Report

of the Leibniz Institute of Atmospheric Physics in Kühlungsborn

KMCM - Kühlungsborn Mechanistic Circulation Model

- Documentation -

by
Erich Becker and Christoph Zülicke
with contributions from
Heiner Körnich, Stefan Lossow and many others

Documentation Version 1, Release 1, Update 10, Ch. Zülicke, 02 Sep 2010

relating to
Software Version 1, Release 1, E. Becker, November 2009
@ atlas.iap-kborn.de:/home/theo/kmcm/

Contents

1 In brief

KMCM (Kühlungsborn Mechanistic Circulation Model) is a so called Simple General Circulation Model (SGCM), describing atmospheric flow of dry air under physically idealized conditions. The basic numerical schemes are discretization in vertical direction according to Simmons & Burridge (1981), representation by spherical harmonics in horizontal direction (Machenhauer & Rasmussen, 1972) and semi-implicit time stepping (Asselin, 1972; Hoskins & Simmons, 1975). Realistic large-scale orography can be included due to the reference state reformulation of Simmons & Chen (1991). The thermal forcing of the model consists of temperature relaxation, prescribed tropical heat sources and self-induced heating in the middle latitudes. Orography and heating functions can independently be substituted by their zonal averages. KMCM includes a realistic boundary layer parameterization with the surface temperature being defined consistently with the thermal forcing. Horizontal diffusion is formulated consistently with elementary hydrodynamics (Becker, 2000). In this model version, no parameterization of inertia-gravity waves is used.

2 Numerics

2.1 Governing equations

The present simplified GCM is the *Kühlungsborn Mechanistic general Circulation Model* (*KMCM*) coded by the author. These governing equations are transformed using a terrain-following vertical hybrid coordinate η as follows. Pressure p is represented as a function of η and surface pressure p_s:

$$p(\eta; p_s) = a(\eta) + b(\eta)\, p_s \,. \tag{1}$$

The coefficients a and b must guarantee monotonic growth of p with η, as well as

$$p(\eta = 0; p_s) = 0 \quad \text{and} \quad p(\eta = 1; p_s) = p_s \,. \tag{2}$$

The flexibility of (1) is used to let surfaces of constant η correspond to σ-levels near the ground and to pressure levels at high altitudes.

The prognostic equations for horizontal vorticity ξ, horizontal divergence D, temperature T, and surface pressure p_s may be written as

$$\partial_t \xi = (\nabla \times \mathbf{f}) \cdot \mathbf{e}_z \tag{3}$$

$$\partial_t D = \nabla \cdot \mathbf{f} - \nabla^2 (\frac{\mathbf{v}^2}{2} + \Phi) \tag{4}$$

$$\mathbf{f} := \mathbf{v} \times (\, f + \xi \,)\, \mathbf{e}_z - \dot{\eta}\, \partial_\eta \mathbf{v} - \frac{R\,T}{p} \nabla p + \mathbf{H} + \mathbf{Z} \tag{5}$$

$$\partial_t T = -(\mathbf{v}\cdot\nabla + \dot{\eta}\,\partial_\eta\,)\,T + \frac{R\,T}{c_p\,p}\,\omega + Q + \mu_h + \mu_z + \frac{\epsilon_h + \epsilon_z}{c_p} \tag{6}$$

$$\partial_t p_s = -\int_0^1 \nabla\cdot(\,\partial_\eta\,p\,\mathbf{v}\,)\,d\,\eta\,. \tag{7}$$

These equations are formulated in geophysical spheric coordinates in terms of radius $r = a_e + z$ including the Earth radius a_e, geographical longitude λ and latitude ϕ. Hence, the horizontal gradient operator according to

$$dx = r\cos(\phi)\,d\lambda \tag{8}$$

$$dy = r\,d\phi \tag{9}$$

is given by

$$\nabla = \mathbf{e}_x\,\partial_x + \mathbf{e}_y\,\partial_y = \mathbf{e}_x\frac{\partial_\lambda}{r\cos(\phi)} + \mathbf{e}_y\frac{\partial_\phi}{r} \tag{10}$$

