2** input/output.
PRERAD	presets some constants for radiation before first line of latitude.
PRES	calculates half-level pressures.
PRES1	routine to initialise **SCAN1** input/output.
PRES2	routine to initialise **SCAN2** input/output.
PRESF	computes full-level pressures.
PRESTAT	sets up logical switches and blank statistics at beginning of timestep.
PUSHMEM	memory manager routine to compress managed memory space.
PUTDDR	routine to process data description records.
QMEM	determines how much additional memory can be requested by a job.
QNEGAT	avoids negative values for specific humidity.
RADHEAT	computes temperature changes due to radiation.
RADIAT	calculates long- and short-wave radiative flux densities.
RADINT	organizes the radiation full computations.
RARRAY	prints name and value of real array.
READR	reads a record from disk.
RELBUF	routine to release buffers.
RENAM	memory manager routine to rename data.
REORD	reorders Legendre polynomials.
RESCON	resets some constants to zero at the start of **SCAN1**.

RESETH	resets array hpa(kdim) to hpvalue.
RESETI	resets array ka(kdim) to kvalue.
RESETL	resets array lpa(kdim) to lpvalue.
RESETP	resets parameterisation scheme data.
RESETR	resets real array pa(kdim) to pval.
RESTART	routine to control the restart of an experiment.
RVAR	prints name and value of real variable.
SAVEHIS3	saves history files.
S1FCIO	control routine for **SCAN1** Fourier coefficient input/output.
S1GPIO	control routine for **SCAN1** grid point input/output.
S2IO	control routine for **SCAN2** input/output.
SCALEI	scales integer array by constant.
SCALER	scales real array by constant.
SCAN1	1st scan over the latitude lines.
SCAN2	2nd scan over the latitude lines.
SCCD	calculates final solution of the divergence equation.
SCCTP	adds the implicit contribution of divergence to temperature- and surface-pressure-equation.
SCPAR	computes parameters used for computations in spectral space.
SETBUF	routine to allocate buffers.
SETDDR	routine to process data description records.
SETDIG	presets constants for zonal mean and mask diagnostics.
SETDYN	presets and modify constants in dynamics,initialisation of commonblocks.
SETPHYS	presets constants in physical common blocks.
SETPRES	set surface pressure to a constant value.
SETRAD	presets and modify constants in radiation common blocks.
SETZERO	resets accumulated diagnostic variables (disk version).
SETZEROI	resets accumulated diagnostic variables (incore version).
SI1	1st part of the semi-implicit scheme(done in grid point space).
SI2	2nd part of the semi-implicit scheme(done in Fourier space).
SIGNI	scales real array by constant.
SIGNR	scales real array by constant.
SKINTEM	calculates sea-ice skin temperature.
SOLANG	for solar zenith angle and relative daylength.
SOLRAD	organizes calculation of short-wave radiative flux densities.
START	routine to control the initial start-up of the experiment.
STATD	computes statistics on prognostics variables.
STATP	computes diagnostics of physical variables.
STEPON	controls the time stepping.
SUBJOB	submits a job at the end of a run.

SURF	updates land values of temperature, moisture and snow.
SW2DDR	creates a model switch data description record.
SW4DDR	routine to create model switch for **G4** file.
SYM1	computes symmetric and antisymmetric parts of Fourier components.
SYM2	computes Fourier components from their symmetric-asymmetric parts.
TF12	1st and 2nd part of time filtering.
TTRACE	memory manager backtracing routine.
UNLOC	memory manager routine to release an array.
UNPKVEG	unpacks vegetation parameters.
VARIA	computes temperature profiles, heights of ozone maximum and absorber amounts for radiation calculation.
VDIFF	does the vertical exchange of u,v,T,q by turbulence.
WAITR	waits for completion of an I/O process.
WRITER	writes a record to disk.

4.5 TECHNICAL DETAILS

4.5.1 Internal data flow of the model

Figure 19 is a simplified diagram of the data flow in the model. As an example, the various memory buffers, data transformations and I/O processes are shown for the prognostic variable X (cloud water).

As decribed in section 2.5 two time levels of prognostic variables are required. They are stored in two global buffers. The data for the $t-1$ level are stored as grid point values in the global grid point buffer and the t level is stored as Fourier coefficients in the global Fourier buffer. A third global buffer is provided for Legendre constants. In the in-core version of the model the global buffers are stored in memory and in the disk version they reside on work files.

At the beginning of SCAN1 $t-1$ and t-values are copied from global buffers via S1GPIO and S1FCIO into their respective latitude slice buffers.

Symmetric and antisymmetric Fourier components of cloud water (FSX, FAX) and their meridional derivatives (FSXM, FAXM) of time level t are recombined in SYM2 and are stored in arrays X and DXM respectively. Routine EWD computes Fourier components of the zonal derivatives (DXL). The Fourier components of the main field and their horizontal derivatives are transformed to grid point space via the inverse Fourier transformation FFTI. In DYN the derivatives are used to compute horizontal advection which is part of the adiabatic tendency stored in array XE. In TF12 the first part of the time-filter is applied to the main field (equation (2.5.11)). The result is stored in array XF and is copied to the global grid point buffer via S1GPIO.

Grid point values of time level $t-1$ are stored in array XM1. In TF12 the second part of the time filter is applied (equation (2.5.12)) which gives fully filtered values XM1. XM1 and the adiabatic tendency XE contribute to the diabatic tendency which is calculated in PHYSC. The first part of the semi implicit time integration is performed in SI1 where the accumulated adiabatic and diabatic tendency XE is added. The result is transformed to Fourier space via the direct Fourier transformation FFTD. In the second part of the semi implicit scheme SI2 some implicit contributions are added. In SYM1 the result is split into symmetric and antisymmetric components which are stored in arrays FSX1 and FAX1. The accumulation of spectral coefficients SX is performed via the the direct Legendre transform LTD using Legendre constants from the third global buffer. After the first scan through the latitudes is completed a global set of spectral coefficients is provided in array SX. In spectral space the semi implicit scheme is completed and horizontal diffusion is computed. In the second scan inverse Legendre transforms are performed in routine LTI which computes symmetric and antisymmetric parts of Fourier components for the main field (FSX, FAX) and their meridional derivates (FSXM, FAXM).

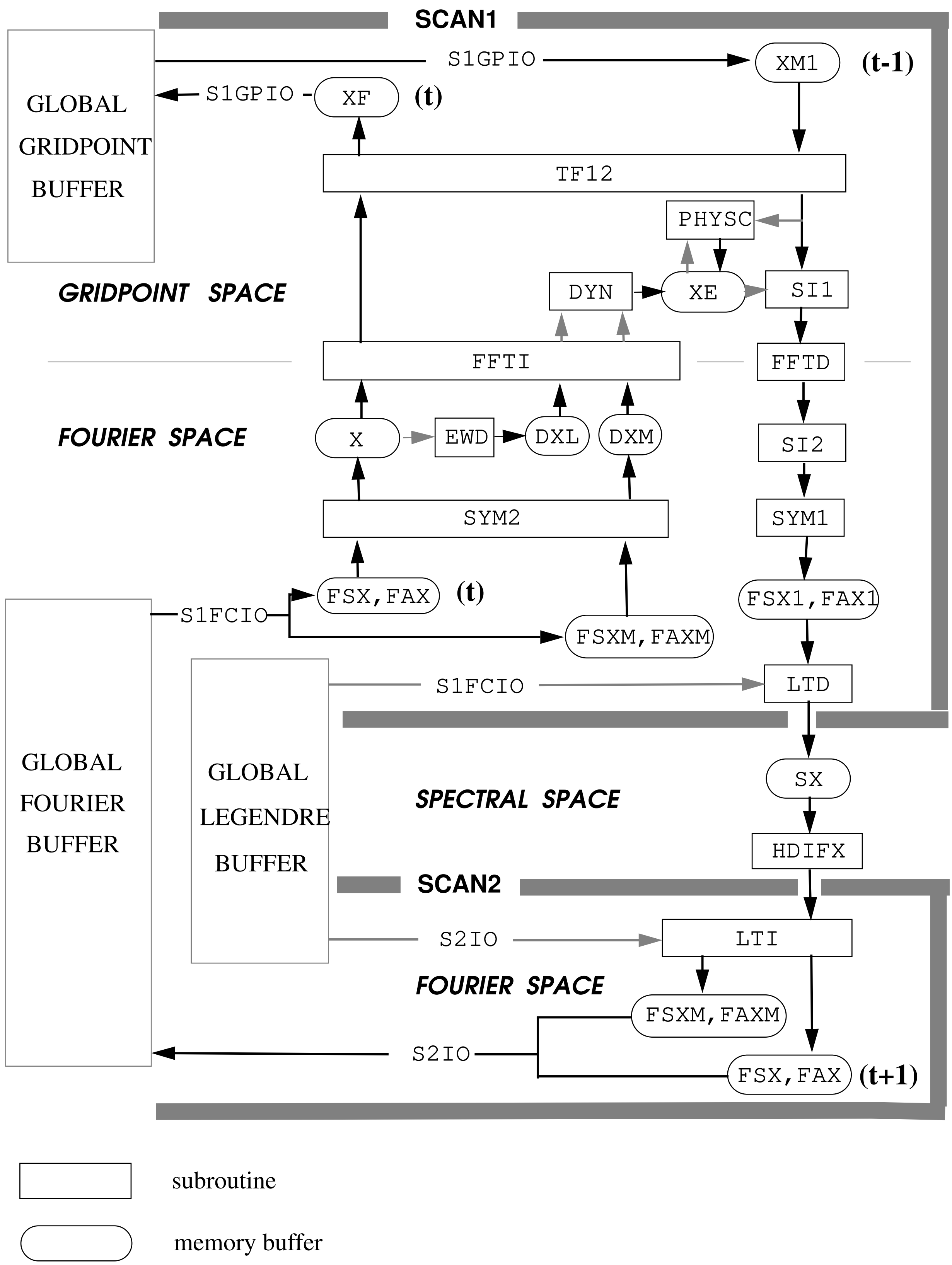


Figure 19 Data flow for the prognostic variable *X* (cloud water) in the ECHAM3 spectral model

4.5.2 Connection between buffers, work- and history-files

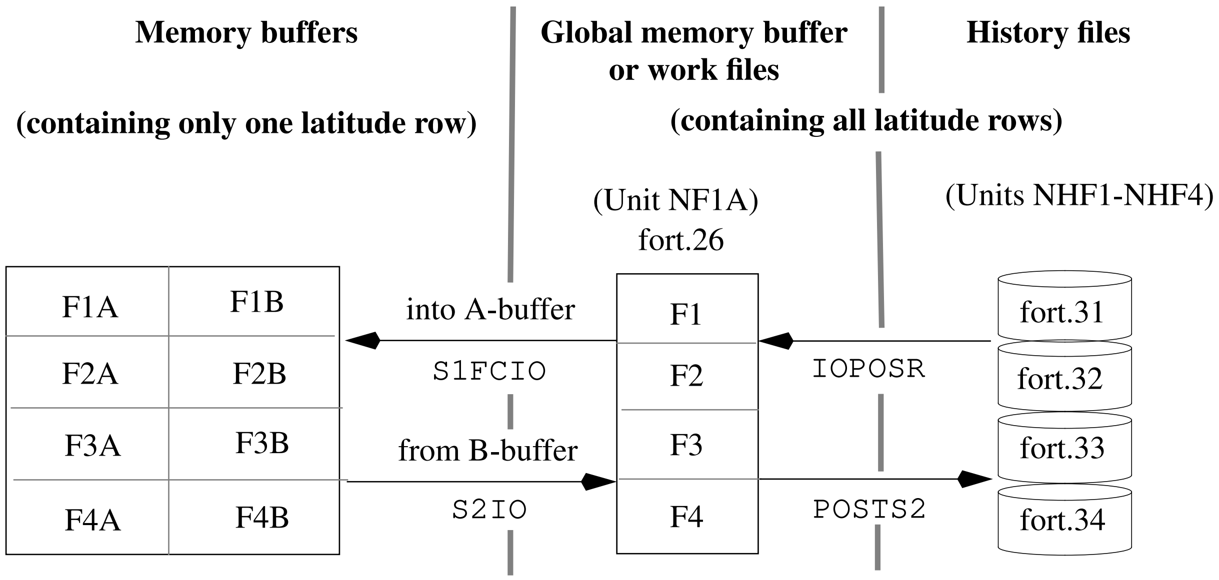


Figure 20 Associated memory buffers, work- and history files for Fourier components.

The F-buffers are allocated in subroutine BUFF4A. FA and FB buffers both have the same memory addresses, because they are not used at the same time. FA buffers are used in the first scan (SCAN1) and contain fields and their meridional derivatives of prognostic variables and wind components at time t, while FB buffers are used in the second scan (SCAN2) for variables at time $t+1$.

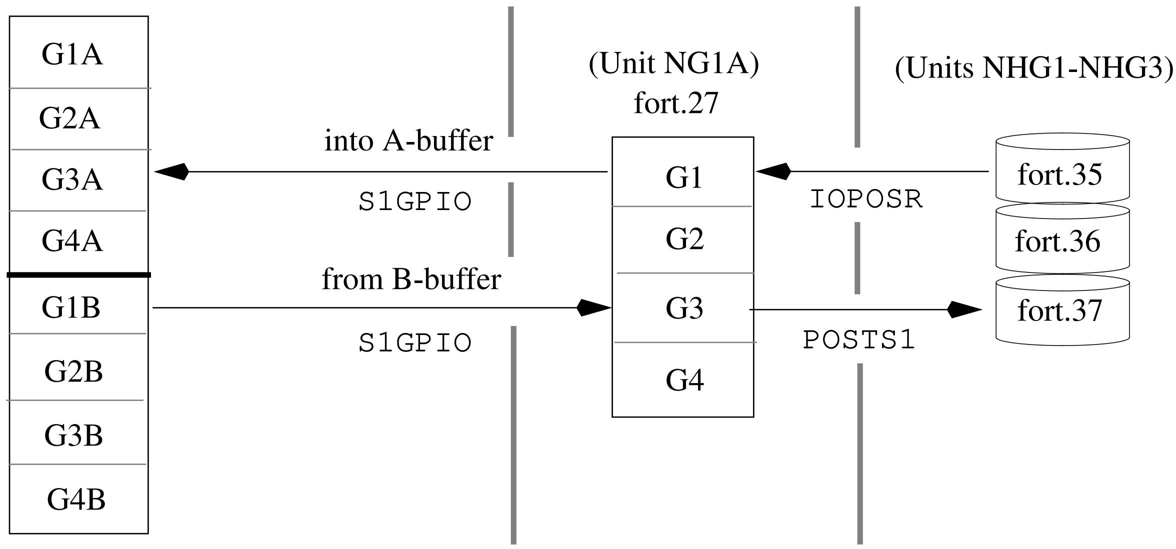


Figure 21 Associated memory buffers, work- and history files for grid point data.

The G-buffers are allocated in subroutine BUFGRD. G1 and G2-buffers contain all prognostic variables and also the wind components at time $t-1$ in grid point space. G3 contains the so called **uninterpolated fields**, e.g. surface fields, which are needed continously. The G4 buffer is never used.

Addresses and lengths of buffers are stored in COMMON /COMBUF/ . For example:

address of buffer F2A	: NPTRB(NF2ABK),
length of buffer G3B	: NLENB(NG3BBK)

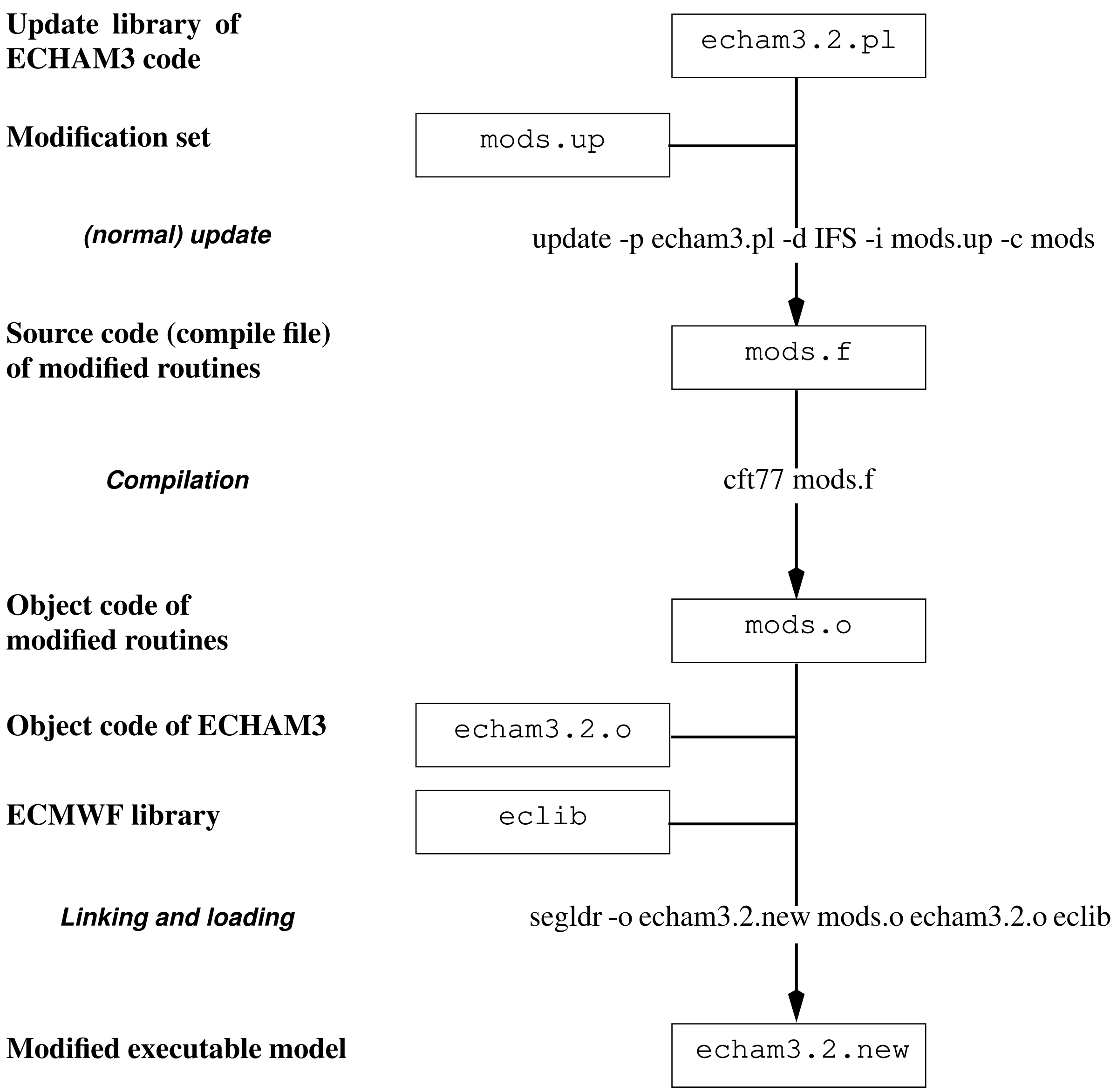
5. USER'S MANUAL

5.1 MODIFICATION OF THE SOURCE CODE

5.1.1 General

Modification of the model source code is done using the CRAY UNICOS UPDATE utility. First of all a modification set has to be created (edited). UPDATE uses the modification set and the update library to build the source code of the modified routines. The source code is compiled creating the object code of the modified routines, which then has to be linked together with the ECHAM3 object code and the ECMWF library to create the modified executable model.

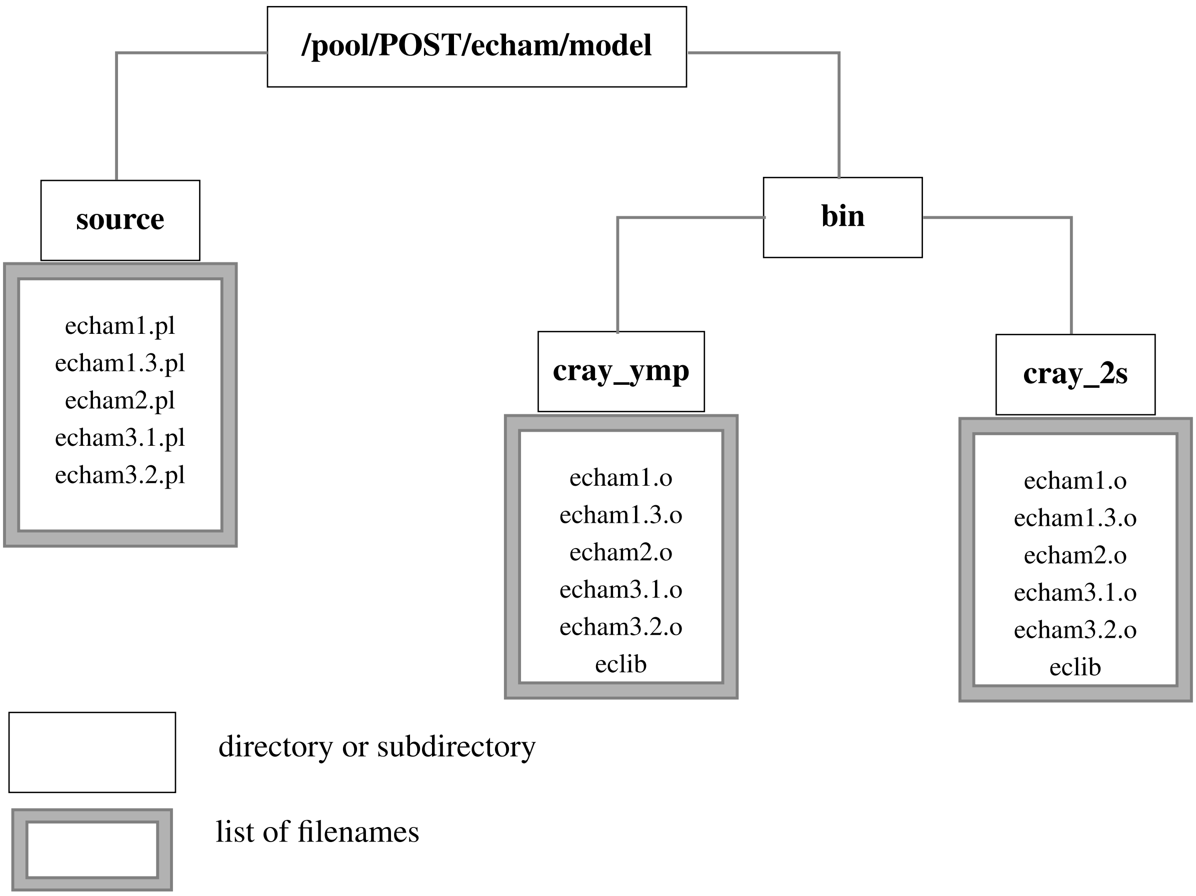
5.1.2 Modification tree



5.1.3 Location of libraries

On the CRAY-2S the ECMWF library (eclib), the object code of ECHAM3 (echam3*o) and the update libraries of several model versions (echam3*pl) reside in global accessible directories. The path names are shown in the diagram following.

Unfortunately disk space on the CRAY-YMP is very low, so a similar structure of global accessible directories could not be installed up to now. Using the model on the YMP requires to transfer the files needed from the CRAY-2S to the CRAY-YMP via FTP.



On the CRAY-YMP, global accessible ECHAM3 libraries reside at the following path's, until the same structure as on the CRAY-2S can be installed there:

/pf/k/k204003/ham3dir/echam3.1.pl
/pf/k/k204003/ham3dir/echam3.1.o
/pf/k/k204003/ham3dir/echam3.2.pl
/pf/k/k204003/ham3dir/echam3.2.o
/pf/k/k204003/eclib

5.1.4 Example script for model modification

```
#QSUB
set -e
# example script for model modifications
LIB=/pool/POST/echam/model
#-----------------------------------------------------------------------
# 1.  update-input-file (=modification set) with modifications for
#     for subroutines VDIFF and SURF
#
cat > mods.up << EOR
*ID  US141289
*/
*/    MODS IN SUBROUTINE VDIFF
*C   VDIFF
*I    VDIFF.123
         ZTP1(JL, JK)=TM1(JL, JK)+ZTIMST*TE(JL, JK)
*/
*/    MODS IN SUBROUTINE SURF
*C   SURF
*D   SURF.456
EOR
#-----------------------------------------------------------------------
# 2.  make source code (=compile file) of modified subroutines
#
update -p $LIB/source/echam3.2.pl -d IFS -i mods.up -c mods
#-----------------------------------------------------------------------
# 3.  compile modified subroutine
#
cft77 mods.f
#-----------------------------------------------------------------------
# 4.  load modified model
#
segldr -o echam3.2.new \
mods.o \
$LIB/bin/cray_2s/echam3.2.o \
$LIB/bin/cray_2s/eclib
#-----------------------------------------------------------------------
exit
```

The output of this job will contain the following messages:

```
*Caution* Duplicate entry point 'SURF' encountered;
     Entry in module 'SURF' from file 'mods.o' is used
     Entry in module 'SURF' from file '/pool/POST/echam/model/bin/cray_2s/echam3.2.o' is ignored
*Caution* Duplicate entry point 'VDIFF' encountered;
     Entry in module 'VDIFF' from file 'mods.o' is used
     Entry in module 'VDIFF' from file '/pool/POST/echam/model/bin/cray_2s/echam3.2.o' is ignored
*Caution* Duplicate entry point 'IC2YMD' encountered;
     Entry in module 'IC2YMD' from file 'mods.o' is used
     Entry in module 'IC2YMD' from file '//pool/POST/echam/model/bin/cray_2s/echam3.2.o' is ignored
*Caution* Duplicate entry point 'CD2DAT' encountered;
     Entry in module 'CD2DAT' from file 'mods.o' is used
     Entry in module 'CD2DAT' from file '/pool/POST/echam/model/bin/cray_2s/echam3.2.o' is ignored
*Caution* Duplicate entry point 'IDAT2C' encountered;
     Entry in module 'IDAT2C' from file 'mods.o' is used
     Entry in module 'IDAT2C' from file '/pool/POST/echam/model/bin/cray_2s/echam3.2.o' is ignored
```

Note that the last three caution messages are default.

The source code of this script can be found at:

/pool/POST/echam/example_scripts/ham3load

5.2 I/O

5.2.1 Diagnostic output

Output is written in GRIB format to a file connected to unit 29. The output interval is controlled by NPTIME in namelist RUNCTL (default: 12 hours).

Prognostic variables at time $t+1$ are written out as spherical harmonics by subroutine OUTSP. They are taken from the spectral buffers containing SD, SVO, STP, SQ and SX (see Appendix E, Table 9). The content of the spectral output is controlled by namelist POSTCTL (see section 5.3.2). SX is not written out by default.

Grid point variables from the **G3B** buffer are written out by subroutines OUTGPI for the in-core version and OUTGP for the disk version respectively. The content of the output is controlled by the code table of the input file stdin (see Appendix F, Table 10).

For details refer to section 5.3(example job).

5.2.2 Newstart

Two initial files are needed:
A file connected to **unit 23** (NISP) containing prognostic variables as spectral coefficients.
A file connected to **unit 24** (NIGP) containing some fields used in the **G3** buffer.

Spectral coefficients are read from **unit 23** by routine IOPOSI. During a fictive time step NSTEP = -1 they are transferred into fourier coefficients (buffer **F1**-**F4**) by an inverse legendre transform (done by subroutine LTI in SCAN2), which are then written to the global memory buffer containing Fourier coefficients or to the workfile connected to **unit 26** (done by subroutine S2IO in SCAN2) latitude row by latitude row.

Grid point fields are read from **unit 24** via subroutine IOPOSI. Fields, that are needed to fill up the **G3** buffer, but not read from **unit 24** (mostly accumulated diagnostic fields), are preset by subroutine RESETP (usually set to 0.). The whole **G3** buffer is then written to the global memory buffer containing grid point data or to the workfile connected to **unit 27**.

G1 and **G2** buffers normally contain forecast variables in grid point space at time $(t-1)$, which are not yet available, because the model is starting to run. $(t-1)$ -values are obtained by copying the actual values (time t) into the **G1** and **G2** buffers in the first scan (subroutine SCAN1) at timestep NSTEP = 0. This first timestep is only a forward timestep and not a leap-frog-step, because of the equality of t- and $(t-1)$ -values.

Actual values (time t) are the spectral coefficients read and transformed to Fourier coefficients in SCAN2 during time step NSTEP = -1. In SCAN1 they are transformed to grid point values via subroutine FFTI (inverse Fourier transform).

5.2.3 Rerun

History files of a previous run are read into the connected memory buffers and are copied to the respective global memory buffers or workfiles (done by subroutine IOPOSR).

At the end of the run the same is done vice versa by subroutines POSTS1 and POSTS2.

5.2.4 Input of sea surface temperature

The sea-surface temperature (SST) is read each time step from the file connected to **unit 20** by subroutine CLSST.

5.2.5 Files used for a model run

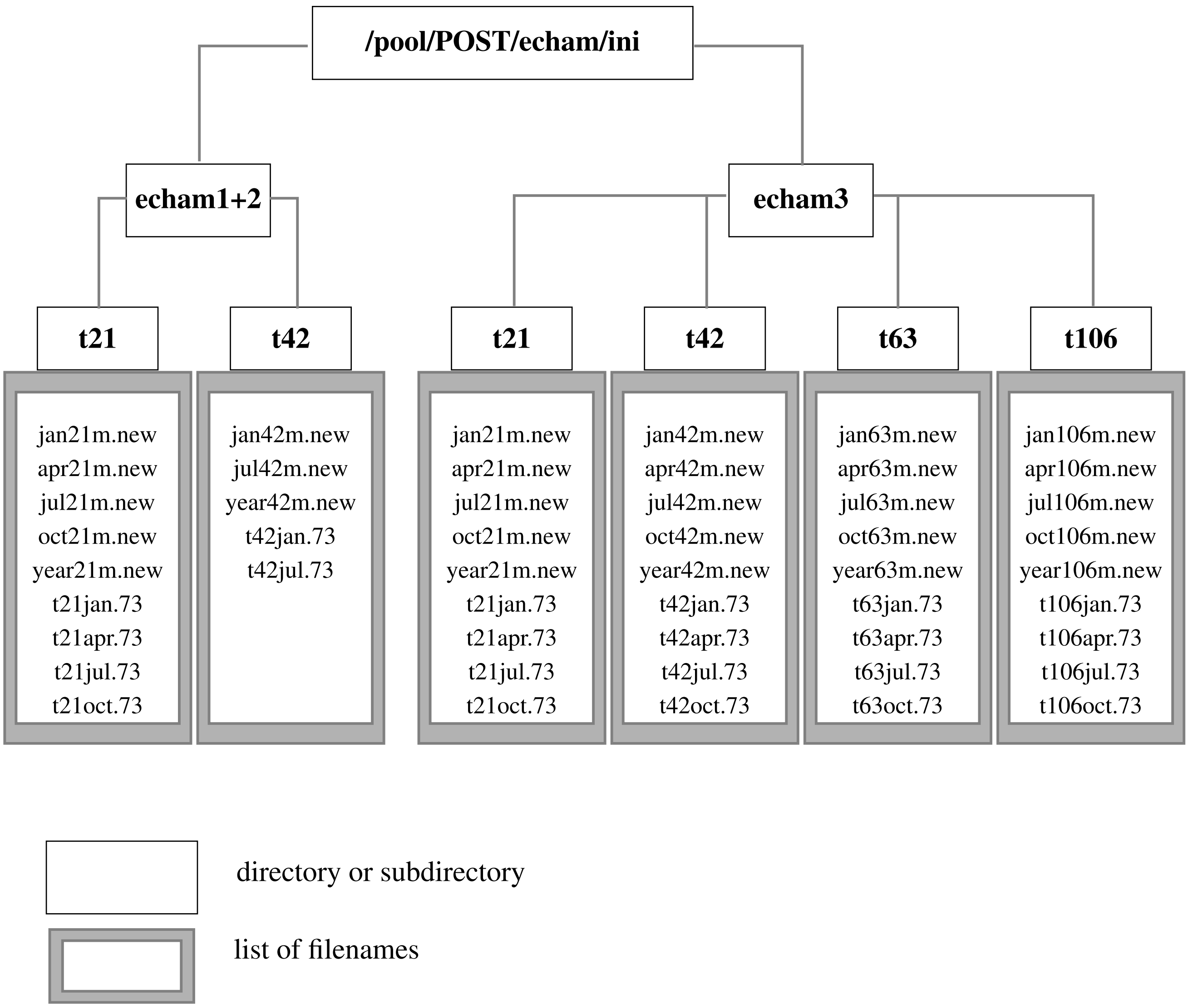
Table 5 Files used for a model run

file type	file name	unit name	unit no.	input routine	output routine	file content
INITIAL	tNNXXX.73[a]	NISP	23	IOPOSI	-	spectr. atmospheric field (prognostic variables)
	XXXNNm.new[b]	NIGP	24	IOPOSI	-	gridpoint surface fields
	yearNNm.new[c]	-	20	CLSST INISOIL	-	12 month climatology of surface temperature
CONTROL	stdin	NIN	5	INIccc[d]	-	namelists, code table
WORK	l1a	NL1A	25	S1FCIO	S1FCIO	Legendre constants
	f1a	NF1A	26	S1FCIO	S1FCIO	Fourier coefficients
	g1a	NG1A	27	S1GPIO	S1GPIO	gridpoint fields
HISTORY	${EXP}.31[e]	NHF1	31	IOPOSR	POSTS2	Fourier F1 buffer
	${EXP}.32[e]	NHF2	32	IOPOSR	POSTS2	Fourier F2 buffer
	${EXP}.33[e]	NHF3	33	IOPOSR	POSTS2	Fourier F3 buffer
	${EXP}.34[e]	NHF4	34	IOPOSR	POSTS2	Fourier F4 buffer
	${EXP}.35[e]	NHG1	35	IOPOSR	POSTS1	gridpoint G1 buffer
	${EXP}.36[e]	NHG2	36	IOPOSR	POSTS1	gridpoint G2 buffer
	${EXP}.37[e]	NHG3	37	IOPOSR	POSTS1	gridpoint G3 buffer
OUTPUT	${EXP}_yymm[f]	NUNITSP	29	-	OUTSP	spectral fields on hybrid levels
		NUNITUN	29	-	OUTGP OUTGPI	gridpoint fields uninterpolated fields
	stdout	NOUT	6	-	various	quick diagnostics, parameters, switches
${EXPNO}HDIF_yymm[g]		-	4	-	HDIFD	diagnostics of horizontal diffusion

a. NN : Resolution (21, 42, 63 or 106) XXX : Month (jan, apr, jul, oct) Example : t21jan.73
b. NN : Resolution (21, 42, 63 or 106) XXX : Month (jan, apr, jul, oct) Example : jul42m.new
c. NN : Resolution (21, 42, 63 or 106) Example : year42m.new
d. ccc : Prefix of COMMON-Blocks (CON, CTL, DIA, FFT, etc.) Example : INICTL for COMCTL
e. EXP : User-defined filename-prefix (Shell-variable)
f. EXP : same as in [e] yy : Year (01, 02,...) mm : Month (01, 02, ...,12)
g. EXPNO: first five characters of EXP yy : Year (01, 02,...) mm : Month (01, 02, ...,12)

5.2.6 Location of initial files

As the libraries, the initial files reside in global accessible directories on the CRAY-2S. The initial files are divided into categories. The two main categories distinguish the model versions (echam1/2 and echam3). Further subdivision is neccessary to distinguish the initial files for different horizontal resolutions (T21/T42/T63/T106). For path's refer to the diagram following.



5.3 RUNNING THE MODEL

5.3.1 Example script for running the T21-model (Bourne shell)

```
#QSUB -lT 2500
#QSUB -lM 3 Mw
#QSUB -eo
#QSUB
#
cd $MFHOME
set -e
#################
#set variables: #
#                                                 #
EXP=01120ctrl                                     # <-- experiment identifier
#                                                 #
MODEL=echam3                                      # <-- name of executable model
#                                                 #
RERUN=F                                           # <-- rerun switch
#                                                 #     (set TRUE after initial
#                                                 #      run is submitted to
#                                                 #      enable rerun chain !)
DPATH=$MFHOME/                                    # <-- output file directory
#                                                 # ###########
INI=/pool/POST/echam/ini/echam3/t21               # <-- paths to initial files
#################
#
# specification of file structures (activate/deactivate respective blocks)
#
################# CRAY-2 ########################
assign -s pure       -a $INI/t21jan.73       u:23 # <-- atmosphere initial file
assign -s pure       -a $INI/jan21m.new      u:24 # <-- surface initial file
assign -s pure       -a $INI/year21m.new     u:20 # <-- sst climate file
#                                                 #
assign -s pure       -a $TMPDIR/l1a          u:25 # <
assign -s pure       -a $TMPDIR/f1a          u:26 # <-- work files (scratch)
assign -s pure       -a $TMPDIR/g1a          u:27 # <
#                                                 #
assign -s pure                               u:29 # <-- output file (GRIB format)
#                                                 #     filename: ${EXP}_yymm
assign -s pure       -a $DPATH${EXP}.31      u:31 # <
assign -s pure       -a $DPATH${EXP}.32      u:32 # <
assign -s pure       -a $DPATH${EXP}.33      u:33 # <
assign -s pure       -a $DPATH${EXP}.34      u:34 # <-- rerun files
assign -s pure       -a $DPATH${EXP}.35      u:35 # <
assign -s pure       -a $DPATH${EXP}.36      u:36 # <
assign -s pure       -a $DPATH${EXP}.37      u:37 # <
#
################# CRAY-YMP #####################
#assign -s unblocked -a $INI/t21jan.73       u:23 # <-- atmosphere initial file
#assign -s unblocked -a $INI/jan21m.new      u:24 # <-- surface initial file
#assign -s unblocked -a $INI/year21m.new     u:20 # <-- sst climate file
#                                                 #
#assign -s u         -a $TMPDIR/l1a          u:25 # <
#assign -s u         -a $TMPDIR/f1a          u:26 # <-- work files (scratch)
#assign -s u         -a $TMPDIR/g1a          u:27 # <
#                                                 #
#assign -s unblocked                         u:29 # <-- output file (GRIB format)
#                                                 #     filename: ${EXP}_yymm
#assign -s unblocked -a $DPATH${EXP}.31      u:31 # <
#assign -s unblocked -a $DPATH${EXP}.32      u:32 # <
#assign -s unblocked -a $DPATH${EXP}.33      u:33 # <
#assign -s unblocked -a $DPATH${EXP}.34      u:34 # <-- rerun files
#assign -s unblocked -a $DPATH${EXP}.35      u:35 # <
#assign -s unblocked -a $DPATH${EXP}.36      u:36 # <
#assign -s unblocked -a $DPATH${EXP}.37      u:37 # <
```

```
#######################################################
#
# stdin file: control parameter namelists, output path, experiment name
# and code table as "here document"
#
cat > namelist << EOR
 &SDSCTL
   LRES=$RERUN
 &END
 &RUNCTL
   NSTOP=20,
 &END
 &DYNCTL
 &END
 &NMICTL &END
 &PHYSCTL
 &END
 &RADCTL
 &END
 &POSTCTL
 &END
$DPATH
$EXP
 129 GEOSP
 134 APS
 139 TS
 140 WS
 141 SN
 142 APRL
 143 APRC
 144 APRS
 145 VDIS
 146 AHFS
 147 AHFL
 159 USTAR3
 160 RUNOFF
 163 ACLCV
 164 ACLCOV
 165 U10
 166 V10
 167 TEMP2
 168 DEW2
 169 TSURF
 170 TD
 171 WIND10
 172 SLM
 173 AZ0
 175 ALBEDO
 176 SRADS
 177 TRADS
 178 SRAD0
 179 TRAD0
 180 USTR
 181 VSTR
 182 EVAP
 183 TDCL
 194 WLM1
 201 T2MAX
 202 T2MIN
 203 SRAD0U
 204 SRADSU
 205 TRADSU
 206 TSN
 207 TD3
 208 TD4
 209 TD5
 210 SEAICE
 214 TSMAX
```

```
 215 TSMIN
 216 WIMAX
 217 TOPMAX
 218 SNMEL
 220 TSLIN
 221 DSNAC
 222 ALWCAC
 223 ACLCAC
 224 SRAD0F
 225 TRAD0F
 226 ACCNTF
 227 SRAD0C
 228 TRAD0C
 229 ACCNTC
 230 QVI
 231 ALWCVI
 999 END
EOR
#
#
ja                                                 # <-- start job accounting
#
$MFHOME/$MODEL < namelist                          # <-- execute model
#
ja -lsc                                            # <-- print job accounting
ja -t                                              # <-- terminate job accounting
#
qsub ham3run                                       # <-- submit next job
#                                                  #     (rerun chain)
#
chmod a+r $DPATH${EXP}_*                           # <-- make output file public
```

The source code of this script can be found in:

/pool/POST/echam/example_scripts/ham3run

5.3.2 Selectable model parameters

SDSCTL - information necessary to start or resume the experiment

Name	Type	Purpose	Default
LRES	LOGICAL	Rerun switch: TRUE: Restart from rerun files (fort.31...37) FALSE: Start from initial files (fort.23, fort.24)	TRUE

RUNCTL - run control variables

Name	Type	Purpose	Default
NSTOP	INTEGER	Time step to stop the experiment NSTOP > 0 : stop after NSTOP timesteps NSTOP < 0 : stop after -NSTOP days	10 timesteps
NPTIME	INTEGER	Frequency of post processing (= output data write out) NPTIME > 0 : every NPTIME timesteps NPTIME < 0 : every -NPTIME days	12 hours
NWTIME	INTEGER	Frequency of rerun files write-up (causes interrupt of model run) NWTIME > 0 : every NWTIME timesteps NWTIME < 0 : every -NWTIME days	30 days
LWMONTH	LOGICAL	Interrupt a run at the end of a month including write-up of rerun files (set to FALSE if the default value is not used)	TRUE
NWLAG	INTEGER	Frequency for saving rerun files in months (inhibits overwriting)	13 (i.e. no saving)
LABORT	LOGICAL	True for exit at end of the experiment (timestep=NSTOP) to stop rerun chain	TRUE
DTIME	REAL	Time step in seconds	T21: 2400. T42: 1440. T63: 900. T106: 720.
NSPACE1	INTEGER	Size of space managed by memory manager	auto-adjustment
LCORE	LOGICAL	True for incore version of model which reduces i/o overhead	TRUE
NG3XP	INTEGER	Number of extra G3-fields (G3Xnn)	0
NG3XL	INTEGER ARRAY	Number of levels of extra G3-fields	99*1
LXACCU	LOGICAL ARRAY	Switches for accumulation (= mean over postprocessing interval) of extra G3-fields	99*F
NXPBITS	INTEGER	Number of GRIB code bits for extra G3-fields	99*16