The prognostic equations (3)-(7) are completed by (1) and (32), and expressions for geopotential Φ, vertical velocity $\dot{\eta}$, and pressure velocity ω:

$$\Phi = \Phi_s + \int_\eta^1 \frac{R\,T}{p}\frac{\partial p}{\partial\tilde{\eta}}\,d\tilde{\eta} \tag{11}$$

$$\dot{\eta} = -\frac{1}{\partial_\eta\,p}\left(b\,\partial_t\,p_s + \int_0^\eta \nabla\cdot(\,\partial_{\tilde{\eta}}p\,\mathbf{v}\,)\,d\tilde{\eta}\right) \tag{12}$$

$$\omega = b\,(\,\mathbf{v}\cdot\nabla)\,p_s - \int_0^\eta \nabla\cdot(\,\partial_{\tilde{\eta}}\,p\,\mathbf{v}\,)\,d\tilde{\eta}\,. \tag{13}$$

Equation (11) follows from vertical integration of the hydrostatic approximation

$$\partial_\eta\,\Phi = -\frac{R\,T}{p}\,\partial_\eta\,p \tag{14}$$

with Φ_s denoting the orography. Vertical velocity, pressure velocity, and surface pressure tendency follow from integrations of the continuity equation

$$\partial_t\,(\,\partial_\eta\,p\,) + \nabla\cdot\,(\,\partial_\eta\,p\,\mathbf{v}\,) + \partial_\eta\,(\,\partial_\eta\,\dot{\eta}\,) = 0 \tag{15}$$

with respect to the kinematic boundary conditions

$$\dot{\eta} = 0 \quad \text{for} \quad \eta = 0 \quad \text{and} \quad \eta = 1\,. \tag{16}$$

The prognostic equations further contain quantities representing parameterized physical processes. These will be specified in chapter 3. The momentum diffusion and dissipation are formulated in terms of the horizontal and vertical shear stress tensors

$$\Sigma_h = \rho K_h \left(\left[(\nabla + \frac{\mathbf{e}_z}{r}) \circ \mathbf{v} \right] + [...]^T \right) \tag{17}$$

$$\Sigma_z = \rho K_z \left([\mathbf{e}_z \partial_z \circ \mathbf{v}] + [...]^T \right) \tag{18}$$

The corresponding horizontal and vertical momentum diffusion terms in (5) are

$$\begin{aligned} \mathbf{H} &= \frac{1}{\rho} (\nabla + \mathbf{e}_z \partial_z) \Sigma_h \qquad (19) \\ &\approx K_h \left(\nabla^2 \mathbf{v} + \nabla D + \frac{2\mathbf{v}}{r^2} \right) \\ \mathbf{Z} &= \frac{1}{\rho} (\nabla + \mathbf{e}_z \partial_z) \, \Sigma_z \qquad (20) \\ &= \frac{1}{\rho} \partial_z (\rho \, K_z \, \partial_z \mathbf{v}) \end{aligned}$$

and the related dissipation rates appearing in (6) are

$$\begin{aligned} \epsilon_h &= \frac{1}{\rho} (\Sigma_h \cdot (\nabla + \mathbf{e}_z \partial_z)) \cdot \mathbf{v} = K_h ||\Sigma_h||^2 \qquad (21) \\ &= K_h \left(2(D - \partial_y v)^2 + 2(\partial_y v)^2 + (\xi + 2\partial_y u)^2 \right) \\ \epsilon_z &= \frac{1}{\rho} (\Sigma_z \cdot (\nabla + \mathbf{e}_z \partial_z)) \cdot \mathbf{v} \qquad (22) \\ &= K_z \, (\partial_z \mathbf{v})^2 \end{aligned}$$

The approximations refer to the cases of constant density and diffusion.