DYNCTL - control options for the dynamics and general control

Name	Type	Purpose	T21	T42	T63	T106	Default
NDIADFR	INTEGER	Frequency of diagnostic printout NDIADFR > 0 : every NDIADFR time step NDIADFR = 0 : no diagnosticnostics NDIADFR < 0 : every -NDIADFR hour					5 days
NDIAVFR	INTEGER	Frequency of dynamical diagnostics for each level NDIAVFR > 0 : every NDIAVFR time step NDIAVFR = 0 : no diagnostic NDIAVFR < 0 : every -NDIAVFR hour					0 (no diagn.)
LZONDIA	LOGICAL	Zonal diagnostics in NDIADFR interval					FALSE
LUMAX	LOGICAL	Print information on maximum wind in NDIADFR interval					FALSE
APSURF	REAL	Global mean surface pressure [Pa] (depends on orography)					98200.
DIFVO	REAL	Coefficient for horizontal diffusion of vorticity	1.0 E7	11.E16	8.5E6	5.E6	
DIFD	REAL	Same for divergence	5.0 E7	55. E16	42.5E6	25.E6	
DIFT	REAL	Same for temperature	0.2 E7	4.4 E16	1.7E6	1.E6	
DIFQ	REAL	Same for specific humidity	0.2 E7	4.4 E16	1.7E6	1.E6	
DIFX	REAL	Same for liquid water	0.2 E7	4.4 E16	1.7E6	1.E6	
LDIAHDF	LOGICAL	Diagnostics of horizontal diffusion					FALSE

PHYSCTL - control options for diabatic processes

Name	Type	Purpose	Default
LPHYS	LOGICAL	True for parameterization of diabatic processes	TRUE
LVDIFF	LOGICAL	True for turbulent vertical diffusion	TRUE
LMFPEN	LOGICAL	True for cumulus convection (mass flux)	TRUE
LKUO0	LOGICAL	True for zero order Kuo convection	TRUE
LSCV	LOGICAL	True for shallow convection	TRUE
LGWDRAG	LOGICAL	True for gravity wave drag scheme	TRUE T21: FALSE
LCOND	LOGICAL	True for large scale condensation scheme	TRUE
LSURF	LOGICAL	True for soil model	TRUE
LQNEGAT	LOGICAL	True for correction of negative specific humidity	FALSE
NDIAPFR	INTEGER	Frequency of physics budget diagnostics NDIAPFR > 0 : every NDIAPFR timesteps NDIAPFR < 0 : every -NDIAPFR hours	10 days

RADCTL - control options for the radiation scheme

Name	Type	Purpose	Default
LRAD	LOGICAL	True for radiation	TRUE
LDIUR	LOGICAL	True for diurnal cycle	TRUE
NMONTH	INTEGER	Month for perpetual month experiments NMONTH = 1...12 : number of month (Jan...Dec) (**choose appropiate initial data !**) NMONTH = 0 : annual cycle	0
NRADFR	INTEGER	Frequency of full radiation computations NRADFR > 0 : every NRADFR timesteps NRADFR < 0 : every -NRADFR hours	2 hours
NRADPFR	INTEGER	Print frequency for radiation statistics in number of radiation steps	126
NRADPLA	INTEGER	Print radiation statistics every NRADPLA latitude line	10
CO2FAC	REAL	Factor for CO_2 concentration	ECHAM1/2: 1. (330 ppm) ECHAM3: 1.045 (345 ppm)

POSTCTL- control of spectral variables write out

Name	Type	Purpose	Default
LPPSPE	LOGICAL	True for write out of all spectral variables except liquid water SX	TRUE
LPPD	LOGICAL	True for write out of divergence	TRUE
LPPVO	LOGICAL	Same for vorticity	TRUE
LPPT	LOGICAL	Same for temperature	TRUE
LPPP	LOGICAL	Same for surface pressure	TRUE
LPPQ	LOGICAL	Same for specific humidity	TRUE
LPPX	LOGICAL	Same for liquid water content	FALSE

Note: Write out of grid point variables is controlled by the code table as shown in the example job in section 5.3.1 (for definition of codes and code numbers refer to Appendix F, Table 10).

5.3.3 Output examples

to be made by anyone with procedure
/pool/POST/t21.post

5.4 STRUCTURE OF DATASETS

All files except **stdin** and **stdout** are pure data files, i.e. no control words or record terminators are used. BUFFER IN/OUT is used for I/O and SETPOS for file positioning. Use the following assign commands for specification of file structure:

CRAY-2:

```
assign -s pure        -a $INI/t21jan.73          u:23
assign -s pure        -a $INI/jan21m.new         u:24
assign -s pure        -a $INI/year21m.new        u:20
assign -s pure        -a $TMPDIR/l1a             u:25
assign -s pure        -a $TMPDIR/f1a             u:26
assign -s pure        -a $TMPDIR/g1a             u:27
assign -s pure                                   u:29
assign -s pure        -a $DPATH${EXP}.31         u:31
assign -s pure        -a $DPATH${EXP}.32         u:32
assign -s pure        -a $DPATH${EXP}.33         u:33
assign -s pure        -a $DPATH${EXP}.34         u:34
assign -s pure        -a $DPATH${EXP}.35         u:35
assign -s pure        -a $DPATH${EXP}.36         u:36
assign -s pure        -a $DPATH${EXP}.37         u:37
```

CRAY-YMP:

```
assign -s unblocked   -a $INI/t21jan.73          u:23
assign -s unblocked   -a $INI/jan21m.new         u:24
assign -s unblocked   -a $INI/year21m.new        u:20
assign -s u           -a $TMPDIR/l1a             u:25
assign -s u           -a $TMPDIR/f1a             u:26
assign -s u           -a $TMPDIR/g1a             u:27
assign -s unblocked                              u:29
assign -s unblocked   -a $DPATH${EXP}.31         u:31
assign -s unblocked   -a $DPATH${EXP}.32         u:32
assign -s unblocked   -a $DPATH${EXP}.33         u:33
assign -s unblocked   -a $DPATH${EXP}.34         u:34
assign -s unblocked   -a $DPATH${EXP}.35         u:35
assign -s unblocked   -a $DPATH${EXP}.36         u:36
assign -s unblocked   -a $DPATH${EXP}.37         u:37
```

The data written to the output file (unit 29, name: ${EXP}_yymm) is in GRIB format.
For processing these data a variety of utility routines are available:

"***Afterburner***"	**mod21**, **mod42**, **mod63**, and **mod106**
"***Tape accessing scripts***"	**extract** and **extrakt**
"***grib moduls***"	

Documentation on this postprocessing routines can be found in

Description of programs handling files in WMO GRIB and EXTRA format, DKRZ Report, to be published;

The output files produced on CRAY-2 can be processed on CRAY-YMP and vice versa. For details see corresponding documentation.

As with the pure file structure, “private” utilities reading the output data must be aware of how the file was written in order to read it properly. For reading in a FORTRAN program use the assign command to specify the file structure:

```
assign -s pure “filename”          on CRAY-2S
assign -s unblocked “filename”     on CRAY-YMP
```

6. REFERENCES

Alexander, R.C. and Mobley, R.L., 1976:
Monthly average sea-surface temperatures and ice-pack limits on a global grid;
Mon. Wea. Rev., 104, 143-148.

Asselin, R., 1972:
Frequency filter for time integrations;
Mon. Wea. Rev., 100, 487-490.

Baede, A.P.M., M. Jarraud and U. Cubasch, 1979:
Adiabatic formulation and organization of ECMWF's spectral model;
ECMWF Tech. Rep., 15, 40 pp.

Bauer, H., E. Heise, J. Pfaendtner and V. Renner, 1985:
Development of an economical soil model for climate simulation.
In: A. Ghazi and R. Fantechi (Eds.)
Current Issues in Climate Research, Proceedings of EC Climatology Programme Symposium, Sophia Antipolis, France, 2-5 Oct. 1984, D. Reidel, Dordrecht, 219-226.

Bennetts, A., and B.J. Hoskins, 1979:
Conditional symmetric instability - a possible explanation for frontal rainbands;
Quart. J. Roy. Meteor. Soc., 105, 945-962.

Bennetts, A., and J.C. Sharp, 1982:
The relevance of conditional symmetric instability to the prediction of meso-scale frontal rainbands;
Quart. J. Roy. Meteor. Soc., 108, 595-602.

Betts, A.K., 1973:
Non-precipitating cumulus convection and its parameterization;
Quart. J. Roy. Meteor. Soc., 99, 178-196.

Blackadar, A.K., 1962:
The vertical distribution of wind and turbulent exchange in a neutral atmosphere;
J. Geophys. Res., 67, 3095-3102.

Blondin, C., 1989:
Research on land surface parameterization schemes at ECMWF;
Proceedings of the workshop on "Parameterization of fluxes over land surface", ECMWF, Reading, UK.

Blondin, C. and H. Böttger, 1987:
The surface and sub-surface parameterisation scheme in the ECMWF forecasting system. Revision and operational assessment of weather elements;
ECMWF Tech. Memo., 135, 48 pp.

Boer, G.J., N.A. McFarlane, R. Laprise, J.D. Henderson, and J.-P. Blanchet, 1984:
The Canadian Climate Centre spectral atmospheric general circulation model;
Atmos.-Ocean, 22, 397-429.

Bourke, W., 1972:
An efficient one level primitive-equation spectral model;
Mon. Wea. Rev., 100, 683-689.

Bourke, W., 1974:
A multi-level spectral model: Formulation and hemispheric integrations;
Mon. Wea. Rev., 102, 688-701.

Brinkop, S., 1991:
Inclusion of cloud processes in the ECHAM PBL parameterization;
In: R. Sausen (Ed.)
Studying Climate with the ECHAM Atmospheric Model.
Large Scale Atmospheric Modelling, Report No. 9, 5-14,
Meteorologisches Institut der Universität Hamburg.

Brinkop, S., 1992:
Parameterisierung von Grenzschichtwolken für Zirkulationsmodelle;
Berichte aus dem Zentrum für Meeres- und Klimaforschung, Reihe A: Meteorologie,
Nr. 2, Meteorologisches Institut der Universität Hamburg, 77 pp.

Browning, K.A., M.E. Hardman, T.W. Harrold and C.W. Pardoc, 1973:
The structure of rainbands within a mid-latitude depression;
Quart. J. Roy. Meteor. Soc., 99, 215-231.

Burridge, D.M. and J. Haseler, 1977:
A model for medium range weather forecasting - Adiabatic formulation;
ECMWF Tech. Rep. No 4, 46 pp.

Cess, R.D. and G.L. Potter, 1987:
Explanatory studies of cloud radiative forcing with a General Circulation Model;
Tellus, 39A, 460-473.

Cess, R.D. and G.L. Potter, 1988:
A methodology for understanding and intercomparing atmospheric climate
feedback processes in General Circulation Models;
J. Geophys. Res., 93, 8305-8314.

Charnock, M., 1955:
Wind stress on a water surface;
Quart. J. Roy. Meteor. Soc., 81, 639-640.

Cheng, L., T.-C. Yip and H.-R. Cho, 1980:
Determination of mean cumulus cloud vorticity from GATE A/B-scale potential
vorticity budget;
J. Atm. Sci., 37, 797-811.

Dorman, J.L. and P.J. Sellers, 1989:
A global climatology of albedo, roughness length and stomatal resistence for atmospheric general circulation models as represented by the Simple Biosphere Model (SiB);
J. Appl. Meteor., 28, 833-855.

Dufour, L. and J. van Mieghem, 1975:
Thermodynamic de l'Atmosphere;
Institut Royal Météorologique de Belgique, 278 pp.

Dümenil, L. and E. Todini, 1992:
GCM parameterization of soil hydrology using a catchment scheme;
in preparation.

Eickerling, H., 1989:
Parameterisierung des infraroten Strahlungstransports für Kohlendioxid, Wasserdampf und Ozon in einem breitbandigen Strahlungstransportmodell;
Diplomarbeit, Institut für Meteorologie und Geophysik, Univ. Köln, FRG.

Eliasen, E., B. Machenhauer and E. Rasmussen, 1970:
On a numerical method for integration of the hydrodynamical equations with a spectral representation of the horizontal fields;
Inst. of Theor. Met., Univ. of Copenhagen, No. 2.

Ellison, T.H., 1957:
Turbulent transfer of heat and momentum from an infinite rough plane;
J. Fluid. Mech., 2, 456-466.

Emanuel, K.A., 1982:
Inertial instability and mesoscale convective systems,
Part II: Symmetric CISK in a baroclinic flow;
J. Atm. Sci., 39, 1080-1097.

Foster, D.S., 1958:
Thunderstorm gusts compared with computed downdraft speeds;
Mon. Wea. Rev., 86, 91-94.

Fritsch, J. M. and C. G. Chapell, 1980:
Numerical prediction of convectively driven mesoscale pressure systems,
Part I: Convective parameterization;
J. Atm. Sci., 37, 1722-1733.

Geleyn, J.F. and H.J. Preuss, 1983:
A new data set of satellite-derived surface albedo values for operational use at ECMWF;
Arch. Meteor. Geophys. Bioclim., Ser. A, 32, 353-359.

Girard, C. and M. Jarraud, 1982:
Short and medium range forecast differences between a spectral and a grid-point model: An extensive quasi-operational comparison;
ECMWF Tech. Rep. No. 32, 178pp.

Hense, A., M. Kerschgens and E. Raschke, 1982:
An economical method for computing radiative transfer in circulation models;
Quart. J. Roy. Meteor. Soc., 108, 231-252.

Herzegh, P.H. and P.V. Hobbs, 1980:
The mesoscale and microscale structure and organization of clouds and precipitation in mid-latitude cyclones, Part II: Warm frontal clouds;
J. Atm. Sci., 37, 597-611.

Heymsfield, A.J., 1977:
Precipitation development in stratiform ice clouds: A microphysical and dynamical study;
J. Atm. Sci., 34, 367-381.

Hoskins, B.J. and Simmons, A.J., 1975:
A multi-layer spectral model and the semi-implicit method;
Quart. J. Roy. Meteor. Soc., 101, 637-655.

Houze, R.A., J.D. Locatelli and P.V. Hobbs, 1976:
Dynamics and cloud microphysics of the rainbands in an occluded frontal system;
J. Atm. Sci., 35, 1921-1936.

ICAO, 1964:
Manual of the ICAO Standard Atmosphere;
U.S. Government Printing Office, Washington, D.C.

Jarraud, M., C. Girard, and U. Cubasch, 1981:
Comparison of medium-range forecasts made with the models using spectral or finite-difference techniques in the horizontal;
ECMWF Tech. Rep. No. 23, 96 pp.

Jarraud, M., C. Girard, and J.-F. Geleyn, 1982:
Note on a possible linearization of the vorticity equation in a primitive equations spectral model;
Research Activities in Atmospheric and Oceanic Modelling, Report No. 3, pp 4.2-4.4

Johnson, R.H., 1976:
The role of convective-scale precipitation downdrafts in cumulus and synoptic scale interactions;
J. Atm. Sci., 33, 1890-1910.

Johnson, R.H., 1980:
Diagnosis of convective and mesoscale motions during Pase III of GATE;
J. Atm. Sci., 37, 733-753.

Kasahara, A., 1974:
Various vertical coordinate systems used for numerical weather prediction;
Mon. Wea. Rev., 102, 509-522.

Kerschgens, M., U. Pilz, E. Raschke, 1978:
A modified two-stream approximation for computations of the solar radiation budget in a cloudy atmosphere;
Tellus, 30, 429-435.

Kessler, E., 1969:
On the distribution and continuity of water substance in atmospheric circulation;
Meteorological Monographs, 10, Americ. Meteor. Soc., Boston, MA.

Krylov, V.I., 1962:
Approximate calculation of integrals;
Macmillan, New York, 357 pp.

Kukla, G. and D. Robinson, 1980:
Annual cycle of surface albedo;
Mon. Wea. Rev., 108, 56-58.

Kuo, H. L., 1965:
On formation and intensification of tropical cyclones through latent heat release by cumulus convection;
J. Atm. Sci., 22, 40-63.

Kuo, H. L., 1974:
Further studies of the parameterization of the influence of cumulus convection on large-scale flow;
J. Atm. Sci., 31, 1232-1240.

Laursen. L. and E. Eliasen, 1989:
On the effects of the damping mechanisms in an atmospheric general circulation model;
Tellus, 41A, 385-400.

Le Mone, M. A. and W. T. Pennell, 1976:
The relationship of trade wind cumulus distribution to subcloud layer fluxes and structure;
Mon. Wea. Rev., 104, 524-539.

Lindzen, R. S., 1981:
Some remarks on cumulus parameterization;
Rep. on NASA-GISS Workshop "Clouds in Climate": Modelling and Satellite Observational Studies, 42-51.

Louis, J. F., 1979:
A parametric model of vertical eddy fluxes in the atmosphere;
Boundary Layer Meteorology, 17, 187-202.

Louis, J. F., Tiedtke, M., Geleyn, J.-F., 1982:
A short history of the PBL parameterisation at ECMWF;
Proceedings, ECMWF workshop on planetary boundary layer parameterization, Reading, 25-27 Nov. 81, 59-80.

London, J., Bojkov, R. D., Oltmans, S., Kelley, J. I., 1976:
Atlas of the global distribution of total ozone, July 1957 - June 1967;
NCAR Technical Note, 113+STR.

Lowe, P.R., 1977:
An approximating polynomial for the computation of saturation vapor pressure;
J. Appl. Meteor., 16, 100-103.

Machenhauer, B. and E. Rasmussen, 1972:
On the integration of the spectral hydrodynamical equations by a transform method;
Inst. of Theor. Met., Univ. of Kopenhagen, Report No. 4.

Manabe, S., 1969:
Climate and the ocean circulation
1. The atmospheric circulation and the hydrology of the earth's surface;
Mon. Wea. Rev., 97, 739-774.

Mason, B.J., 1971:
The physics of clouds
Clarendon Press, Oxford, 671 pp.

Matthews, E., 1983:
Global vegetation and land use: New high-resolution data bases for climate studies;
J. Clim. Appl. Meteor., 22, 474-487.

Matveev, L. T., 1984:
Cloud dynamics;
Atm. Sci. Library, D. Reidel Publishing Company, Dordrecht, 340 pp.

Miller, M. J., T. N. Palmer and R. Swinbank, 1989:
Parameterization and influence of sub-grid scale orography in general circulation and numerical weather prediction models;
Met. Atm. Phys., 40, 84-109.

Miller, M. J., A. Beljaars and T. N. Palmer, 1992:
The sensitivity of the ECMWF model to the parameterization of evaporation from the tropical oceans;
J. Clim., in press.

Nitta, T., 1975:
Observational determination of cloud mass flux distributions;
J. Atm. Sci., 32, 73-91.

Orszag, S.A., 1970:
Transform method for calculation of vector coupled sums;
J. Atm. Sci., 27, 890-895.

Palmer, T. N., G. J. Shutts and R. Swinbank, 1986:
Alleviation of a systematic westerly bias in general circulation and numerical weather prediction models through an orographic gravity wave drag parameterization;
Quart. J. Roy. Meteor. Soc., 112, 1001-1031.

Paltridge, G. W. and C. M. R. Platt, 1976:
Radiative processes in Meteorology and Climatology;
Developments in Atmospheric Science,Vol. 5, Elsevier Scientific Publ. Co., Amsterdam etc., 318 pp.

Pandolfo, J.P., 1967:
Wind and temperature profiles for constant-flux boundary layer in lapse rate conditions with a variable eddy conductivity to eddy viscosity ratio;
J. Atm. Sci., 23, 495-502.

Peltier, W.R. and T.L. Clark, 1987:
Nonlinear mountain waves and wave-mean flow interaction: elements of a drag parameterization;
ECMWF Seminar/Workshop on Observation, Theory and Modelling of Orographic Effects, ECMWF, Reading, 15-20 Sep. 1986, Vol 1, 223-249.

Phillips, N.A., 1957:
A coordinate system having some special advantages for numerical forecasting;
J. Met., 14, 184-185.

Reynolds, R. W., 1988:
A real-time global sea surface temperature analysis;
J. of Climate, 1, 75-86.

Robert, A. J., 1981:
A stable numerical integration scheme for the primitive meteorological equations;
Atmos. Ocean, 19, 35-46.

Robert, A. J., 1982:
A semi-Lagrangian and semi-implicit numerical integration scheme for the primitive meteorological equations;
J. Met. Soc. Japan, 60, 319-325.