In another diffusion model the trace of the shear stress tensor Σ_h vanishes

$$\Sigma_{h,0} = \Sigma_h - D \, (\mathbf{e}_x \circ \mathbf{e}_x + \mathbf{e}_y \circ \mathbf{e}_y) \tag{23}$$

corresponding to the friction and dissipation

$$\mathbf{H}_0 \approx K_h (\nabla^2 \mathbf{v} + \frac{2\mathbf{v}}{r^2}) \tag{24}$$

$$\epsilon_{h,0} = K_h \, ||\Sigma_{h,0}||^2 = K_h \, \left((D - \partial_y v)^2 + (\xi + 2\partial_y u)^2 \right) \tag{25}$$

The herewith used horizontal and vertical diffusivities (K_h and K_z) are to be specified in section 3.1. Heat fluxes are included with horizontal and vertical Prandtl numbers (Pr_h and Pr_z) - usually set to One - implying the following terms

$$\mu_h = \frac{K_h}{Pr_h} \nabla^2 T \tag{26}$$

$$\mu_z = \frac{1}{\rho} \partial_z \left(\rho \frac{T}{\Theta} \frac{K_z}{Pr_z} \partial_z \Theta \right) \tag{27}$$

The heat balance is subject to a bulk heating which has the following structure

$$Q = Q_{rad} + Q_{lat} \tag{28}$$

It consists of a radiation term (Q_{rad}, see section 3.4) and latent heating term (Q_{lat}, see section 3.5).

2.2 Vertical discretization

The model equations are prepared for numerical computation using an energy and angular momentum conserving finite-difference scheme introduced by Simmons & Burridge (1981) and modified by a reference state reformulation (Simmons & Chen, 1991).

First of all, the intermediate hybrid levels

$$0 = \eta_{1/2} < \eta_{3/2} < \ \dots \ < \eta_{n_{lev}-1/2} < \eta_{n_{lev}+1/2} = 1$$

must be fixed. The corresponding intermediate pressure levels

$$0 = p_{1/2} < p_{3/2} < \ \dots \ < p_{n_{lev}-1/2} < p_{n_{lev}+1/2} = p_s$$

are known from (1). Full pressure levels and centered pressure differences are defined as

$$p_l := (p_{l-1/2} + p_{l+1/2})/2 \quad \text{and} \quad \Delta p_l := p_{l+1/2} - p_{l-1/2} \quad \text{for} \quad l = 1 \dots n_{lev} \ . \tag{29}$$

In *KMCM*, n_{lev} is arbitrary, and the distribution of levels can be adjusted by a few parameters. This implemented with a recursive formulation

$$\eta_{l-1/2} = C_l \eta_{l+1/2} \tag{30}$$

$$C_l = \begin{cases} F_1 : n_{lev} \geq l > n_{lev} - l_1 + l_{12}/2 \\ F_2 + (F_1 - F_2)\frac{l - n_{lev} + l_1 + l_{12}/2}{l_{12}} : \dots \geq l > n_{lev} - l_1 - l_{12}/2 \\ F_2 : \dots \geq l > n_{lev} - l_1 - l_2 + l_{23}/2 \\ F_3 + (F_2 - F_1)\frac{l - n_{lev} + l_1 + l_2 + l_{23}/2}{l_{23}} : \dots \geq l > n_{lev} - l_1 - l_2 - l_{23}/2 \\ F_3 : \dots \geq l > n_{lev} - l_1 - l_2 - l_3 + l_{34}/2 \\ F_4 + (F_3 - F_4)\frac{l - n_{lev} + l_1 + l_2 + l_3 + l_{34}/2}{l_{34}} : \dots \geq l > n_{lev} - l_1 - l_2 - l_3 - l_{34}/2 \\ F_4 : \dots \geq l > 2 \end{cases} \tag{31}$$

with a height-dependent factor C_l. From these initial hybrid levels $\eta_{0,l}$, associated coefficients a and b according to (2) are constructed with the formula

$$a_l := p_{00} \frac{\eta_{0,l}}{2} \left(1 + \cos\left(\pi \eta_{0,l}\right)\right) \quad \text{and} \quad b_l := \frac{\eta_{0,l}}{2} \left(1 - \cos\left(\pi \eta_{0,l}\right)\right) . \tag{32}$$

Here, $p_{00} := 1013$ hPa corresponds to the mass of the atmosphere in case of zero orography. These coefficients are used for the further calculation of the pressure from (1).