Robert, A. J., J. Henderson and C. Turnbull, 1972:
An implicit time integration scheme for baroclinic models in the atmosphere;
Mon. Wea. Rev., 100, 329-335.

Robock, A., 1980:
The seasonal cycle of snow cover, sea-ice and surface albedo;
Mon. Wea. Rev., 108, 267-285.

Rockel, B., E. Raschke and B. Weyres, 1991:
A parameterization of broad band radiative transfer properties of water, ice and mixed clouds;
Beitr. Phys. Atmosph., 64, 1-12.

Roeckner, E. and U. Schlese, 1985:
January simulation of clouds with a prognostic cloud cover scheme;
ECMWF Workshop on "Cloud cover in numerical models", 26-28 Nov. 1984, 87-108, ECMWF, Reading, UK.

Roeckner, E., M. Rieland and E. Keup, 1991:
Modelling of cloud and radiation in the ECHAM model;
ECMWF/WCRP Workshop on "clouds, radiative transfer and the hydrological cycle", 12-15 Nov. 1990, 199-222, ECMWF, Reading, UK.

Roeckner, E., K. Arpe, L. Bengtsson, S. Brinkop, L. Dümenil, M. Esch, E. Kirk, F. Lunkeit, M. Ponater, B. Rockel, R. Sausen, U. Schlese, S. Schubert and M. Windelband, 1992:
Simulation of the present-day climate with the ECHAM model:
Impact of model physics and resolution;
Max-Planck-Institut für Meteorologie, Report No. 93.

Sasamori, T., 1975:
A statistical model for stationary atmospheric cloudiness, liquid water content and rate of precipitation;
Mon. Wea. Rev., 103, 1037-1049.

Schneider, E.K. and R.S. Lindzen, 1976:
A discussion of the parameterization of momentum exchange of cumulus convection;
J. Geophys. Res., 81, 3158-3160.

Sellers, P.J., Y. Mintz, Y.C. Sud and A. Dalcher, 1986:
A simple biosphere model (Sib) for use within general circulation models;
J. Atm. Sci., 43, 505-531.

Shettle, E.P. and R. Fenn, 1976:
Models of the atmospheric aerosols and their optical properties;
AGARD Conference Proceedings, No. 183, AGARD-CP-183.

Simmons, A.J. and D.M. Burridge, 1981:
An energy and angular-momentum conserving vertical finite difference scheme and hybrid vertical coordinates;
Mon. Wea. Rev., 109, 758-766.

Simmons, A.J. and R. Strüfing, 1981:
An energy and angular-momentum conservinq finite difference scheme, hybrid coordinates and medium-range weather prediction;
ECMWF Tech. Rep. No. 28, 68 pp.

Simmons, A.J., B.J. Hoskins and D.M. Burridge, 1978:
Stability of the semi-implicit time scheme;
Mon. Wea. Rev., 106, 405-412.

Simmons, A.J. and J. Chen, 1991:
The calculation of geopotential and pressure-gradient in the ECMWF atmospheric model: Influence on the simulation of the polar atmosphere and on temperature analyses;
Quart. J. Roy. Meteor. Soc., 117, 29-58.

Simpson, J., 1971:
On cumulus entrainment and one-dimensional models;
J. Atm. Sci., 28, 449-455.

Simpson, J. and V. Wiggert, 1969:
Models of precipitating cumulus towers;
Mon. Wea. Rev., 97, 471-489.

Smith, R.N.B., 1990:
A scheme for predicting layer clouds and their water content in a general circulation model;
Quart. J. Roy. Meteor. Soc., 116, 435-460.

Stephens, G.L., 1978:
Radiation profiles in extended water clouds: 2. parameterization schemes;
J. Atm. Sci., 35, 2123-2132.

Sundquist, H., 1978:
A parameterization scheme for non-convective condensation including prediction of cloud water content;
Quart. J. Roy. Meteor. Soc., 104, 677-690.

Temperton, C., 1983:
Fast mixed-radix real Fourier transforms;
ECMWF Tech. Memo., 71, 18 pp.

Tiedtke, M., 1989:
A comprehensive mass flux scheme for cumulus parameterization in large-scale models;
Mon. Wea. Rev., 117, 1779-1800.

Tiedtke, M., W.A. Heckley, and J. Slingo, 1988:
Tropical forecasting at ECMWF: On the influence of physical parameterization on the mean structure of forecasts and analyses;
Quart. J. Roy. Meteor. Soc., 114, 639-664.

Warrilow, D.A., A.B. Sangster and A. Slingo, 1986:
Modelling of land surface processes and their influence on European climate;
Meteorological Office, Met O 20 Tech. Note DCTN 38, Bracknell, U.K.

Wexler, R., and D. Atlas, 1959:
Precipitation generating cells;
J. Meteor., 16, *327-332.*

Wilcox, R. W. and A. D. Belmont, 1977:
Ozone concentration by latitude, altitude and month, near 80^{o}W;
Report, Control Data Corporation, Contract No. DOT-FA77WA-3999.

Wilson, M.F. and A. Henderson-Sellers, 1985:
A global archive of land cover and soil data sets for use in general circulation climate models;
J. Climat., 5, 119-143.

Xu, K.M., and S.K. Krueger, 1991:
Evaluation of cloudiness parameterizations using a cumulus ensemble model;
Mon. Wea. Rev., 119, 342-367.

Yanai, M., S. Esbensen and J.-H. Chu, 1973:
Determination of bulk properties of tropical cloud clusters from large-scale heat and moisture budgets;
J. Atm. Sci., 30, 611-627.

Zdunkowski, W.G., R.M. Welch, G. Korb, 1980:
An investigation of the structure of typical two-stream methods for the calculation of solar fluxes and heating rates in clouds;
Beitr. Phys. Atm., 53, 147-166.

Appendix A THE UNPARAMETRIZED EQUATIONS

A.1 INTRODUCTION

To derive the governing equations given by (2.2.1) - (2.2.7) and (2.2.12) - (2.2.15), we take start from the unparametrized equations for a mixture of dry air, water vapour, liquid water and ice, and work for convenience in a Cartesian coordinate system. An individual component is denoted by a subscript "k", where $k = d, v, l$ or i for dry air, water vapour, liquid water or ice, respectively. The specific mass of component k, denoted by q_k, is defined by

$$q_k = \frac{m_k}{m} = \frac{\rho_k}{\rho} \tag{A.1.1}$$

where

m_k	is the mass of component k in a small material volume moving with the local velocity of the atmosphere,
$m = \sum m_k$	is the total mass of the material volume,
ρ_k	is the density of component k, and
$\rho = \sum \rho_k$	is the density of the atmosphere.

The rate of change of m_k is denoted by $\dot{m}_k$. This change occurs because of

(a) internal phase changes

(b) rainfall, snowfall, and surface exchanges.

The rate of change due to (a) alone is denoted by $\dot{m}_{ki}$, and that due to (b) by $\dot{m}_{ke}$. Then

$$\dot{m}_k = \dot{m}_{ki} + \dot{m}_{ke} \tag{A.1.2}$$

$$\dot{m}_{di} = \dot{m}_{de} = 0 \tag{A.1.3}$$

$$\sum_k \dot{m}_{ki} = 0 \tag{A.1.4}$$

The rate of change of total mass is given by

$$\dot{m} = \sum_k \dot{m}_k = \sum_k \dot{m}_{ke} \tag{A.1.5}$$

The rate of change of density of component k satisfies the equation

$$\dot{\rho}_k = \frac{\rho}{m}\dot{m}_k \tag{A.1.6}$$

provided (as is reasonable) volume changes due to precipitation or phase changes are neglected. The net rate of change of density, $\dot{\rho}$, is then given by

$$\dot{\rho} = \frac{\rho}{m}\sum_k \dot{m}_k = \frac{\rho}{m}\dot{m} \tag{A.1.7}$$

A.2 THE ADVECTIVE FORM OF THE UNPARAMETRIZED EQUATIONS

A.2.1 The material derivative

The material derivative is denoted by $\frac{d}{dt}$. Its definition is

$$\frac{d}{dt} \equiv \frac{\partial}{\partial t} + \underset{\sim}{v} \cdot \nabla \qquad \text{(A.2.1.1)}$$

where $\underset{\sim}{v}$ here denotes the three-dimensional velocity vector, and ∇ the ususal three-dimensional vector operator. Horizontal vectors and operators will subsequently be denoted by a subscript "h" .

A.2.2 The equation of state

We consider a volume V of atmosphere, of which dry air and water vapour occupy a volume V_{d+v}. The equations of state for dry air and water vapour are

$$p_d V_{d+v} = m_d R_d T \qquad \text{(A.2.2.1)}$$

and

$$p_v V_{d+v} = m_v R_v T \qquad \text{(A.2.2.2)}$$

where p_d and p_v are partial pressures. Dalton´s Law then shows that the total pressure p is given from (A.2.2.2) by

$$p = \frac{m_d R_d T + m_v R_v T}{V_{d+v}} \qquad \text{(A.2.2.3)}$$

Introducing the specific volumes of liquid water v_l, and ice v_i,

$$\begin{aligned} V_{d+v} &= V - m_l v_l - m_i v_i \\ &= \frac{m}{\rho}\left(1 - \rho\left(q_l v_l + q_i v_i\right)\right) \end{aligned} \qquad \text{(A.2.2.4)}$$

and (A.2.2.3) becomes

$$p = \rho T \frac{R_d q_d + R_v q_v}{1 - \rho\left(q_l v_l + q_i v_i\right)} \qquad \text{(A.2.2.5)}$$

or

$$p = \rho T R_d \frac{1 + \left(\frac{1}{\varepsilon} - 1\right) q_v - q_l - q_i}{1 - \rho\left(q_l v_l + q_i v_i\right)} \qquad \text{(A.2.2.6)}$$

where

$$\varepsilon = R_d / R_v \qquad \text{(A.2.2.7)}$$

A.2.3 Mass conservation

Conservation of mass for element k leads to the equation

$$\frac{d\rho_k}{dt} + \rho_k (\nabla \cdot \underset{\sim}{v}) = \dot{\rho}_k = \frac{\rho \dot{m}_k}{m} \tag{A.2.3.1}$$

Summing over k then gives

$$\frac{d\rho}{dt} + \rho (\nabla \cdot \underset{\sim}{v}) = \frac{\rho \dot{m}}{m} = \dot{\rho} \tag{A.2.3.2}$$

In addition, by definition

$$\frac{dm_k}{dt} = \dot{m}_k \tag{A.2.3.3}$$

which gives

$$\frac{dq_k}{dt} = \frac{\dot{m}_k}{m} - \frac{m_k \dot{m}}{m^2} = \frac{1}{m} (\dot{m}_k - q_k \dot{m}) \tag{A.2.3.4}$$

A.2.4 The velocity equation

The advective form of the equations for the horizontal components of velocity is unaltered by mass changes. The horizontal velocity components thus satisfy the equation

$$\frac{d\underset{\sim}{v}_h}{dt} = -\frac{1}{\rho} \nabla_h p - 2 (\underset{\sim}{\Omega} \times \underset{\sim}{v}_h)_h \tag{A.2.4.1}$$

where $\underset{\sim}{\Omega}$ is the earth's rotation vector. Changes due to molecular stresses are neglected.

A.2.5 The thermodynamic equation

As discussed by Dufour and Van Mieghem (1975, Eq. 5.21), the first law of thermodynamics may be written

$$\delta Q + \alpha dp = d_i H = d_i \left(\sum m_k h_k\right) \tag{A.2.5.1}$$

where the h_k are specific enthalpies, $\alpha = 1/\rho$ is the specific volume and the subscript i denotes changes independent of the mass changes due to precipitation. As molecular diffusion is neglected, δQ represents the heat received by the atmospheric element due to radiation and to heat exchange with falling rain or snow.

Under the usual assumptions of perfect gas behaviour for dry air and water vapour, and neglect of variations of the specific enthalpies of water and ice with pressure, we can write

$$h_k = h_k^0 + C_{pk} T \tag{A.2.5.2}$$

and (A.2.5.1) becomes

$$m C_p dT = \alpha dp + \delta Q - \sum_k h_k d_i m_k \tag{A.2.5.3}$$

where

$$C_p = \sum_k C_{pk} q_k \tag{A.2.5.4}$$

Thus considering a material volume of the atmosphere, we obtain the thermodynamic equation

$$C_p \frac{dT}{dt} = \frac{1}{\rho}\frac{dp}{dt} + Q_R + Q_M - \sum_k h_k \frac{\dot{m}_{ki}}{m} \tag{A.2.5.5}$$

where Q_R and Q_M are the heating rates due to respectively radiation and the heat transferred from falling rain or snow.

A.3 THE FLUX FORMS OF THE EQUATIONS

It is convenient to define the differential operator $\frac{D}{Dt}$ by

$$\frac{DX}{Dt} = \frac{dX}{dt} + X(\nabla \cdot \underset{\sim}{v}) = \frac{\partial X}{\partial t} + \nabla \cdot (X\underset{\sim}{v}) \qquad \text{(A.3.1)}$$

Note that

$$\rho = \frac{dx}{dt} = \frac{D\rho x}{Dt} \ \text{ if } \ \dot{\rho} = 0 \quad . \qquad \text{(A.3.2)}$$

Equations (A.2.3.4), (A.2.4.1) and (A.2.5.5) may then be written

$$\frac{D\rho}{Dt} = \frac{\rho}{m}\dot{m} = \dot{\rho} \qquad \text{(A.3.3)}$$

$$\frac{D\rho q_k}{Dt} = \frac{\rho}{m}\dot{m}_k = \dot{\rho}_k \qquad \text{(A.3.4)}$$

$$\frac{D\rho \underset{\sim}{v}_h}{Dt} = \dot{\rho}\underset{\sim}{v}_h - \nabla_h p - 2\rho(\underset{\sim}{\Omega} \times \underset{\sim}{v}_h)_h \qquad \text{(A.3.5)}$$

$$C_p\frac{D\rho T}{Dt} = C_p\dot{\rho}T + \frac{dp}{dt} + \rho(Q_R + Q_M) - \rho\sum_k h_k\frac{\dot{m}_{ki}}{m} \qquad \text{(A.3.6)}$$

From the definition (A.2.5.4) of C_p we obtain

$$\frac{DC_p\rho T}{Dt} = C_p\frac{D\rho T}{Dt} + \rho T\frac{d}{dt}\sum_k C_{pk}q_k \qquad \text{(A.3.7)}$$

and using (A.2.5.4) and (A.3.6) gives

$$\begin{aligned}\frac{DC_p\rho T}{Dt} &= C_p\dot{\rho}T + \frac{dp}{dt} + \rho(Q_R + Q_M) - \rho\sum_k (h_k^0 + C_{pk}T)\frac{\dot{m}_{ki}}{m} \\ &\quad + \rho T\sum_k C_{pk}\left(\frac{\dot{m}_k}{m} - \frac{q_k\dot{m}}{m}\right)\end{aligned} \qquad \text{(A.3.8)}$$

Using (A.1.2), (A.1.7) and (A.2.5.4), we obtain from (A.3.8):

$$\frac{DC_p\rho T}{Dt} = \frac{dp}{dt} + \rho(Q_R + Q_M) - \rho\sum_k h_k^0\frac{\dot{m}_{ki}}{m} + \rho T\sum_k C_{pk}\frac{\dot{m}_{ke}}{m} \qquad \text{(A.3.9)}$$

A.4 THE INTRODUCTION OF DIFFUSIVE FLUXES

We now introduce a separation of dependent variables into components that will be explicitly resolved in the model and components the effect of which will require parametrization.

If the operator $(\overline{\ })$ represents an average over unresolved scales in space and time, then we write:

$$X = \overline{X} + X' \quad \text{with} \quad \overline{X'} = 0$$

and

$$X = \overline{\overline{X}} + X'' \quad \text{with} \quad \overline{\overline{X''}} = 0 \quad ,$$

where

$$\overline{\overline{X}} = \frac{\overline{\rho X}}{\overline{\rho}} \quad \text{is a mass weighted average.}$$

It follows that

$$\frac{\overline{\overline{D}}\,\overline{\rho}\,\overline{\overline{X}}}{Dt} = \frac{\overline{D\rho X}}{Dt} - (\nabla \cdot \overline{\rho \underset{\sim}{v}'' X''})$$

$$\frac{\overline{\overline{d}}\,\overline{X}}{dt} = \frac{\overline{dX}}{dt} - (\overline{\underset{\sim}{v}'' \cdot \nabla X})$$

$$\overline{\rho} XY = \overline{\rho XY} = \overline{\rho}\,\overline{\overline{X}}\,\overline{\overline{Y}} + \overline{\rho X'' Y''}$$

Using these results, equations (A.3.2) - (A.3.4) and (A.3.8) become

$$\frac{\overline{\overline{D}}\,\overline{\rho}}{Dt} = \overline{\dot{\rho}} = \overline{\rho}\left(\frac{\overline{\overline{\dot{m}}}}{m}\right) \tag{A.4.1}$$

$$\frac{\overline{\overline{D}}\,\overline{\rho}\,\overline{\overline{q}}_k}{Dt} = \overline{\dot{\rho}_k} - (\nabla \cdot \overline{\rho \underset{\sim}{v}'' q''_k}) = \overline{\rho}\left(\frac{\overline{\overline{\dot{m}_k}}}{m}\right) - (\nabla \cdot \overline{\rho \underset{\sim}{v}'' q''_k}) \tag{A.4.2}$$

$$\begin{aligned} \frac{\overline{\overline{D}}\,\overline{\rho}\,\overline{\overline{\underset{\sim}{v}}}_h}{Dt} &= \overline{\dot{\rho}\underset{\sim}{v}_h} - \nabla_h \overline{p} - 2\overline{\rho}\,(\underset{\sim}{\Omega} \times \overline{\overline{\underset{\sim}{v}}}_h)_h - (\nabla \cdot \overline{\rho \underset{\sim}{v}'' \underset{\sim}{v}''_h}) \\ &= \overline{\rho}\left(\frac{\overline{\overline{\dot{m}}}}{m}\right)\overline{\overline{\underset{\sim}{v}}}_h - \nabla_h \overline{p} - 2\overline{\rho}\,(\underset{\sim}{\Omega} \times \overline{\overline{\underset{\sim}{v}}}_h)_h - (\nabla \cdot \overline{\rho \underset{\sim}{v}'' \underset{\sim}{v}''_h}) - \overline{\rho\,(\frac{\dot{m}}{m})''\,\underset{\sim}{v}''_h} \end{aligned} \tag{A.4.3}$$

and

$$\begin{aligned} \frac{\overline{\overline{D}}}{Dt}(\overline{\rho}\,\overline{\overline{C}}_p\overline{\overline{T}} + \overline{\rho C''_p T''}) = {} & \frac{\overline{\overline{d}}\,\overline{p}}{dt} + \overline{\rho}\,(\overline{\overline{Q}}_R + \overline{\overline{Q}}_M) - \overline{\rho}\sum_k h_k^0 \left(\frac{\overline{\overline{\dot{m}_{ki}}}}{m}\right) \\ & + \overline{\rho}\,\overline{\overline{T}}\sum_k C_{pk}\left(\frac{\overline{\overline{\dot{m}_{ke}}}}{m}\right) + \overline{\underset{\sim}{v}'' \cdot \nabla p} - (\nabla \cdot \overline{\rho \underset{\sim}{v}'' C_p T''}) \\ & + \sum_k C_{pk}\overline{\rho T''\,(\frac{\dot{m}_{ke}}{m})''} \end{aligned} \tag{A.4.4}$$

The equation of state (A.2.2.5) gives

$$p = \rho R T \tag{A.4.5}$$

where $R = (R_d q_d + R_v q_v) / \{1 - \rho\,(q_l v_l + q_i v_i)\}$

whence

$$\bar{p} = \overline{\rho R T} = \bar{\rho}\,\overline{\overline{R}}\,\overline{\overline{T}} + \overline{\rho R'' T''} \tag{A.4.6}$$

Using $\overline{\overline{C_p}} = \sum C_{pk}\overline{\overline{q_k}}$, (A.4.2) and (A.4.4) may be written

$$\begin{aligned}\overline{\overline{C_p}}\frac{\overline{\overline{D}}\bar{\rho}\overline{\overline{T}}}{Dt} = {} & \frac{\overline{\overline{d}}\bar{p}}{dt} + \bar{\rho}\,(\overline{\overline{Q_R}} + \overline{\overline{Q_M}}) - \bar{\rho}\sum_k \overline{\overline{h_k}}\left(\overline{\overline{\frac{\dot{m}_{ki}}{m}}}\right) + \bar{\rho}\,\overline{\overline{C_p}}\,\overline{\overline{T}}\left(\overline{\overline{\frac{\dot{m}}{m}}}\right)\\ & + \overline{\underset{\sim}{v}''\nabla p} - \nabla\cdot\overline{\rho\underset{\sim}{v}''\,(C_p T)''} + \overline{\overline{T}}\sum_k C_{pk}\nabla\cdot\overline{\rho\underset{\sim}{v}''q''_k}\\ & - \frac{\overline{\overline{D}}}{Dt}(\overline{\rho C''_p T''}) + \sum_k C_{pk}\overline{\rho T''\,(\frac{\dot{m}_{ke}}{m})}\end{aligned} \tag{A.4.7}$$

A.5 APPROXIMATIONS AND DEFINITIONS

At this stage, we make two approximations. The first is to neglect the higher-order correlations

$$\overline{\rho T''\,(\frac{\dot{m}_{ke}}{m})''},\ \frac{\overline{\overline{D}}}{Dt}(\overline{\rho C_p'' T''}),\ \overline{\rho T'' R''}\ \text{and}\ \overline{\rho\,(\frac{\dot{m}}{m})''\underset{\sim}{v}''_h}\ .$$

This is equivalent to assuming higher-order terms are important only when eddy velocities and derivatives are involved. The second is to neglect the term $\rho\,(q_l \cdot v_l + q_i \cdot v_i)$ in the equation of state, or equivalently to neglect the volume occupied by liquid water and ice compared with that occupied by dry air and water vapour.

In addition we introduce the following notation:

a) The vertical flux of a variable X, $\overline{\rho w'' X''}$, is denoted by J_X. Here w is the vertical velocity component.

b) The term $\overline{v''\cdot\nabla p}$ is added to the term $\frac{\partial}{\partial z}\overline{\rho w''\,(C_p T)''}$ and the resulting sum is expressed as the derivative $\frac{\partial J_S}{\partial z}$ of the vertical flux of dry static energy, plus a term which is written $\bar{\rho}\overline{\overline{Q_D}}$ and regarded as representing unorgnized transfers between enthalpy and sub-grid scale kinetic energy. The latter is parametrized by the heating implied by the dissipation of kinetic energy due to the parametrized vertical momentum fluxes $J_{\underset{\sim}{v}_h}$.

c) The net effect of horizontal fluxes is represented only by their contribution K_X to the tendency of variable X.

d) The term $-\bar{\rho}\sum_k \overline{\overline{h}}_k \left(\overline{\overline{\frac{\dot{m}_{ki}}{m}}}\right)$ representing the latent heat release associated with internal phase changes is written $\bar{\rho}\overline{\overline{Q_L}}$.

A.6 RETURN TO THE ADVECTIVE FORM

With the above approximations and definitions, we obtain from the equations of Appendix A.4, on dropping the $(\bar{\ })$ and $(\bar{\bar{\ }})$ signs,

$$\frac{d\rho}{dt} + \rho\nabla \cdot \underset{\sim}{v} = \rho\frac{\dot{m}}{m} \tag{A.6.1}$$

$$\frac{dq_k}{dt} = S_{q_k} - \frac{1}{\rho}\frac{\partial J_{q_k}}{\partial z} + K_{q_k} \tag{A.6.2}$$

$$\frac{d\underset{\sim}{v}_h}{dt} = -\frac{1}{\rho}\nabla_h p - 2\,(\underset{\sim}{\Omega}\times\underset{\sim}{v}_h)_h - \frac{1}{\rho}\frac{\partial J_{\underset{\sim}{v}_h}}{\partial z} + K_{\underset{\sim}{v}_h} \tag{A.6.3}$$

$$\frac{dT}{dt} = \frac{1}{\rho C_p}\frac{dp}{dt} + \frac{1}{C_p}\left(Q_R + Q_L + Q_M + Q_D - \frac{1}{\rho}\left[\frac{\partial J_S}{\partial z} - T\sum_k C_{pk}\frac{\partial J_{q_k}}{\partial z}\right]\right) + K_T \tag{A.6.4}$$

where

$$S_{q_k} = \frac{\dot{m}_k}{m} - q_k\frac{\dot{m}}{m} \tag{A.6.5}$$

In addition we have the equation of state

$$p = \rho T\,(R_d q_d + R_v q_v) \tag{A.6.6}$$

and the hydrostatic equation

$$\frac{\partial p}{\partial z} = -g\rho \quad . \tag{A.6.7}$$

A.7 THE MODEL EQUATIONS

The model equations (2.2.1) - (2.2.7) and (2.2.12) - (2.2.15) are finally obtained by neglecting density changes due to precipitation or evaporation, setting $\dot{m} = 0$ in (A.6.1). This approximation is traditionally made, although it is open to question.

In addition, Q_M is set to zero, an approximation of the same order as the assumption of no variation of latent heat with temperature that is made in the parametrizations.

The governing equations are

$$\frac{d\underset{\sim}{v}_h}{dt} = -\frac{1}{\rho}\nabla_h p - 2\,(\underset{\sim}{\Omega} \times \underset{\sim}{v}_h)_h - \frac{1}{\rho}\frac{\partial J_{\underset{\sim}{v}_h}}{\partial z} + K_{\underset{\sim}{v}_h} \tag{A.7.1}$$

$$\frac{dT}{dt} = \frac{R_d T_v}{pC_p}\frac{dp}{dt} + \frac{1}{C_p}\left(Q_R + Q_L + Q_D - \frac{1}{\rho}\left[\frac{\partial J_S}{\partial z} - C_{pd}T\,(\delta - 1)\frac{\partial J_{q_v}}{\partial z}\right]\right) + K_T \tag{A.7.2}$$

$$\frac{dq_v}{dt} = S_{q_v} - \frac{1}{\rho}\frac{\partial J_{q_v}}{\partial z} \tag{A.7.3}$$

$$\frac{dq_w}{dt} = S_{q_w} - \frac{1}{\rho}\frac{\partial J_{q_w}}{\partial z} \tag{A.7.4}$$

$$p = \rho R_d T_v \tag{A.7.5}$$

$$\frac{\partial p}{\partial z} = -g\rho \tag{A.7.6}$$

with

$$T_v = T\,(1 + (\frac{1}{\varepsilon} - 1)\,q_v) \tag{A.7.7}$$

In this case

$$C_p = C_{pd}\,(1 - q_v) + C_{pv}q$$

which is written

$$C_p = C_{pd}\,(1 + (\delta - 1)\,q_v) \tag{A.7.8}$$

where $\delta = \dfrac{C_{pv}}{C_{pd}}$.

The model equations then follow from a change from z- to η-coordinates, the formalism for which is given by Kasahara (1974), and from rewriting the adiabatic terms in their usual form for a spherical geometry.

Appendix B LISTS OF CONSTANTS AND SYMBOLS

Symbol	Definition		Value	Unit
a	radius of the earth		6.371×10^6	m
A_k	constant defining the vertical coordinate			Pa
B	Planck function			Wm^{-2}
B_k	constant defining the vertical coordinate			-
C_m, C_h	drag coefficient for momentum and heat			-
C_p	$= C_{pd}[1 + (\delta - 1)\, q_v]$:= specific heat of moist air at constant pressure			$J\ kg^{-1}\ K^{-1}$
C_{pd}	specific heat of dry air at constant pressure		1005.46	$J\ kg^{-1}\ K^{-1}$
C_{pv}	specific heat of water vapour at constant pressure		1869.46	$J\ kg^{-1}\ K^{-1}$
D	$= \frac{1}{a}\left(\frac{1}{1-\mu^2}\frac{\partial U}{\partial \lambda} + \frac{\partial V}{\partial \mu}\right)$:= divergence			s^{-1}
E	$= \frac{1}{2} \cdot (U^2 + V^2) / (1 - \mu^2)$:= kinetic energy/unit mass			$m^2 s^{-2}$
f	$= 2\Omega \sin\theta$:= Coriolis parameter			s^{-1}
g	acceleration of gravity		9.80665	$m\ s^{-2}$
G	$= \phi + E$:= total dry energy per unit mass			$m^2 s^{-2}$
I_0	solar constant	ECHAM1/2:	1365	$W\ m^{-2}$
		ECHAM3:	1367	$W\ m^{-2}$
J_X	vertical eddy flux of variable X			$[x]\ kg\ m^{-2}\ s^{-1}$
k	von Karman constant		0.4	-
K_m, K_h	vertical eddy diffusion coefficient for momentum and heat			$m^2\ s^{-1}$
K_x	tendency of variable X due to horizontal diffusion			$[x]\ s^{-1}$
l_m, l_h	mixing lengths for momentum and heat			m
L	Monin-Obukhov length			m
L_s	latent heat of sublimation		$2.8345 * 10^6$	$J\ kg^{-1}$
L_t	leaf area index		4	-
L_v	latent heat of vapourisation		$2.5008 * 10^6$	$J\ kg^{-1}$
m	zonal wave number			-
M_+	upward radiative flux density			Wm^{-2}
M_-	downward radiative flux density			Wm^{-2}
n	meridional index			-
N_G	number of Gaussian latitudes			-

p	pressure		Pa
p_0	reference pressure for vertical coordinate		Pa
p_S	surface pressure		Pa
P_x	parameterized tendency of variable x		$[x]s^{-1}$
$P_n^m(\mu)$	associated Legendre function of the first kind		-
q_i	ice mixing ratio		kg kg^{-1}
q_l	liquid water mixing ratio		kg kg^{-1}
q_s	saturation water vapour mixing ratio		kg kg^{-1}
q_v	water vapour mixing ratio		kg kg^{-1}
q_w	cloud water mixing ratio ($q_w = q_i + q_l$)		kg kg^{-1}
Q_Y	heating rate due to physical process Y		K s^{-1}
R	precipitation rate		kg m^{-2} s^{-1}
R_d	gas constant for dry air	287.05	J $kg^{-1}K^{-1}$
R_v	gas constant for water vapour	461.51	J $kg^{-1}K^{-1}$
Ri	Richardson number		-
Ri_{crit}	critical Richardson number		-
Ri_m	moist Richardson number		-
s	dry static energy		m^2 s^{-2}
s_*	scaling dry static energy		m^2 s^{-2}
S_{q_v}	change of q_v due to condensation or evaporation		s^{-1}
S_{q_w}	change of q_w due to condensation, evaporation or precipitation		s^{-1}
Sn	snow depth (liquid water equivalent)		m
Sn_{cr}	critical snow depth (liquid water equivalent)		m
t	time		s
T	temperature		K
T_m	ice melting temperature	273.16	K
T_S	surface temperature		K
T_{Sn}	temperature of the snow pack		K
T_v	$= T[1 + (1/\varepsilon - 1)\,q_v]$:= virtual temperature		K
u	zonal wind		m s^{-1}
u_*	scaling velocity		m s^{-1}
U	$= u \cdot cos\theta$:= scaled zonal wind		m s^{-1}
v	meridional wind		m s^{-1}
$\underset{\sim}{v}_h$	$= (u, v)$:= horizontal wind vector		m s^{-1}

V	$= v \cdot cos\theta$:= scaled meridional wind		m s^{-1}
$w(\mu)$	quadrature (or Gaussian) weight		-
W_l	skin reservoir content		m
W_{lmax}	maximum skin reservoir content	2.*10^{-4}	m
W_{lmx}	local maximum skin reservoir		m
W_S	soil moisture		m
W_{Smax}	field capacity	0.2	m
z	height		m
z_0	roughness length		m
δ	$= C_{pv}/C_{pd}$		-
δ_s	solar declination		-
ε	R_d/R_v		-
η	$\eta_k = A_k/p_0 + B_k$:= generalized vertical coordinate		-
$\dot{\eta}$	$= d\eta/dt = \eta$ -coordinate vertical velocity		s^{-1}
θ	latitude		-
Θ	potential temperature		K
Θ_l	liquid water potential temperature		K
Θ_v	virtual potential temperature		K
ϕ	$g \cdot z$, geopotential		m^{2}s^{-2}
ϕ_m, ϕ_h	universal flux profile functions for momentum and heat		-
κ	$= R_d/C_{pd}$		-
λ	longitude		-
λ_m, λ_h	asymptotic mixing length for momentum and heat		-
μ	$= sin\theta$		-
ξ	$= \frac{1}{a}\left(\frac{1}{1-\mu^2}\frac{\partial V}{\partial \lambda} - \frac{\partial U}{\partial \mu}\right)$: relative vorticity		s^{-1}
ρ	air density		kg m^{-3}
ρ_w	density of liquid water	1000	kg m^{-3}
ρ_{Sn}	density of snow		kg m^{-3}
σ	$= p/p_s$		-
σ	Stefan-Boltzmann constant	5.67*10^{-8}	W m^{-2}K^{-4}
ϕ	$= g \cdot z$:= geopotential height		m^{2}s^{-2}
ϕ_s	surface geopotential height		m^{2}s^{-2}
χ	velocity potential		m^{2}s^{-1}

ψ	stream function		m^2s^{-1}
ω	$= dp/dt$:= p-coordinate vertical velocity		Pa s^{-1}
$\tilde{\omega}$	single scattering albedo		-
Ω	angular velocity of earth	$7.292*10^{-5}$	s^{-1}

Appendix C GAUSSIAN LATITUDES FOR ECHAM3-TRUNCATIONS

NOTE: The Gaussian latitudes are symmetrical to the equator !
Therefore only the values from the North pole to the equator are printed.

The Gaussian latitudes are scanned by subroutine SCAN1. To get the current numbers of ILAT, the number count from the North pole, and IROW, the number count in the model, write

```
ITASK=IQTASK()
ILAT=NLAT(ITASK+1)
IROW=NROW(ITASK+1)
```

NLAT and NROW belong to the commonblock COMCTL.

C.1 THE "GAUSSIAN" LATITUDES FOR T21-TRUNCATION

No. from North pole (ILAT)	No. in the model (IROW)	Latitude (^{o})	No. from North pole (ILAT)	No. in the model (IROW)	Latitude (^{o})
1	1	85.760	9	17	41.532
2	3	80.268	10	19	35.995
3	5	74.744	11	21	30.457
4	7	69.212	12	23	24.919
5	9	63.678	13	25	19.382
6	11	58.142	14	27	13.844
7	13	52.606	15	29	8.306
8	15	47.069	16	31	2.768

C.2 THE "GAUSSIAN" LATITUDES FOR T42-TRUNCATION

No. from North pole (ILAT)	No. in the model (IROW)	Latitude (^{o})	No. from North pole (ILAT)	No. in the model (IROW)	Latitude (^{o})
1	1	87.863	17	33	43.254
2	3	85.096	18	35	40.463
3	5	82.312	19	37	37.673
4	7	79.525	20	39	34.882
5	9	76.736	21	41	32.091
6	11	73.947	22	43	29.301
7	13	71.157	23	45	26.510
8	15	68.367	24	47	23.720
9	17	65.577	25	49	20.929
10	19	62.787	26	51	18.138
11	21	59.997	27	53	15.348
12	23	57.206	28	55	12.557
13	25	54.416	29	57	9.767
14	27	51.625	30	59	6.976
15	29	48.835	31	61	4.185
16	31	46.044	32	63	1.395

C.3 THE "GAUSSIAN" LATITUDES FOR T63-TRUNCATION

No. from North pole (ILAT)	No. in the model (IROW)	Latitude (^{o})	No. from North pole (ILAT)	No. in the model (IROW)	Latitude (^{o})
1	1	88.572	25	49	43.833
2	3	86.722	26	51	41.968
3	5	84.861	27	53	40.102
4	7	82.998	28	55	38.237
5	9	81.134	29	57	36.372
6	11	79.270	30	59	34.507
7	13	77.405	31	61	32.641
8	15	75.541	32	63	30.776
9	17	73.676	33	65	28.911
10	19	71.811	34	67	27.046
11	21	69.946	35	69	25.180
12	23	68.080	36	71	23.315
13	25	66.215	37	73	21.450
14	27	64.350	38	75	19.585
15	29	62.485	39	77	17.719
16	31	60.620	40	79	15.854
17	33	58.755	41	81	13.989
18	35	56.890	42	83	12.124
19	37	55.024	43	85	10.258
20	39	53.159	44	87	8.393
21	41	51.294	45	89	6.528
22	43	49.429	46	91	4.663
23	45	47.563	47	93	2.797
24	47	45.698	48	95	0.932

C.4 THE "GAUSSIAN" LATITUDES FOR T106-TRUNCATION

No. from North pole (ILAT)	No. in the model (IROW)	Latitude (^{o})	No. from North pole (ILAT)	No. in the model (IROW)	Latitude (^{o})
1	1	89.141	41	81	44.298
2	3	88.029	42	83	43.177
3	5	86.910	43	85	42.055
4	7	85.790	44	87	40.934
5	9	84.669	45	89	39.812
6	11	83.548	46	91	38.691
7	13	82.427	47	93	37.569
8	15	81.306	48	95	36.448
9	17	80.185	49	97	35.326
10	19	79.063	50	99	34.205
11	21	77.942	51	101	33.083
12	23	76.821	52	103	31.962
13	25	75.699	53	105	30.840
14	27	74.578	54	107	29.719
15	29	73.457	55	109	28.597
16	31	72.335	56	111	27.476
17	33	71.214	57	113	26.355
18	35	70.092	58	115	25.233
19	37	68.971	59	117	24.112
20	39	67.849	60	119	22.990
21	41	66.728	61	121	21.869
22	43	65.606	62	123	20.747
23	45	64.485	63	125	19.626
24	47	63.363	64	127	18.504
25	49	62.242	65	129	17.383
26	51	61.120	66	131	16.261
27	53	59.999	67	133	15.140
28	55	58.878	68	135	14.018
29	57	57.756	69	137	12.897

No. from North pole (ILAT)	No. in the model (IROW)	Latitude (^{o})	No. from North pole (ILAT)	No. in the model (IROW)	Latitude (^{o})
30	59	56.635	70	139	11.775
31	61	55.513	71	141	10.654
32	63	54.392	72	143	9.532
33	65	53.270	73	145	8.411
34	67	52.149	74	147	7.289
35	69	51.027	75	149	6.168
36	71	49.906	76	151	5.046
37	73	48.784	77	153	3.925
38	75	47.663	78	155	2.803
39	77	46.541	79	157	1.682
40	79	45.420	80	159	0.560