The discretization method yields partial differential equations for the hybrid level representations of the prognostic variables $\xi_l(\lambda, \phi, t)$, $D_l(\lambda, \phi, t)$, $T_l(\lambda, \phi, t)$, and $p_s(\lambda, \phi, t)$, where $l = 1, 2, \ldots, n_{lev}$. The corresponding tendencies are computed by evaluating the right hand sides of (3)-(7) at full model levels. This includes diffusion, dissipation and gravity wave tendencies as well as bulk heating.

A finite-difference representation of the dynamical core in an angular-momentum and energy conserving way is as follows Simmons & Burridge (1981):

$$\left(\frac{RT}{p} \nabla p\right)_1 = \frac{RT_1}{\Delta p_1} \nabla \Delta p_1 \tag{33}$$

$$\left(\frac{RT}{p} \nabla p\right)_l = \frac{RT_l}{\Delta p_l} \left(\ln \frac{p_{l+1/2}}{p_{l-1/2}} \nabla p_{l-1/2} + \alpha_l \nabla \Delta p_l \right), \quad l = 2 \ldots n_{lev} \tag{34}$$

$$\Phi_1 = \Phi_s + \ln 2 \, RT_1 + R \sum_{n=2}^{n_{lev}} T_n \ln \frac{p_{n+1/2}}{p_{n-1/2}} \tag{35}$$

$$\Phi_l = \Phi_s + \alpha_l RT_l + R \sum_{n=l+1}^{n_{lev}} T_n \ln \frac{p_{n+1/2}}{p_{n-1/2}} , \quad l = 2 \ldots n_{lev} \tag{36}$$

$$\alpha_l = 1 - \frac{p_{l-1/2}}{\Delta p_l} \ln \frac{p_{l+1/2}}{p_{l-1/2}} , \quad l = 2 \ldots n_{lev} \tag{37}$$

$$\left(\frac{RT}{c_p p} \omega\right)_1 = -\frac{RT_1}{c_p \Delta p_1} \ln 2 \left\{ \Delta p_1 D_1 + \mathbf{v}_1 \cdot \nabla \Delta p_1 \right\} \tag{38}$$

$$+ \frac{1}{c_p} \mathbf{v}_1 \cdot \left(\frac{RT}{p} \nabla p\right)_1 \tag{39}$$

$$\left(\frac{RT}{c_p p} \omega\right)_l = -\frac{RT_l}{c_p \Delta p_l} \left(\ln \frac{p_{l+1/2}}{p_{l-1/2}} \sum_{n=1}^{l-1} \left\{ \Delta p_n D_n + \mathbf{v}_l \cdot \nabla \Delta p_n \right\} \right.$$

$$\left. + \alpha_l \left\{ \Delta p_l D_l + \mathbf{v}_l \cdot \nabla \Delta p_l \right\} \right) \tag{40}$$

$$+ \frac{1}{c_p} \mathbf{v}_l \cdot \left(\frac{RT}{p} \nabla p \right)_l , \quad l = 2 \ldots n_{lev}$$

$$\Big(\dot\eta \, \partial_\eta \, \{u,v,T\} \Big)_l \;=\; \frac{1}{2\,\Delta p_l} \Bigg((\,\dot\eta \partial_\eta \, p\,)_{l+1/2} \Big(\{u,v,T\}_{l+1} - \{u,v,T\}_l \Big) \qquad (41)$$

$$+ \;\; (\,\dot\eta \partial_\eta \, p\,)_{l-1/2} \Big(\{u,v,T\}_l - \{u,v,T\}_{l-1} \Big) \Bigg), \; l = 1 \ldots n_{lev}$$