Appendix D LAND-SEA MASKS FOR ECHAM3-TRUNCATIONS

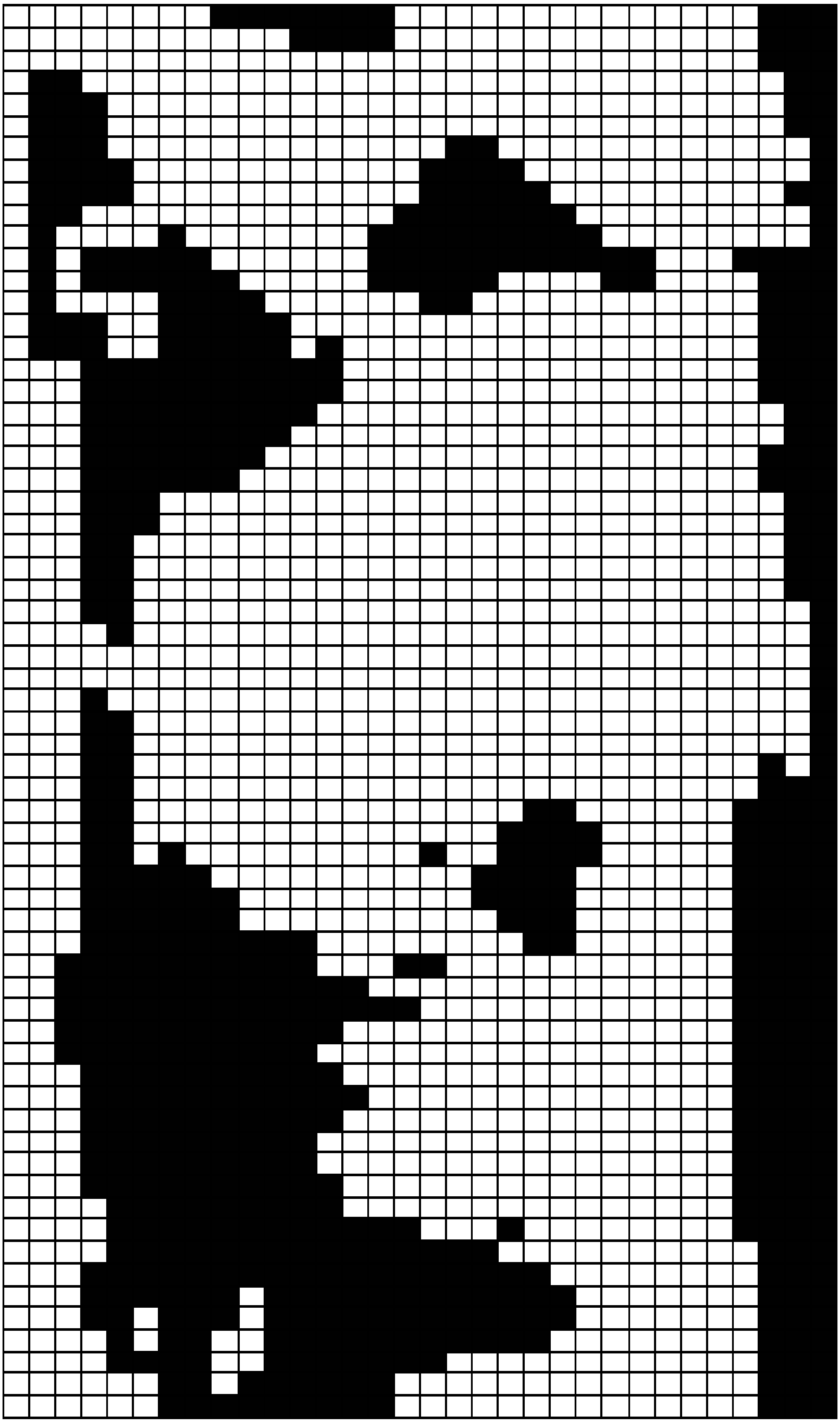


Figure 22 Land-sea mask for T21 model truncation

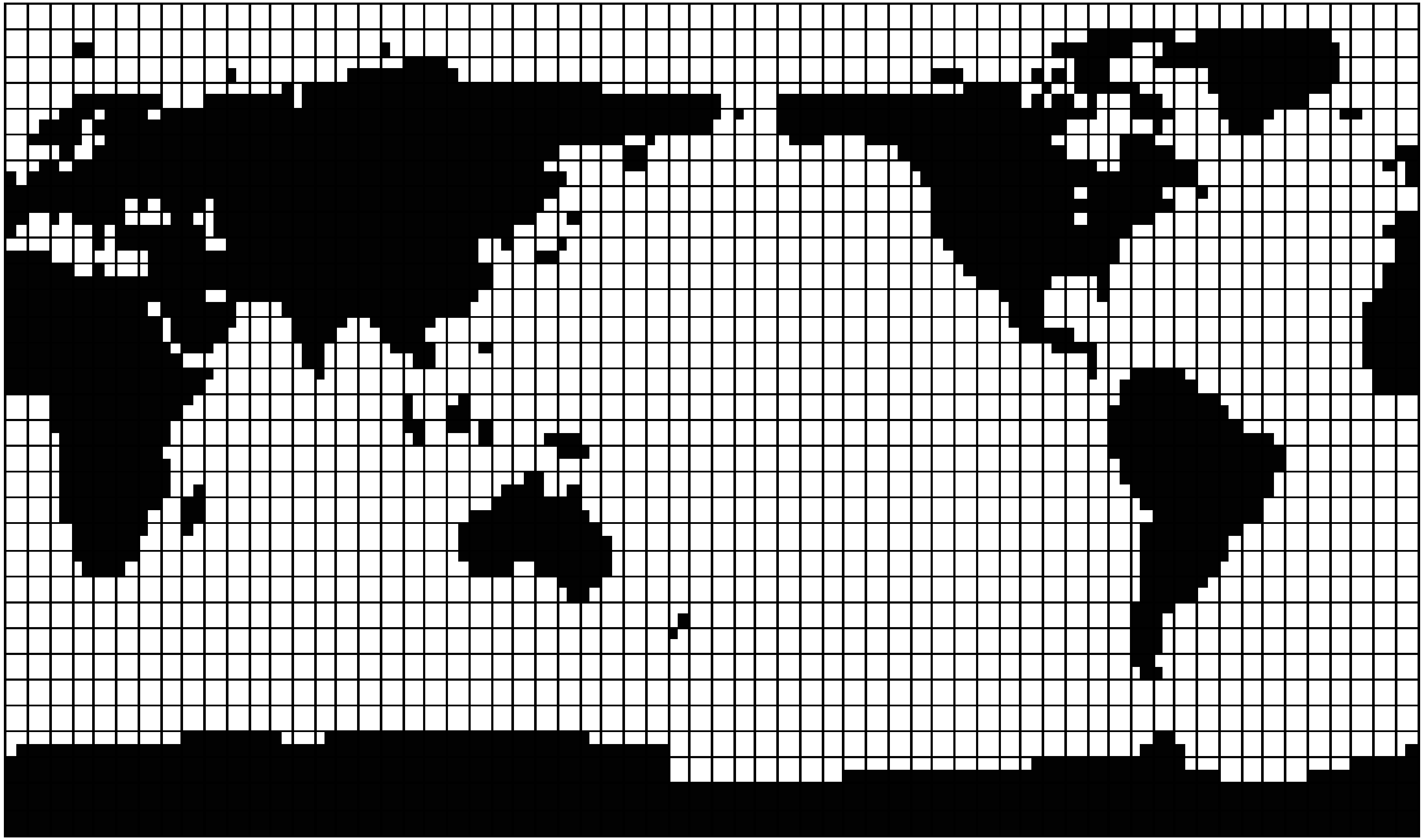


Figure 23 Land-sea mask for T42 model truncation

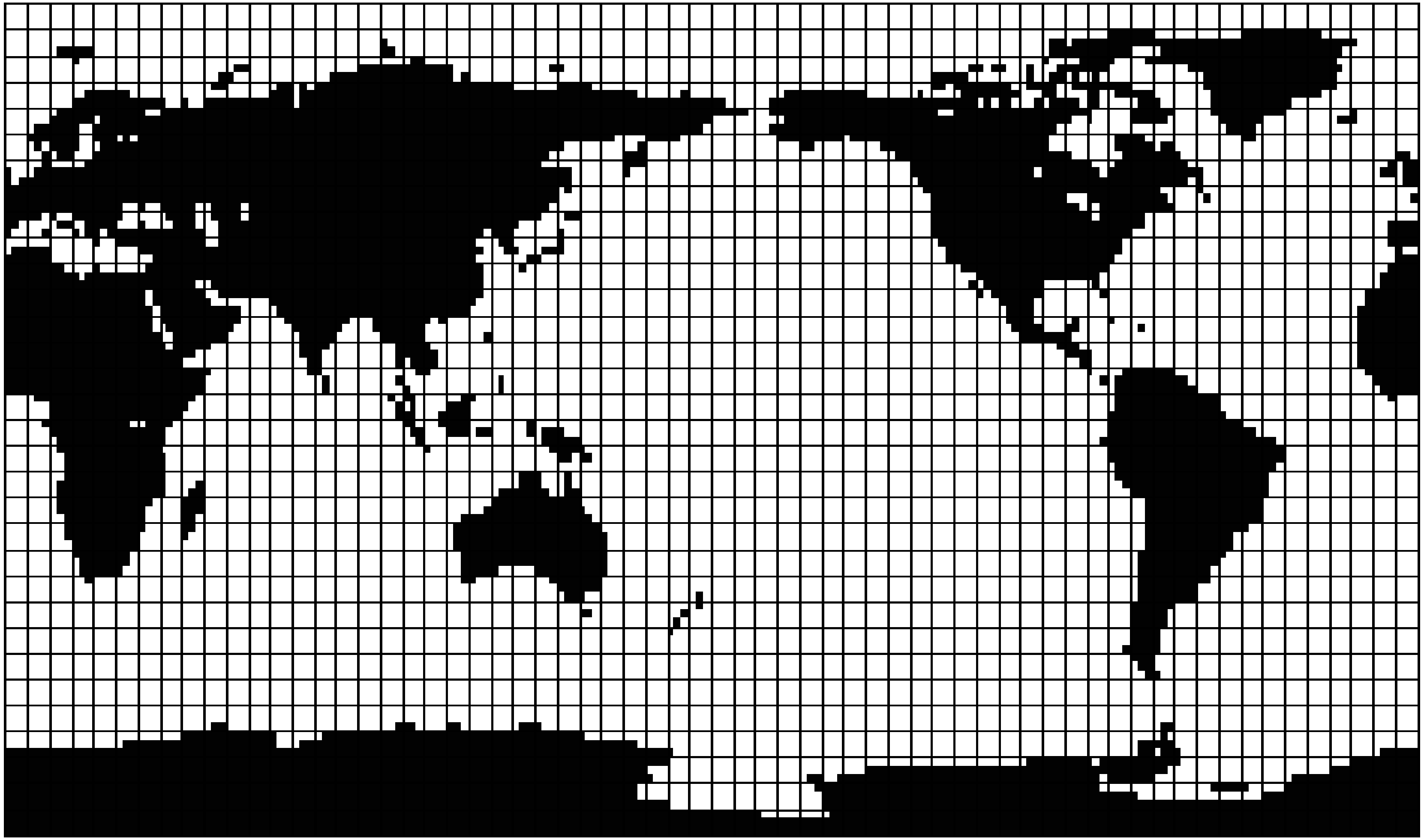


Figure 24 Land-sea mask for T63 model truncation

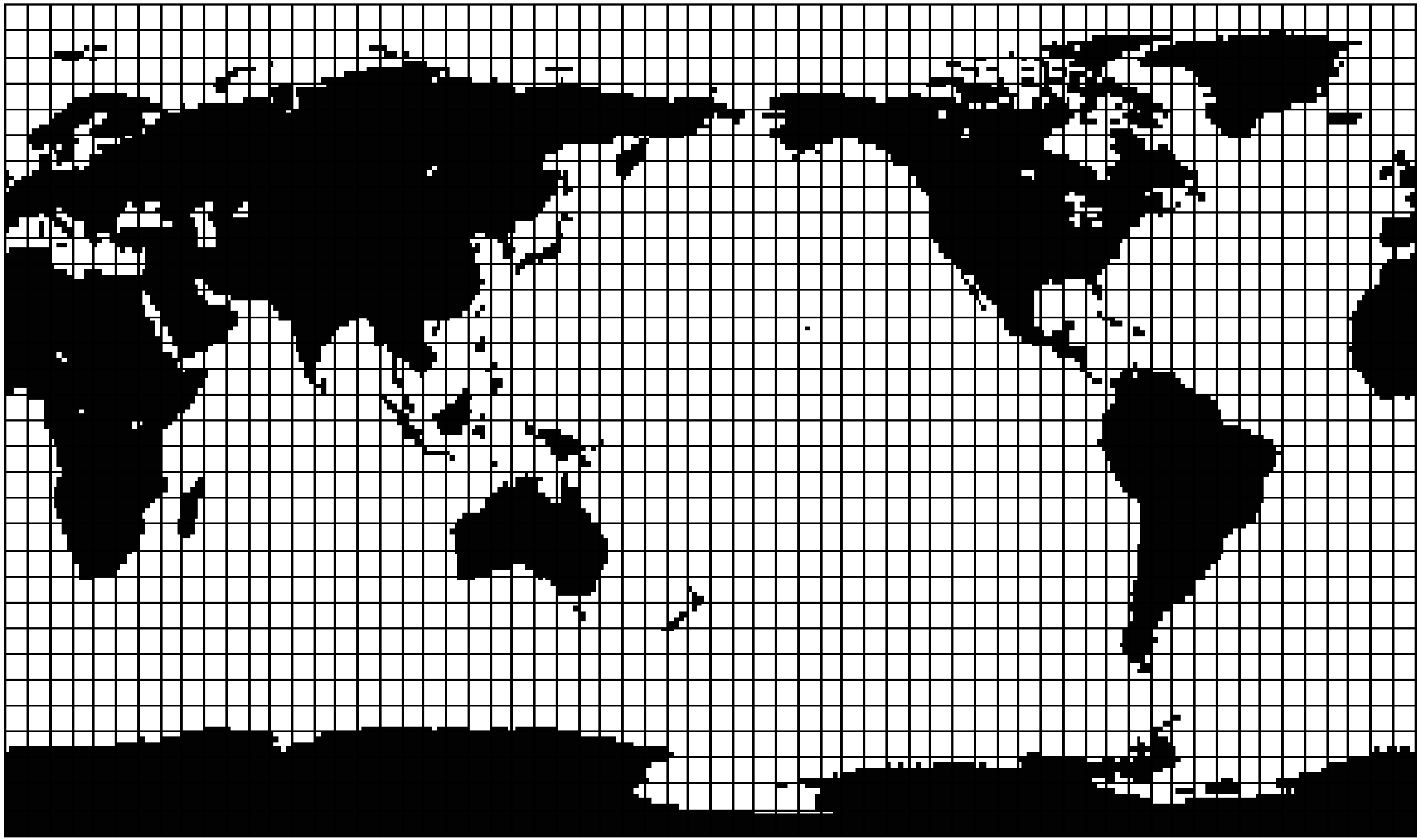


Figure 25 Land-sea mask for T106 model truncation

Appendix E PROGRAMMING CONVENTIONS

E.1 INTRODUCTION

The programming standard on which the following conventions are based was derived from the OLYMPUS system , and is called DOCTOR.

E.2 PROGRAM STRUCTURE

E.2.1 Subtasks

Systems developed using this standard are, where reasonable, organised into a hierarchical, tree-like structure. The climate simulation is divided into major subtasks, each with a distinct purpose. Subtasks consist of a control routine together with a set of computational routines. The subtask is always entered by calling the control routine. The control routine need not always be called from the same location within the code, but is always called to control the calling of some or all of the set of computational routines comprising the subtask. In this way subtasks can be self contained units, and alternative subtasks can be substituted when desired.

Subtasks may extend over a wide range of logically connected processes, such as radiation, dynamics, parameterization, etc., or may be of more limited scope, such as horizontal diffusion, etc.

E.2.2 Utility routines

A set of utility routines is available. These are taken from the set of OLYMPUS utility routines. They are used to output messages, list values of variables, reset values, etc. . Table 6 gives a list of the utility routines available, and the functions they perform.

Table 6 Utility routines

NAME	PARAMETERS	PURPOSE
RVAR	KNAME, PVALUE	Print name and value of real variable
IVAR	KNAME, KVALUE	Print name and value of integer variable
HVAR	KNAME, HPVALUE	Print name and value of Hollerith variable
LVAR	KNAME, LPVAL	Print name and value of logical variable
RARRAY	KNAME, PA, KDIM	Print name and values of real array
IARRAY	KNAME, KA, KDIM	Print name and values of integer array
HARRAY	KNAME, HPA, KDIM	Print name and values of Hollerith array
LARRAY	KNAME, LPA, KDIM	Print name and values of logical array
RARAY2	KNAME, PA, KDIMX, KX, KY	Print name and values of doubly subscripted real array
RESETR	PA, KDIM, PVALUE	Reset real array to specified value
RESETI	KA, KDIM, KVALUE	Reset integer array to specified value
RESETH	HPA, KDIM, HPVALUE	Reset Hollerith array to specified value
RESETL	LPA, KDIM, KLVAL	Reset logical array to specified value
SCALER	PA, KDIM, PC	Scale real array by a real value
SCALEI	KA, KDIM, KC	Scale integer array by an integer value
SIGNR	PA, KDIM	Change the sign of a real array
SIGNI	KA, KDIM	Change the sign of an integer array
COPYR	PA1, PA2, KDIM	Copy one real array into another
COPYI	KA1, KA2, KDIM	Copy one integer array into another

E.3 SUBROUTINE STRUCTURE

Subroutines should perform a single well defined task. The first section of each subroutine should contain a documentation of the following form:

a) A C**** statement with the title beginning with the subroutine name.
b) The author's name and the date written.
c) Modification details (name and date).
d) Headed sections, giving the following information:
    1) PURPOSE — the function of the routine,
    2) INTERFACE — how the routine receives its data and returns its results,
    3) METHOD — how the results are obtained,
    4) EXTERNALS — what external routines are required,
    5) REFERENCE — references to external documentation, if any,
    6) MODIFICATIONS — date and nature of any modification made to the routine.

This section of the subroutine should be regarded as the documentation that would be supplied to an external user of the routine. It should contain information which can stand alone as a separate document, and sufficient information to enable an external user to decide whether the routine is suitable for his/her pupose, and to enable him/her to use it.

The computational part of each subroutine should be divided into sections and sub-sections. These should be numbered for cross reference purposes. Subsection n of section m should be numbered "m.n". Sections should be ruled of from one another by a comment containing C in column 1 and "-----------" in columns 7 to 71. Each section should begin with a title, which should be underlined by the next statement. Sections should begin with a CONTINUE statement of the form

```
m00   CONTINUE
```

enabling branches to be made to the beginning of the section. A sub-section should begin with a title, followed by a CONTINUE statement of the form

```
mn0   CONTINUE
```

enabling branches to be made to the sub-section. Sub-section titles should not be underlined, and may be intended with respect to section titles to produce well formatted documentation when listed.

E.4 COMMON BLOCK STRUCTURE

COMMON blocks contain a documentation defining the variables they contain in tabular form:

a) A C** statement beginning with the COMMON block name and indicating its purpose.
b) The following table headings:
   NAME TYPE PURPOSE
c) A table entry for each variable in the COMMON block.

To avoid duplication of these comments in each routine calling the COMMON block, some or all of the comments may be enclosed in UPDATE *DEFINE brackets, using

*IF DEF,DOC

and

*ENDIF

Sequences of comments of the form

```
C
C       ----------------------------------------------
C
```

should be used where appropriate to improve the layout.

E.5 FORMAT STATEMENTS

FORMAT statements are used where several READ/WRITE use the same format, or where long FORMAT definitions would make a READ or WRITE statement long, complicated, or difficult to read. In general, use should be made of the ability to code format specifications within the READ or WRITE statement. This ability is espacially useful for printing messages.

Where used, FORMAT statements should be numbered within the range of 9901 to 9999, and gathered together at the end of the routine in which they are referenced.

E.6 BLOCK STRUCTURE

Statement numbers should not be associated with executable statements. With the exception of FORMAT statements, the only numbered statement should be CONTINUE statements. Each DO loop should end on a separate CONTINUE statement.

IF-THEN-ELSE sequences should make use of indentation to illustrate the scope of each conditional block. ENDIF should always be coded as a single word.

E. g.:

```
IF (NROW.EQ.MAXROW) THEN
   JJ=JJ+1
   CALL SPOLE
   IF (JJ.GT.JTEST) THEN
      JJ2=JJ-ILIM
      CALL SSPOLE
   ENDIF
ELSE  CALL NOTSPOL
ENDIF
```

E.7 VARIABLE NAMES

Variable names begin with significant prefix letters. These prefix letters not only indicate the type of variable (real, integer, etc.) but also the status of the variable (local, common, etc.).

1) Variable type

LOGICAL	variables begin with	L.
INTEGER	variables begin with	I, J, K, M, or N.
HOLLERITH	variables begin with	H.
CHARACTER	variables begin with	Y and are defined *8.

All other variables are of type REAL

2) Variable status

The status of variables is indicated in the following way:

a) Dummy arguments:

Dummy arguments begin with	LP	if LOGICAL
	K	if INTEGER
	P	if REAL
	HP	if HOLLERITH
	YP	if CHARACTER

b) Local variables:

variables local to a routine begin with	LO	if LOGICAL
	I	if INTEGER
	Z	if REAL
	HO	if HOLLERITH
	YO	if CHARACTER

c) Loop control variables:
All loop control variables begin with J.

d) Variables in COMMON blocks:
All remaining prefixes are available for variables located in COMMON blocks.

Typing of variables should be achieved by using:

IMPLICIT LOGICAL(L), INTEGER(H), CHARACTER*8(Y)

E.8 MEMORY MANAGEMENT

E.8.1 General

The memory manager is application independent. The remainder of this section describes the way in which the model uses it.

E.8.2 Method of use

The memory manager recognizes two types of storage - long term storage of arrays required by several routines, and temporary storage used for work space within a routine.

A list of memory manager routines used by the model is given in Table 7.

Arrays are allocated by calling ALLOCA. Contiguous memory locations may be obtained by repeated calls to the alternative allocation routine ALLOCB. Arrays allocated in one routine may be located in a subsequent routine by calling LOCATE. Storage no longer required is returned to the manager for possible future allocation by calling UNLOC.

Table 7 Memory manager routines

ROUTINE	PURPOSE
ALLOCA	Allocate space
ALLOCB	Allocate contiguous space
LOCATE	Locate allocated space
UNLOC	Release allocated space

E.8.3 Example

A simple example for a subroutine using the memory manager is given below.

Four arrays are involved in this example:

1. Long term arrays which are already allocated but not local to this subroutine:
   TM1(NLP2,NLEV), and
   TE(NLP2,NLEV).
2. Short term arrays (only for use in this subroutine):
   ZTP1(NLON,NLEV), and
   ZTEX(NLON,NLEV).

```
      SUBROUTINE EXAMPLE
C
C**** *EXAMPLE* - DEMONSTRATES PROGRAMMING CONVENTIONS
C
C     U. SCHLESE        DKRZ - HAMBURG   JUL-92
C
C     PURPOSE.
C     -------
C
C     THIS ROUTINE DEMONSTRATES THE *DOCTOR* PROGRAMMING CONVENTIONS
C     AND THE USE OF THE MEMORY MANAGER.
C
C     INTERFACE.
C     ----------
C
C     *EXAMPLE* IS CALLED FROM *MAIN*.
C     IT TAKES ITS INPUT FROM THE LONG TERM STORAGE:
C     TM1 : TEMPERATURE AT FULL LEVELS
C     TE  : TENDENCY OF TEMPERATURE
C
C     IT RETURNS ITS OUTPUT TO THE SAME SPACE:
C     TE: UPDATED BY PROCESSES CALCULATED IN *SUB1*
C
C     METHOD.
C     -------
C
C     STRAIGHTFORWARD.
C
C     EXTERNALS.
C     ----------
C
C     *LOCATE*   LOCATE LONG TERM STORAGE
C     *ALLOCA*   ALLOCATE TEMPORARY STORAGE
C     *SUB1*     UPDATES TENDENCIES
C
C     REFERENCE.
C     ----------
C
C     NONE.
C
C     MODIFICATIONS.
C     --------------
C
C     NONE.
C
C
C*    *COMMON* *COMCTL* - CONTROL VARIABLES FOR MODEL HOUSEKEEPING.
C
C     NAME      TYPE        PURPOSE
C     ----      ----        -------
C
C     *NLEV*    *INTEGER*   NUMBER OF VERTICAL LEVELS.
C     *NLON*    *INTEGER*   NUMBER OF POINTS ON EACH LATITUDE LINE.
C     *NLP2*    *INTEGER*   NUMBER OF POINTS ON EACH LATITUDE LINE + 2.
C
C
      COMMON /COMCTL/
     I          NLEV,NLON,NLP2
```

```
C
C     ---------------------------------------------------------------
C
      POINTER   (ITM1,TM1(NLP2,NLEV)),            ! Declare managed arrays
     *          (ITE,TE(NLP2,NLEV)),              ! by means of
     *          (IZTP1,ZTP1(NLON,NLEV)),          ! POINTER statements
     *          (IZTEX,ZTEX(NLON,NLEV))
C
C     ---------------------------------------------------------------
C
C*    1. LOCATE AND POSITION SPACE.
C        ------ --- -------- -----
C
  100 CONTINUE
      ITASK=IQTASK()                              ! Get current task number.
      IGPTYPE=ITASK+3                             ! Set type of data
C
      CALL LOCATE(ITM1,'TM1',IGPTYPE)             ! Get addresses (=POINTER
      CALL LOCATE(ITE,'TE',IGPTYPE)               ! values) of TM1 and TE
C
      CALL ALLOCA(IBASE,2*NLON*NLEV,'EXAMPLE',99) ! Allocate temporary storage
      IZTP1=IBASE                                 ! for ZTP1 and ZTEX and
      IZTEX=IZTP1+NLON*NLEV                       ! calculate addresses (alter-
                                                  ! native: automatic arrays)
C
C    ---------------------------------------------------------------
C
C*    2. PREPARE INPUT FOR *SUB1*.
C        ------- ----- --- ------
C
  200 CONTINUE
C
      DO 202 JK=1,NLEV
      DO 201 JL=1,NLON
      ZTP1(JL,JK)=TM1(JL,JK)+ZTMST*TE(JL,JK)      ! Use arrays within the
      ZTEX(JL,JK)=EXP(ZTP1(JL,JK))                ! code as usual
  201 CONTINUE
  202 CONTINUE
C
C----------------------------------------------------------------------------
C
C*    3. UPDATE TENDENCIES.
C        ------ -----------
C
  300 CONTINUE
C
      CALL SUB1(ZTP1,ZTEX,TE)
C
C----------------------------------------------------------------------------
C
C*    4. RELEASE SPACE.
C        ------- ------
C
  400 CONTINUE
      CALL UNLOC('EXAMPLE',99)                    ! Release temporary space
C
C----------------------------------------------------------------------------
C
      RETURN
      END
```

E.8.4 Managed arrays

Long term arrays are identified by the memory manager by means of a name and a code. Throughout the climate simulation routines code are allocated according to grid type or sub-system, as indicated in Table 8, while the actual array name is used where possible. The purpose of this dual identification is to ensure uniqeness even if the same name is used for different storage areas in different subsystems.

Short-term arrays are identified by means of a name and a specific code of 99. By convention, the name used is that of the calling routine. A routine may repeatedly request temporary array space in this way, and releas all of the allocated space by a single call to UNLOC.

Managed arrays are declared by means of POINTER statements. The pointer values are assigned by the memory manager.

Table 8 Grid code used with the Memory Manager

GRID CODE	CODE TYPE	TYPE OF DATA
1	ISY	Basic control blocks (SDS, DDR's, etc.)
3	IGP	Grid point data
20	ILC	Transformation constants
30	ISP	Spectral coefficients
40	IFC	Fourier coefficients
60	IDI	Other diagnostics
70	IRD	Radiation workspace
99	none	Temporary storage

Table 9 gives a list of managed arrays used by the ECHAM3 climate model and the name of the subroutines in which allocation (ALLOC) and release (UNLOC) of the arrays is done. Refer to the footnotes to obtain the higher level calling subroutines.

Table 9 List of managed arrays, ALLOCS and UNLOCS (Sheet 1 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
ACLC ACLCM	IGP	BUFGRD [1]	RELBUF [2]	cloud cover [fract.]
ACLCAC ACLCACM	IGP	BUFGRD [1]	RELBUF [2]	cloud cover [fract.*s] (accumulated) [9]
ACLCOV ACLCOVM	IGP	BUFGRD [1]	RELBUF [2]	total cloud cover [fract.*s] (accumulated) [9]
AHFL AHFLM	IGP	BUFGRD [1]	RELBUF [2]	surface latent heat flux [J/m**2] (accumulated) [9]
AHFS AHFSM	IGP	BUFGRD [1]	RELBUF [2]	surface sensible heat flux [J/m**2] (accumulated) [9]
ALB ALBM	IGP	BUFGRD [1]	RELBUF [2]	surface background albedo [fract.]
ALBEDO ALBEDOM	IGP	BUFGRD [1]	RELBUF [2]	surface albedo [fract.]
ALNPR	IGP	SCAN1	PHYSC	logarithm of pressure ratios
ALPHA	IGP	SCAN1	PHYSC	α_k (see equation (2.4.2.12) in section 2.4.2)
ALPS	IGP	SCAN1	SCAN1	logarithm of surface pressure
ALPSE	IGP	SCAN1	SCAN1	tendency of logarithm of surface pressure
ALPSF	IGP	BUFGRD [1]	RELBUF [2]	logarithm of surface pressure (semi filtered) [6]
ALPSM1	IGP	BUFGRD [1]	RELBUF [2]	logarithm of surface pressure at t - Δt
ALWC ALWCM	IGP	BUFGRD [1]	RELBUF [2]	liquid water content [kg/kg]
ALWCAC ALWCACM	IGP	BUFGRD [1]	RELBUF [2]	liquid water content [(kg/kg)*s] (accumulated) [9]
ALWCVI ALWCVIM	IGP	BUFGRD [1]	RELBUF[2]	vertically integrated liquid water content [(kg/m**2)*s] (accumulated) [9]
AMU0	IGP	RADHEAT RADINT	RADINT RADHEAT	cosine of solar zenith angle
ANMD	ILC	BUFNL1 [3]	RELBUF [2]	constants for Legendre transforms
ANMI	ILC	BUFNL2 [4]	RELBUF [5]	constants for Legendre transforms
ANMIUV	ILC	BUFNL3 [4]	RELBUF [5]	constants for Legendre transforms
APH	IGP	SCAN1	STAT	half level pressure [Pa]
APHM1	IGP	PHYSC	SCAN1	half level pressure at t - Δt [Pa]
APHP1	IGP	PHYSC	PHYSC	half level pressure at t + Δt [Pa]

Table 9 List of managed arrays, ALLOCS and UNLOCS (Sheet 2 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
APM1	IGP	PHYSC	SCAN1	full level pressure at t - Δt [Pa]
APP1	IGP	PHYSC	PHYSC	full level pressure at t + Δt [Pa]
APRC APRCM	IGP	BUFGRD [1]	RELBUF [2]	convective precipitation [(m/s)*s] (accumulated) [9]
APRL APRLM	IGP	BUFGRD [1]	RELBUF [2]	large scale precipitation [(m/s)*s] (accumulated) [9]
APRS APRSM	IGP	BUFGRD [1]	RELBUF [2]	snow fall [(m/s)*s] (convective + large scale, accumulated) [9]
APS APSM	IGP	BUFGRD [1]	RELBUF [2]	surface pressure [Pa]
ARPRC ARPRCM	IGP	BUFGRD [1]	RELBUF [2]	cloud parameters for radiation (not used)
AZ0 AZ0M	IGP	BUFGRD [1]	RELBUF [2]	surface roughness [m]
CN	ISP	POSTRAD SETDYN	SCAN1	matrix used to invert the divergence equation
CVGHL	IGP	VDIFF	SURF	ratio of moisture fluxes
D	IGP	SCAN1	SCAN1	divergence [1/s]
DALPSL	IGP	SCAN1	DYN	zonal derivative of logarithm of surface pressure
DALPSM	IGP	SCAN1	DYN	meridional derivative of logarithm of surface pressure
DELP	IGP	SCAN1	DYN	pressure thickness of layers [Pa]
DEW2 DEW2M	IGP	BUFGRD [1]	RELBUF [2]	2m dew point temperature [K*s] (accumulated) [9]
DF	IGP	BUFGRD [1]	RELBUF [2]	divergence (semi filtered) [6] [1/s]
DHFQ	IGP	VDIFF	SURF	derivative of moisture flux with respect to soil moisture (not used)
DHFQW	IGP	VDIFF	SURF	derivative of moisture flux with respect to skin reservoir
DHFQS	IGP	VDIFF	SURF	derivative of moisture flux over snow with respect to snow depth
DHFT	IGP	VDIFF	SKINTEM	derivative of sensible heat flux with respect to surface temperature
DIA	IGP	PRESTAT	POSTATP	memory buffer for physical diagnostics
DIA1	IDI	SETRAD		memory buffer for radiation diagnostics
DL	IGP	S12	SCAN1	$\tilde{D}_{\lambda}$ (see equation (2.5.38) in section 2.5)
DM	IGP	SI1	SCAN1	$\tilde{D}_{\mu}$ (see equation (2.5.39) in section 2.5)

Table 9 List of managed arrays, ALLOCS and UNLOCS (Sheet 3 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
DM1	IGP	BUFGRD [1]	RELBUF [2]	divergence at t - Δt [1/s]
DQL	IGP	SCAN1	DYN	zonal derivative of specific humidity
DQM	IGP	SCAN1	DYN	meridional derivative of specific humidity
DSNAC DSNACM	IGP	BUFGRD [1]	RELBUF [2]	snow depth change [m/s]
DTL	IGP	SCAN1	DYN	zonal derivative of temperature
DTM	IGP	SCAN1	DYN	meridional derivative of temperature
DU0	IGP	SCAN1	SCAN1	derivative of zonal reference wind
DXL	IGP	SCAN1	DYN	zonal derivative of cloud water
DXM	IGP	SCAN1	DYN	meridional derivative of cloud water
EMTER EMTERM	IGP	BUFGRD [1]	RELBUF [2]	effective emissivity
EVAP EVAPM	IGP	BUFGRD [1]	RELBUF [2]	evaporation from the surface [(m/s)*s] (accumulated) [9]
FAD	IFC	BUFF4A [3,4]	RELBUF [2,5]	antisymmetric part of divergence
FADL	IFC	SCAN1	SCAN1	antisymmetric part of $\tilde{D}_\lambda$ (see equation (2.5.38) in section 2.5)
FADM	IFC	SCAN1	SCAN1	antisymmetric part of $\tilde{D}_\mu$ (see equation (2.5.39) in section 2.5)
FADU0	IFC	BUFF4A [3,4]	RELBUF [2,5]	antisymmetric part of meridional derivative of zonal reference wind at t - Δt
FAQ	IFC	BUFF4A [3,4]	RELBUF [2,5]	antisymmetric part of specific humidity
FAQ1	IFC	SCAN1	SCAN1	antisymmetric part of specific humidity at t + Δt
FAQM	IFC	BUFF4A [3,4]	RELBUF [5]	antisymmetric part of meridional derivative of specific humidity
FAR	IFC	SCAN1	SCAN1	antisymmetric part of R (see equation (2.5.40) in section 2.5)
FATP	IFC	BUFF4A [3,4]	RELBUF [2,5]	antisymmetric part of temperature and surface pressure [7]
FATP1	IFC	SCAN1	SCAN1	antisymmetric part of temperature and surface pressure at t + Δt [7]
FATPM	IFC	BUFF4A [3,4]	RELBUF [2,5]	antisymmetric part of meridional derivative of temperature and surface pressure [7]
FAU	IFC	BUFF4A [3,4]	RELBUF [2,5]	antisymmetric part of zonal wind
FAU0	IFC	BUFNF4 [3,4]	RELBUF [2,5]	antisymmetric part of zonal reference wind at t - Δt

Table 9 List of managed arrays, ALLOCS and UNLOCS (Sheet 4 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
FAUD	IFC	SCAN2	SCAN2	antisymmetric part of divergent zonal wind
FAUL	IFC	SCAN1	SCAN1	antisymmetric part of zonal reference wind
FAUR	IFC	SCAN2	SCAN2	antisymmetric part of rotational zonal wind
FAV	IFC	BUFF4A [3,4]	RELBUF [2,5]	antisymmetric part of meridional wind
FAVD	IFC	SCAN2	SCAN2	antisymmetric part of divergent meridional wind
FAVO	IFC	BUFF4A [3,4]	RELBUF [2,5]	antisymmetric part of vorticity
FAVR	IFC	SCAN2	SCAN2	antisymmetric part of rotational meridional wind
FAX	IFC	BUFF4A [3,4]	RELBUF [2,5]	antisymmetric part of cloud water
FAX1	IFC	SCAN1	SCAN1	antisymmetric part of cloud water at $t + \Delta t$
FAXM	IFC	BUFF4A	RELBUF[5]	antisymmetric part of meridional derivative of cloud water
FAZL	IFC	SCAN1	SCAN1	antisymmetric part of $\tilde{Z}_{\lambda m}$ (see equation (2.5.24) in section 2.5)
FAZM	IFC	SCAN1	SCAN1	antisymmetric part of $\tilde{Z}_{\mu m}$ (see equation (2.5.25) in section 2.5)
FOREST FORESTM	IGP	BUFGRD [1]	RELBUF [2]	forest cover [fract.]
FSD	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of divergence
FSDL	IFC	SCAN1	SCAN1	symmetric part of $\tilde{D}_{\lambda}$ (see equation (2.5.38) in section 2.5)
FSDM	IFC	SCAN1	SCAN1	symmetric part of $\tilde{D}_{\mu}$ (see equation (2.5.39) in section 2.5)
FSDU0	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of derivative of zonal reference wind at $t - \Delta t$
FSQ	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of specific humidity
FSQ1	IFC	SCAN1	SCAN1	symmetric part of specific humidity at $t + \Delta t$
FSQM	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of meridional derivative of specific humidity
FSR	IFC	SCAN1	SCAN1	symmetric part of R (see equation (2.5.40) in section 2.5)
FSTP	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of temperature and surface pressure [7]
FSTP1	IFC	SCAN1	SCAN1	symmetric part of temperature and surface pressure at $t + \Delta t$ [7]
FSTPM	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of meridional derivative of temperature and surface pressure [7]

Table 9 List of managed arrays, ALLOCS and UNLOCS (Sheet 5 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
FSU	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of zonal wind
FSU0	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of zonal reference wind at t - Δt
FSUD	IFC	SCAN2	SCAN2	symmetric part of divergent zonal wind
FSUL	IFC	SCAN1	SCAN1	symmetric part of zonal reference wind
FSUR	IFC	SCAN2	SCAN2	symmetric part of rotational zonal wind
FSV	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of meridional wind
FSVD	IFC	SCAN2	SCAN2	symmetric part of divergent meridional wind
FSVO	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of vorticity
FSVR	IFC	SCAN2	SCAN2	symmetric part of rotational meridional wind
FSX	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of cloud water
FSX1	IFC	SCAN1	SCAN1	symmetric part of cloud water at t + Δt
FSXM	IFC	BUFF4A [3,4]	RELBUF [2,5]	symmetric part of meridional derivative of cloud water
FSZL	IFC	SCAN1	SCAN1	symmetric part of $\tilde{Z}_{\lambda m}$ (see equation (2.5.24) in section 2.5)
FSZM	IFC	SCAN1	SCAN1	symmetric part of $\tilde{Z}_{\mu m}$ (see equation (2.5.25) in section 2.5)
GEOM1	IGP	PHYSC	SCAN1	geopotential relative to surface [m**2/s**2]
GEOSP GEOSPM	IGP	BUFGRD [1]	RELBUF [2]	surface geopotential (orography) [m**2/s**2]
GLAC CLACM	IGP	BUFGRD [1]	RELBUF [2]	glacier mask [0:no/1:yes]
IDDR1 IDDR2 IDDR3 IDDR4 IDDR5	ISY	GETDDR MAKEDDR	POSTS1 POSTS2 IOPOSI IOPOSR	data description records
ILAB	IGP	PHYSC	COND	flag for convective event
ISDS	ISY	MAKESD INICTL		start data set information
KE	IGP	DYN	PHYSC	change of kinetic energy [m**2/s**3]
LOLAND	IGP	PHYSC	PHYSC	flag for land/sea mask (true: land, false: sea)
NBASEC NBASECM	IGP	BUFGRD [1]	RELBUF [2]	cloud parameter for radiation (not used)

Table 9 List of managed arrays, ALLOCS and UNLOCS (Sheet 6 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
NF1A NF2A NF3A NF4A	IFC	BUFF4A [3,4]	RELBUF [2,5]	padding space for Fourier buffers
NG1A NG2A NG3A NG4A NG1B NG2B NG3B NG4B	IGP	BUFGRD [1]	RELBUF [2]	padding space for grid point buffers
NTOPC NTOPCM	IGP	BUFGRD [1]	RELBUF [2]	cloud parameter for radiation (not used)
PNMD	ILC	BUFNL1 [3]	RELBUF [2]	constants for Legendre transform
PNMI	ILC	BUFNL2 [4]	RELBUF [5]	constants for Legendre transform
PNMIUV	ILC	BUFNL3 [4]	RELBUF [5]	constants for Legendre transform
Q	IGP	SCAN1	SCAN1	specific humidity [kg/kg]
QE	IGP	SCAN1	SCAN1	tendency of specific humidity [1/s]
QF	IGP	BUFGRD [1]	RELBUF [2]	specific humidity (semi filtered) [6] [kg/kg]
QHFL	IGP	VDIFF	SURF	moisture flux at the surface [kg/(m**2*s)]
QM1	IGP	BUFGRD [1]	RELBUF [2]	specific humidity at t - Δt [kg/kg]
QSF	IGP	SI1	SI2	$\tilde{\Delta}_{tt}Q$ for specific humidity (see equation (2.5.8) in section 2.5)
QVI QVIM	IGP	BUFGRD [1]	RELBUF [2]	vertically integrated specific humidity [kg/m**2] (accumulated) [9]
RAD1 RAD2 RAD3 RAD4	IRD	RADINT	RADINT	memory buffers for radiative calculations
RDAYL	IGP	RADHEAT RADINT	RADHEAT RADINT	relative daylength (= 1 for diurnal cycle on)
RH	IGP	SCAN1	SCAN1	R (see equation (2.5.40) in section 2.5)
RNMD	ILC	BUFNL1 [3]	SCAN1	constants for Legendre transform
RSFC	IGP	PHYSC	SURF	convective rain flux at the surface
RSFL	IGP	PHYSC	SURF	large scale rain flux at the surface
RUNOFF RUNOFFM	IGP	BUFGRD [1]	RELBUF [2]	surface runoff [(m/s)*s] (accumulated) [9]

Table 9 List of managed arrays, ALLOCS and UNLOCS (Sheet 7 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
SD	ISP	SCAN1 IOPOSI		divergence [1/s]
SEAICE SEAICEM	IGP	BUFGRD [1]	RELBUF [2]	sea-ice cover [fract.]
SICED SICEDM	IGP	BUFGRD [1]	RELBUF [2]	sea-ice depth [m]]
SLM SLMM	IGP	BUFGRD [1]	RELBUF [2]	land sea mask [0:sea/1:land]
SN SNM	IGP	BUFGRD [1]	RELBUF [2]	snow depth [m]
SNM1 SNM1M	IGP	BUFGRD [1]	RELBUF [2]	snow depth at t - Δt [m]
SNMEL SNMELM	IGP	BUFGRD [1]	RELBUF [2]	snow melt [(m/s)*s] (accumulated) [9]
SQ	ISP	SCAN1 IOPOSI		specific humidity [kg/kg]
SRAD0 SRAD0M	IGP	BUFGRD [1]	RELBUF [2]	top solar radiation [J/m**2] (accumulated) [9]
SRAD0U SRAD0UM	IGP	BUFGRD [1]	RELBUF [2]	top solar radiation upward [J/m**2] (accumulated) [9]
SRADS SRADSM	IGP	BUFGRD [1]	RELBUF [2]	surface solar radiation [J/m**2] (accumulated) [9]
SRADSU SRADSUM	IGP	BUFGRD [1]	RELBUF [2]	surface solar radiation upward [J/m**2] (accumulated) [9]
SRFL	IGP	RADHEAT	SKINTEM	net solar radiative flux at the surface
SSFC	IGP	PHYSC	SURF	convective snow flux at the surface [kg/(m**2*s)]
SSFL	IGP	PHYSC	SURF	large scale snow flux at the surface [kg/(m**2*s)]
STP	ISP	SCAN1 IOPOSI		temperature [K] and logarithm of surface pressure
SU0	ISP	SCAN1 IOPOSI		zonal reference wind [m/s]
SVO	ISP	SCAN1 IOPOSI		vorticity [1/s]
SX	ISP	SCAN1 IOPOSI		liquid water content [kg/kg]
T	IGP	SCAN1	SCAN1	temperature [K]
T2MAX T2MAXM	IGP	BUFGRD [1]	RELBUF [2]	maximum 2m temperature [K] (during output interval)