$$(\,\dot\eta \, \partial_\eta \, p\,)_{1/2} \;=\; (\,\dot\eta \, \partial_\eta \, p\,)_{lev+1/2} \;=\; 0 \qquad (42)$$

$$(\,\dot\eta \, \partial_\eta \, p\,)_{l+1/2} \;=\; -\, b_{l+1/2} \, \partial_t \, p_s - \sum_{n=1}^{l} \Big\{ \, \Delta p_n \, D_n + \mathbf{v}_n \cdot \nabla \, \Delta p_n \Big\}, \; l = 1 \ldots n_{lev} - 1$$

$$\partial_t \, p_s \;=\; - \sum_{n=1}^{lev} \Big\{ \, \Delta p_n \, D_n + \mathbf{v}_n \cdot \nabla \, \Delta p_n \Big\} \, . \qquad (43)$$

This scheme has a specific problem. Let us assume a reference atmosphere at rest with $T = T_{ref}(p)$ and horizontally varying surface pressure $p_s = p_{sg}$ due to orography Φ_s. In this trivial example all tendencies must vanish. And this is, of course, the case for the continuous equations. However, in finite-difference form, the pressure gradient term and the geopotential gradient give rise to physically inconsistent tendencies, known as the *geostrophic wind error*. This error is defined as

$$f\,(\,v_l \, \mathbf{e}_x - u_l \, \mathbf{e}_y\,) = (\,\frac{R\,T}{p} \, \nabla \, p\,)_l + \nabla \, \Phi_l \qquad (44)$$

$$\text{in the case of} \quad T = T_{ref}(p) \quad \text{and} \quad \nabla \Phi_s = - \frac{R\,T_{ref}(p_{sg})}{p_{sg}} \, \nabla \, p_{sg} \, .$$

Owing to (33)-(37) the residuum (44) is generally different from zero and produces significant errors above the tropopause, as already mentioned by Simmons & Burridge (1981). This problem can be avoided by expanding (33)-(36) with respect to a reference state. One way to perform such a modification has been proposed by Simmons & Chen (1991) and is incorporated for instance in the climate model *ECHAM* (DKRZ, 1992). A similar method is applied in *KMCM*. Differences to previous methods arise from the choice of the reference temperature and from using p_s instead of $\ln p_s$ as a prognostic variable.

Let us define

$$\tilde T := T - T_{ref}(p) \qquad (45)$$

and rewrite the continuous formulations of the pressure gradient term and the gradient of the geopotential

$$\frac{R\,T}{p} \, \nabla \, p + \nabla \, \Phi \;=\; \frac{R\,\tilde T}{p} \nabla \, p + \nabla \left(\, \tilde\Phi_s + \int_\eta^1 \frac{R\,\tilde T}{p} \, \partial_{\eta'} \, p \, d\eta' \right) \qquad (46)$$

$$\tilde{\Phi}_s \;:=\; \Phi_s + \int\limits_{p_{00}}^{p_s} \frac{R\,T_{ref}(p)}{p}\,dp\,. \tag{47}$$

The mass of the atmosphere is defined by the mean surface pressure p_{ref} which, in the present model, is constant by definition. For zero orography we have $p_{ref} \equiv p_{00} = 1013$ hPa. When orography is included, we can implicitly define a reference surface pressure distribution $p_{sg}(\lambda, \phi)$ corresponding to $T = T_{ref}(p)$ by the root of the rhs of (47):

$$0 \;=\; \Phi_s(\lambda,\phi) + \int\limits_{p_{00}}^{p_{sg}(\lambda,\phi)} \frac{R\,T_{ref}(p)}{p} dp\,. \tag{48}$$

In *KMCM*, Eq. (48) is solved for p_{sg}, and p_{ref} is defined as the horizontal average of p_{sg}.