Table 9 List of managed arrays, ALLOCS and UNLOCS (Sheet 8 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
T2MIN T2MINM	IGP	BUFGRD [1]	RELBUF [2]	minimum 2m temperature [K] (during output interval)
TD TDM TD3 TD3M TD4 TD4M TD5 TD5M TDCL TDCLM	IGP	BUFGRD [1]	RELBUF [2]	soil temperatures [K] (see description of soil temperatures in section 3.7 (SOIL PROCESSES))
TDM1 TDM1M TD3M1 TD3M1M TD4M1 TD4M1M TD5M1 TD5M1M TDCLM1 TDCLM1M	IGP	BUFGRD [1]	RELBUF [2]	soil temperatures at t - Δt [K] (see description of soil temperatures in section 3.7 (SOIL PROCESSES))
TE	IGP	SCAN1	SCAN1	tendency of temperature [K/s]
TEFF TEFFM	IGP	BUFGRD [1]	RELBUF [2]	(effective) sea-ice skin temperature [K]
TEMP2 TEMP2M	IGP	BUFGRD [1]	RELBUF [2]	2m temperature [K*s] (accumulated) [9]
TF	IGP	BUFGRD [1]	RELBUF [2]	temperature (semi filtered) [6] [K]
THFL	IGP	VDIFF	SKINTEM	VDIFF: sensible heat flux at the surface [W/m**2] SURF: sum of sensible and latent heat flux at the surface [W/m**2]
TM1	IGP	BUFGRD [1]	RELBUF [2]	temperature at t - Δt [K]
TOPMAX TOPMAXM	IGP	BUFGRD [1]	RELBUF [2]	maximum height of convective cloud tops [m] (during output interval)
TRAD0 TRAD0M	IGP	BUFGRD [1]	RELBUF [2]	top thermal radiation [J/m**2] (accumulated) [9]
TRADS TRADSM	IGP	BUFGRD [1]	RELBUF [2]	surface thermal radiation [J/m**2] (accumulated) [9]
TRADSU TRADSUM	IGP	BUFGRD [1]	RELBUF [2]	surface thermal radiation upward [J/m**2] (accumulated) [9]
TRSOL TRSOLM	IGP	BUFGRD [1]	RELBUF [2]	effective transmissivity

Table 9 List of managed arrays, ALLOCS and UNLOCS (Sheet 9 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
TS TSM	IGP	BUFGRD [1]	RELBUF [2]	surface temperature [K]
TSLIN TSLINM	IGP	BUFGRD [1]	RELBUF [2]	land: residual surface heat budget [W/m**2] sea-ice: conductive heat flux [W/m**2]
TSM1 TSM1M	IGP	BUFGRD [1]	RELBUF [2]	surface temperature at t - Δt [K]
TSURF TSURFM	IGP	BUFGRD [1]	RELBUF [2]	surface temperature [K*s] (accumulated) [9]
TSMAX TSMAXM	IGP	BUFGRD [1]	RELBUF [2]	maximum surface temperature [K] (during output interval)
TSMIN TSMINM	IGP	BUFGRD [1]	RELBUF [2]	minimum surface temperature [K] (during output interval)
U	IGP	SCAN1	STAT	zonal wind [m/s]
U0	IGP	SCAN1	SCAN1	reference zonal wind at t - Δt [m/s]
U10 U10M	IGP	BUFGRD [1]	RELBUF [2]	10m u-velocity [(m/s)*s] (accumulated) [9]
UF	IGP	BUFGRD [1]	RELBUF [2]	zonal wind (semi filtered) [6] [m/s]
UL	IGP	SCAN1	SCAN1	zonal reference wind [m/s]
UM1	IGP	BUFGRD [1]	RELBUF [2]	zonal wind at t - Δt [m/s] [8]
USTAR3 USTAR3M	IGP	BUFGRD [1]	RELBUF [2]	TKE for ocean mixed layer [(m**3/s**3)*s] (accumulated) [9]
USTR USTRM	IGP	BUFGRD [1]	RELBUF [2]	u-stress [Pa*s] (accumulated) [9]
USTRGW USTRGWM	IGP	BUFGRD [1]	RELBUF [2]	u-gravity wave stress [Pa*s] (accumulated) [9]
V	IGP	SCAN1	STAT	meridional wind [m/s]
V10 V10M	IGP	BUFGRD [1]	RELBUF [2]	10m v-velocity [(m/s)*s] (accumulated) [9]
VAROR	IGP	DYN	SURF	orographic variance
VARP VARPM	IGP	BURGRD [1]	RELBUF [2]	directional orographic variance (packed)
VDIS VDISM	IGP	BUFGRD [1]	RELBUF [2]	boundary layer dissipation [J/m**2] (accumulated) [9]
VDISGW VDISGWM	IGP	BUFGRD [1]	RELBUF [2]	dissipation by gravity wave drag [J/m**2] (accumulated) [9]
VEGRAT VEGRATM	IGP	BUFGRD [1]	RELBUF [2]	packed array containing orographic variance, skin reservoir and vegetation ratio

Table 9 List of managed arrays, ALLOCS and UNLOCS (Sheet 10 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
VERVEL	IGP	DYN	PHYSC	vertical velocity [Pa/s]
VF	IGP	BUFGRD [1]	RELBUF [2]	meridional wind (semi filtered) [6] [m/s]
VGRAT	IGP	DYN	SURF	vegetation ratio
VM1	IGP	BUFGRD [1]	RELBUF [2]	meridional wind at t - Δt [m/s] [8]
VO	IGP	SCAN1	SCAN1	vorticity [1/s]
VOF	IGP	BUFGRD [1]	RELBUF [2]	vorticity (semi filtered) [6] [1/s]
VOL	IGP	SCAN1	SCAN1	tendency of zonal wind [m/s**2] [8]
VOM	IGP	SCAN1	SCAN1	tendency of meridional wind [m/s**2] [8]
VOM1	IGP	BUFGRD [1]	RELBUF [2]	vorticity at t - Δt [1/s]
VSTR VSTRM	IGP	BUFGRD [1]	RELBUF [2]	v-stress [Pa*s] (accumulated) [9]
VSTRGW VSTRGWM	IGP	BUFGRD [1]	RELBUF [2]	v-gravity wave stress [Pa*s] (accumulated) [9]
WDCL WDCLM	IGP	BUFGRD [1]	RELBUF [2]	not used
WDM1 WDM1M	IGP	BUFGRD [1]	RELBUF [2]	not used
WIMAX WIMAXM	IGP	BUFGRD [1]	RELBUF [2]	maximum 2m windspeed [m/s] (during output interval)
WIND10 WIND10M	IGP	BUFGRD [1]	RELBUF [2]	10m windspeed [(m/s)*s] (accumulated) [9]
WLM	IGP	DYN	SURF	skin reservoir content of plants [m]
WLM1 WLM1M	IGP	BUFGRD [1]	RELBUF [2]	skin reservoir content of plants at t - Δt [m]
WS WSM	IGP	BUFGRD [1]	RELBUF [2]	soil wetness [m]
WSM1 WSM1M	IGP	BUFGRD [1]	RELBUF [2]	soil wetness at t - Δt [m]
X	IGP	SCAN1	SCAN1	cloud water [kg/kg]
XE	IGP	SCAN1	SCAN1	tendency of cloud water [1/s]
XF	IGP	BUFGRD [1]	RELBUF [2]	cloud water (semi filtered) [6] [kg/kg]
XHFL	IGP	PHYSC	PHYSC	cloud water flux at the surface [kg/(m**2*s)] (fog deposition)
XM1	IGP	BUFGRD [1]	RELBUF [2]	cloud water at t - Δt [kg/kg]

Table 9 List of managed arrays, ALLOCS and UNLOCS
(Sheet 11 of 11)

NAME	TYPE	ALLOC	UNLOC	VARIABLE
XSF	IGP	SI1	SI2	$\tilde{\Delta}_{tt}X$ for cloud water (see equation (2.5.8) in section 2.5)

1. Called by PRES1 at the beginning of SCAN1.
2. Called by POSTS1 at the end of SCAN1.
3. Called by SETBUF called by PRES1 at the beginning of SCAN1.
4. Called by SETBUF called by PRES2 at the beginning of SCAN2.
5. Called by POSTS2 at the end of SCAN2.
6. See equation (2.5.11) in section 2.5 (TIME INTEGRATION SCHEME)
7. Temperature at full levels and logarithm of surface pressure are stored in one array of 20 levels with logarithm of surface pressure stored in level 20.
8. Winds in the model are scaled $U = u \cdot \cos\Theta$. For practical purposes the winds and their tendencies are divided by $\cos\Theta$ at the beginning of PHYSC and are rescaled at the end of PHYSC.
9. Accumulated variables are weighted by the time step interval during accumulation. Before write out they are divided by the time of the output interval. For example the unit J/m**2 of the surface fluxes during accumulation is converted to W/m**2 before write out. The units of all postprocessed variables are listed in the code table (Table 10 in Appendix F).

Appendix F ECHAM3 CODE-LIST OF VARIABLES

The following code-list is available from /pool/POST/echam/doc/codes.

All grid variables with an internal name can be requested as model output. All spectral variables (except SX, the liquid water content) are written out by default (see description of namelist POSTCTL in Table 11). Use code 222 for output of liquid water content. All variables without an internal name are derived variables, which can be computed by the ***afterburner***[1] postprocessing package (DKRZ Technical Report: Description of programs handling files in WMO GRIB and EXTRA Format, to be published). The ***afterburner*** also computes grid point versions of spectral variables.

Two methods for the calculation of clear sky diagnostics and cloud forcing exist:

1. **Method I**, applied to the standard model, calls the radiation code once. It diagnoses clear sky or overcast events, counts them and accumulates radiative properties for the two cases . It outputs the accumutated radiative properties and the number of events separately in an output interval (6 or 12 hours, depending on the model version).
2. **Method II** calls the radiation code twice each radiation time step, first with true cloudiness and then with simulated clear sky, and averages both separately over the output interval.

For details of the two methods see Cess and Potter (1987, 1988).

type: g=grid, s=spectral, m=mean over output interval, p=packed

Table 10 Code numbers of the ECHAM-Models (Sheet 1 of 6)

code	levels	internal name	type	variable	unit
-----------------		**clear sky diagnostics (method II), not available in standard model**			----------------------
101	1	SRAFS	g m	net surface solar radiation	(clear sky) [W/m**2]
102	1	TRAFS	g m	net surface thermal radiation	(clear sky) [W/m**2]
103	1	SRAF0	g m	net top solar radiation	(clear sky) [W/m**2]
104	1	TRAF0	g m	net top thermal radiation	(clear sky) [W/m**2]
105	1	SRAF0U	g m	top solar radiation upward	(clear sky) [W/m**2]
106	1	SRAFSU	g m	surface solar radiation upward	(clear sky) [W/m**2]
107	1	TRAFSU	g m	surface thermal radiation upward	(clear sky) [W/m**2]
--					

1. To compute multi level derived variables and for interpolation to pressure levels the ***afterburner*** package needs all spectral variables and the surface geopotential (code 129). Therefore altering of namelist POSTCTL is not recommended. The output of code 129 is forced internally.

Table 10 Code numbers of the ECHAM-Models (Sheet 2 of 6)

code	levels	internal name	type	variable	unit
129	1	GEOSP	g	surface geopotential (orography)	[m**2/s**2]
130	19	STP	s	temperature	[K]
131	19		g	u-velocity	[m/s]
132	19		g	v-velocity	[m/s]
133	19	SQ	s	specific humidity	[kg/kg]
134	1	APS	g	surface pressure	[Pa]
135	19		g	vertical velocity	[Pa/s]
138	19	SVO	s	vorticity	[1/s]
139	1	TS	g	surface temperature	[K] (see also 169)
140	1	WS	g	soil wetness	[m]
141	1	SN	g	snow depth[I]	[m]
142	1	APRL	g m	large scale precipitation	[m/s]
143	1	APRC	g m	convective precipitation	[m/s]
144	1	APRS	g m	snow fall	[m/s]
145	1	VDIS	g m	boundary layer dissipation	[W/m**2]
146	1	AHFS	g m	surface sensible heat flux	[W/m**2]
147	1	AHFL	g m	surface latent heat flux	[W/m**2]
148	19		g	streamfunction	[m**2/s]
149	19		g	velocity potential	[m**2/s]
151	1		g	mean sea level pressure	[Pa]
152	1	STP(20)	s	log surface pressure	
153	19	SX	s	liquid water content	[kg/kg] (see 161, 222)
154					
155	19	SD	s	divergence	[1/s]
156	19		g	geopotential height	[gpm]
157	19		g	relative humidity[II]	[fract.]
158	1		g	tendency of surface pressure[II]	[Pa/s]
159	1	USTAR3	g m	ustar**3	[m**3/s**3]
160	1	RUNOFF	g m	surface runoff (**not ECHAM1 !**)[III]	[m/s] (see also 271)
161	19	ALWC	g	liquid water content	[kg/kg] (see also 222)

Table 10 Code numbers of the ECHAM-Models (Sheet 3 of 6)

code	levels	internal name	type	variable	unit
162	19	ACLC	g	cloud cover	[fract.] (see also 223)
163	1	ACLCV	g	total cloud cover	[fract.] (see also 164)
164	1	ACLCOV	g m	total cloud cover	[fract.]
165	1	U10	g m	10m u-velocity	[m/s]
166	1	V10	g m	10m v-velocity	[m/s]
167	1	TEMP2	g m	2m temperature	[K]
168	1	DEW2	g m	2m dew point temperature	[K]
169	1	TSURF	g m	surface temperature	[K] (see also 139)
170	1	TD	g	deep soil temperature[IV]	[K]
171	1	WIND10	g m	10m windspeed	[m/s]
172	1	SLM	g	land sea mask	[0: sea, 1: land]
173	1	AZ0	g	surface roughness	[m]
174	1	ALB	g	surface background albedo	[fract.]
175	1	ALBEDO	g	surface albedo	[fract.]
176	1	SRADS	g m	surface solar radiation	[W/m**2]
177	1	TRADS	g m	surface thermal radiation	[W/m**2]
178	1	SRAD0	g m	top solar radiation	[W/m**2]
179	1	TRAD0	g m	top thermal radiation	[W/m**2]
180	1	USTR	g m	u-stress	[Pa]
181	1	VSTR	g m	v-stress	[Pa]
182	1	EVAP	g m	evaporation	[m/s]
183	1	TDCL	g	soil temperature[IV]	[K]
194	1	WLM1	g	skin reservoir content of plants	[m]
195	1	USTRGW	g m	u-gravity wave stress	[Pa]
196	1	VSTRGW	g m	v-gravity wave stress	[Pa]
197	1	VDISGW	g m	gravity wave dissipation	[W/m**2]
201	1	T2MAX	g	maximum 2m-temperature	[K]
202	1	T2MIN	g	minimum 2m-temperature	[K]
203	1	SRAD0U	g m	top solar radiation upward	[W/m**2]

Table 10 Code numbers of the ECHAM-Models (Sheet 4 of 6)

code	levels	internal name	type	variable	unit
204	1	SRADSU	g m	surface solar radiation upward	[W/m**2]
205	1	TRADSU	g m	surface thermal radiation upward	[W/m**2]
206	1	TSN	g	snow temperature[IV]	[K]
207	1	TD3	g	soil temperatureIV	[K]
208	1	TD4	g	soil temperature[IV]	[K]
209	1	TD5	g	soil temperature[IV]	[K]
210	1	SEAICE	g	sea ice cover	[fract.]
211	1	SICED	g	sea ice depth	[m]
212	1	FOREST	g	forest cover	[fract.]
213	1	TEFF	g	(effective) sea-ice skin temperature	[K]
214	1	TSMAX	g	maximum surface temperature	[K]
215	1	TSMIN	g	minimum surface temperature	K]
216	1	WIMAX	g	maximum 2m-wind speed	[m/s]
217	1	TOPMAX	g	maximum convective cloud tops	[m]
218	1	SNMEL	g m	snow melt	[m/s]
220	1	TSLIN	g m	land: residual surface heat budget sea-ice: conductive heat flux	[W/m**2] [W/m**2]
221	1	DSNAC	g m	snow depth change	[m/s]
222	19	ALWCAC	g m	liquid water content	[kg/kg]
223	19	ACLCAC	g m	cloud cover	[fract.]
------------------		**clear/overcast sky diagnostics (method I)**			-----------------------
224	1	SRAD0F	g m	top solar radiation in clear-sky areas	[W/m**2]
225	1	TRAD0F	g m	top thermal radiation in clear-sky areas	[W/m**2]
226	1	ACCNTF	g	number of clear-sky events	
227	1	SRAD0C	g m	top solar radiation in overcast areas	[W/m**2]
228	1	TRAD0C	g m	top thermal radiation in overcast areas	[W/m**2]
229	1	ACCNTC	g	number of overcast events	
--					
230	1	QVI	g m	vertically integrated specific humidity	[kg/m**2]
231	1	ALWCVI	g m	vertically integrated liquid water cont.	[kg/m**2]

Table 10 Code numbers of the ECHAM-Models (Sheet 5 of 6)

code	levels	internal name	type	variable	unit
232	1	GLAC	g	glacier mask	[0: no, 1: yes]
241	19	XT(1)	s	tracer gas	
. .					
255	19	XT(15)	s	tracer gas	
259	19		g	windspeed $((u^{**}2+v^{**}2)^{1/2})$	
260	1		g m	total precipitation (142+143)	
261	1		g m	total top radiation (178+179)	
262	1		g m	total surface radiation (176+177)	
263	1		g m	net surface heat flux (146+147+176+177-220-C*218) where C = (Latent heat of fusion) * (density of water)	
264	1		g m	total surface water (142+143+182-160-221)	
265	1		g m	top solar cloud forcing (**method I**) (178-224)	
266	1		g m	top thermal cloud forcing (**method I**) (179-225)	
267	1		g m	total top cloud forcing (**method I**) (178+179-224-225)	
268	1		g m	atmospheric solar radiation (178-176)	
269	1		g m	atmospheric thermal radiation (179-177)	
270	1		g m	total atmospheric radiation (178+179-176-177)	
271	1		g m	surface runoff (**only ECHAM1 !**) (142+143+182-221)	
272	19		g	mass stream function	
273	1		g	$dp_s/d\lambda$ zonal derivative of surface pressure	
274	1		g	$dp_s/d\phi$ meridional derivative of surface pressure	
275	1		g m	precipitation - evaporation (142+143+182)	
-----------------		**cloud forcing (method II), not available in the standard model**			----------------------
364			g m	top solar cloud forcing (178-103)	
365			g m	top thermal cloud forcing (179-104)	
366			g m	top net cloud forcing (364+365)	
370			g m	surface solar cloud forcing (176-101)	
371			g m	surface thermal cloud forcing (177-102)	
372			g m	surface net cloud forcing (370+371)	

Table 10 Code numbers of the ECHAM-Models (Sheet 6 of 6)

code	levels	internal name	type	variable	unit
373			g m	atmospheric solar cloud forcing (364-370)	
374			g m	atmospheric thermal cloud forcing (365-371)	
375			g m	atmospheric net cloud forcing (373+374)	
376			g m	surface net radiation (clear sky) (101+102)	
377			g m	atmospheric solar radiation (clear sky) (103-101)	
378	1		g m	atmospheric thermal radiation (clear sky) (104-102)	
379	1		g m	atmospheric net radiation (clear sky) (377+378)	
--					
380	1		g m	planetary albedo[V] -(203/(178-203))*100.	[perc.]
381	1		g m	planetary albedo[V] -(105/(103-105))*100.	[perc.] (clear sky)
382	1		g m	surface albedo -(204/(176-204))*100.	[[perc.]

I. ECHAM1:
Over land ice (SN > 9.5 m or GLAC = 1) the snow melt is not calculated i.e. the snow gradually accumulates. In long term integrations with coupled models a suitable ice flow to the ocean (glacier calving) has to be introduced for maintaining the water budget.

II. Codes 157 (relative humidity) and 158 (tendency of surface pressure) are predefined but not calculated by the ***afterburner*** package.

III. ECHAM2/ECHAM3:
Due to a coding error the runoff (code 160) should not be used.
A "residual runoff" (code 271) may be applied for long term water budget studies instead.

IV. For description of soil temperatures see section 3.7.2

V. The planetary albedo (code 380 and 381, respectively for clear sky) equals zero, if the numerator is zero.
Calculate albedos only from temporal or eventually spatial mean fluxes.
The global mean results from the global mean of fluxes !

Table 11 Namelist POSTCTL (control of spectral variable write out)

Name	Type	Purpose	Default
LPPSPE	LOGICAL	True for write out all spectral variables except liquid water SX	TRUE
LPPD	LOGICAL	True for write out divergence SD	TRUE
LPPVO	LOGICAL	True for write out vorticity SVO	TRUE
LPPT	LOGICAL	True for write out temperature STP	TRUE
LPPP	LOGICAL	True for write out surface pressure STP(20)	TRUE
LPPQ	LOGICAL	True for write out specific humidity SQ	TRUE
LPPX	LOGICAL	True for write out liquid water content SX (For output of liquid water content request of code 222 is recommended)	FALSE

INDEX OF KEYWORDS