Obviously, both (46) and (47) vanish for the reference state. This suggests to use the rhs of (46) as a basis for finite differencing since the geostrophic wind error is zero by definition. Accordingly, (33)-(36) are reformulated in the following way:

$$\left(\frac{R\,T}{p}\nabla\,p\right)_1 \;=\; \frac{R\,\tilde{T}_1}{\Delta p_1}\,\nabla\,\Delta p_1 \tag{49}$$

$$\left(\frac{R\,T}{p}\nabla\,p\right)_l \;=\; \frac{R\,\tilde{T}_l}{\Delta p_l}\left(\ln\frac{p_{l+1/2}}{p_{l-1/2}}\,\nabla\,p_{l-1/2} + \alpha_l\,\nabla\,\Delta p_l\right), \quad l = 2\ldots n_{lev} \tag{50}$$

$$\Phi_1 \;=\; \tilde{\Phi}_s + \ln 2\;R\,\tilde{T}_1 + R\sum_{n=2}^{n_{lev}}\tilde{T}_n\,\ln\frac{p_{n+1/2}}{p_{n-1/2}} \tag{51}$$

$$\Phi_l \;=\; \tilde{\Phi}_s + \alpha_l\,R\,\tilde{T}_l + R\sum_{n=l+1}^{n_{lev}}\tilde{T}_n\,\ln\frac{p_{n+1/2}}{p_{n-1/2}}\;, \quad l = 2\ldots n_{lev}\,. \tag{52}$$

$$\begin{aligned} T_{ref}(p; p_{bot}, p_{trop}, p_{top}, T_{bot}, T_{trop}, T_{top}) \;&:=\; T_{bot}\,\zeta(p;\ldots) \\ \zeta(p;\ldots) \;&:=\; \zeta_0 + \frac{\zeta_1}{\varpi + p} + \frac{\zeta_2}{(\varpi+p)^2}\,. \end{aligned} \tag{53}$$

This profile can be adjusted with respect to 6 parameters, fixing T_{ref} at the pressure levels p_{bot}, p_{trop}, and p_{top}. These levels are assumed to correspond to bottom, tropopause, and some middle atmospheric pressure level. Then, ζ must satisfy

$$\begin{aligned} \zeta(p_{bot};\ldots) \;&=\; 1 \\ \zeta(p_{trop};\ldots) \;&=\; T_{trop}/T_{bot} \\ \partial_p\zeta(p_{trop};\ldots) \;&=\; 0 \\ \zeta(p_{top};\ldots) \;&=\; T_{top}/T_{bot}\,. \end{aligned} \tag{54}$$

The conditions (54) allow to eliminate the coefficients ζ_0, ζ_1, ζ_2, and ϖ. The procedure is described in Becker et al. (1997, appendix) for the vertical profile of the equilibrium temperature. The default parameter setup for T_{ref} is $p_{bot} = 1013$ hPa, $p_{trop} = 110$ hPa, $p_{top} = 0.1$ hPa, $T_{bot} = 280$ K, $T_{trop} = 210$ K and $T_{top} = 220K$. The corresponding temperature profile is shown in Fig. 32a. It may be compared to the reference profile used in *ECHAM* (DKRZ, 1992) Eq. (2.4.2.14)). Throughout the troposphere both reference states are quite similar. Above the tropopause, (53) yields reasonable temperatures. The *ECHAM* profile (Fig. 32b) has no tropopause and tends to zero for $p \to 0$. Hence, the geostrophic error above the tropopause can hardly be eliminated. Therefore *KMCM* utilizes the more realistic profile.

Owing to (53), Eq. (47) becomes

$$\tilde{\Phi}_s = \Phi_s + R\,T_{bot}\left(\zeta_0 \ln\frac{p_s}{p_{00}} + \left(\frac{\zeta_1}{\varpi} + \frac{\zeta_2}{\varpi^2}\right)\ln\frac{p_s\,(\,\varpi + p_{00}\,)}{p_{00}\,(\,\varpi + p_s\,)} + \frac{\zeta_2}{\varpi}\left(\frac{1}{\varpi + p_s} - \frac{1}{\varpi + p_{00}}\right)\right), \quad (55)$$

and the reference surface pressure $p_{sg}(\lambda, \phi)$ is found by numerically determining the root of the rhs of (55).

For model runs with 24 levels (L24), $p_s = p_{00}$ was used together with the following pressure levels in mb are from ground to top: 987, 903, 789, 684, 589, 503, 426, 357, 296, 242, 196, 155, 121, 92.3, 68.5, 49.3, 33.9, 22.1, 13.5, 7.43, 3.71, 1.86, 0.93, 0.31. For model runs with 60 hybrid layers (L60) the corresponding pressure levels in mb are: 987, 915, 825, 743, 667, 598, 535, 478, 426, 379, 336, 297, 262, 231, 203, 177, 155, 135, 117, 93.7, 68.4, 49.9, 36.5, 26.6, 19.4, 14.2, 10.4, 7.56, 5.52, 4.03, 2.94, 2.15, 1.57, 1.14, 0.835, 0.609, 0.445, 0.325, 0 .237, 0.173, 0.126, 0.0922, 0.0673, 0.0491, 0.0359, 0.0262, 0.0191, 0.0140, 0.0102, 0.00744, 0.00543, 0.00396, 0.00289, 0.00211, 0.00154, 0.00113, 0.00082, 0.00060, 0.00047, 0.00022.

2.3 Spectral presentation

In *KMCM* the spherical harmonics are defined as

$$Y_{n,m}(\lambda, \phi) := \begin{cases} \left(\frac{1}{\pi}\right)^{1/2} P_{n,0}(\sin\phi) & \text{for } m = 0 \\ \left(\frac{2}{\pi}\right)^{1/2} P_{n,m}(\sin\phi)\,\cos(m\lambda) & \text{for } m > 0 \\ \left(\frac{2}{\pi}\right)^{1/2} P_{n,|m|}(\sin\phi)\,\sin(|m|\lambda) & \text{for } m < 0 \end{cases} \quad (56)$$

with $$P_{n,m}(x) := \left((2n+1)\frac{(n-m)!}{(n+m)!}\right)^{1/2} \frac{1}{n!\,2^{n+1}}\,(1-x^2)^{m/2}\,\frac{d^{(n+m)}}{d\,x^{n+m}}\,(x^2-1)^n\,. \tag{57}$$

The normalization is such that

$$\int d\sigma\, Y_{n,m}\, Y_{n',m'} = \int_0^{2\pi} d\lambda \int_{-1}^{1} d\sin\phi\, Y_{m,m}\, Y_{n',m'} \;=\; \delta_{n,n'}\,\delta_{m,m'}\,. \tag{58}$$

Each spherical function has $2m$ Zeroes in the zonal direction and $n-|m|$ Zeroes in the longitudinal direction. Here, the summation includes the zonal wavenumber $m = -M, -M+1, ..., M$ and the total wavenumber $n = |m|, |m|+1, ..., N(|m|)$. Here, we denote M with n_z. For triangular truncation, N is set to n_z - the number of spherical harmonics in this case is $(n_z+1)^2$. It is possible to limit the number of total wavenumbers with n_t (trapezoidal truncation)

$$N = \begin{cases} |m| + n_t : |m| + n_t \leq n_z \\ n_z : n_z < |m| + n_t \end{cases} \tag{59}$$

By this approach the number of Zeroes in the longitudinal direction is limited to n_t. If done so, the total number of base functions is reduced to $(n_z+1)^2 - (n_z-n_t)^2$ (see Figure 1). In the program, the spherical harmonics are labeled with the index k scanning the total wavenumbers for Zero, positive and negative zonal wavenumbers according to

$$k(n,m) = \begin{cases} \sum_{n'=0}^{n} : m = 0 \\ \sum_{m'=0}^{m-1} \sum_{n'=m'}^{N(m')} + \sum_{n'=m}^{n} : 0 < m \\ \sum_{m'=0}^{n_z} \sum_{n'=m'}^{N(m')} + \sum_{m'=1}^{|m|-1} \sum_{n'=m'}^{N(m')} + \sum_{n'=|m|}^{n} : m < 0 \end{cases} \tag{60}$$

This works as shown in the following table 1.

Vorticity, divergence, temperature, and surface pressure are expanded in series of spherical harmonics as

$$\xi_l(\lambda,\phi,t) \;=\; \sum_{m,n} \xi_{l,n,m}(t)\, Y_{n,m}(\lambda,\phi) \tag{61}$$

$$D_l(\lambda,\phi,t) \;=\; \sum_{m,n} D_{l,n,m}(t)\, Y_{n,m}(\lambda,\phi) \tag{62}$$

$$T_l(\lambda,\phi,t) \;=\; \sum_{m,n} T_{l,n,m}(t)\, Y_{n,m}(\lambda,\phi) \tag{63}$$

$$p_s(\lambda,\phi,t) \;=\; p_{ref} + \sum_{m,n} p_{s\,n,m}(t)\, Y_{n,m}(\lambda,\phi)\,. \tag{64}$$

The amplitudes $\xi_{l\,0\,0}$, $D_{l\,0\,0}$, and $p_{s\,0\,0}$ are zero by definition. Eq. (64) differs from other spectral models where $\ln p_s$ instead of p_s is used as a prognostic

k	n	m
1	0	0
2	1	0
...	...	...
(n_t+1)	n_t	0
$(n_t+1)+1$	1	1
$(n_t+1)+2$	2	1
...	...	...
$(n_t+1)+(n_t+1)(n_z-n_t/2)$	n_z	n_z
$(n_t+1)+(n_t+1)(n_z-n_t/2)+1$	1	-1
$(n_t+1)+(n_t+1)(n_z-n_t/2)+2$	2	-1
...	...	...
$(n_t+1)+(n_t+1)(n_z-n_t/2)+(n_t+1)(n_z-n_t/2)=(n_z+1)^2-(n_z-n_t)^2$	n_z	$-n_z$

Table 1: Numbering scheme for the spherical harmonics: first the total wavenumbers for $m=0$ are gathered, then those for positive $m>0$ and finally those for negative $m<0$ zonal wavenumbers.

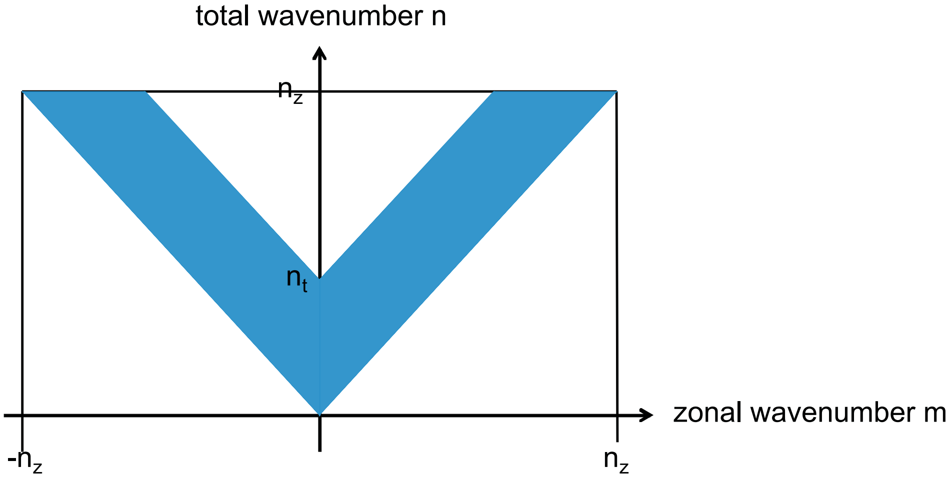

Figure 1: KMCM truncation scheme

variable (e.g., DKRZ (1992)) and hence the mass of the model atmosphere is treated as a prognostic variable.

Horizontal velocities are represented by

$$u_l = -\sum_{m,n} \frac{a_e^2}{n(n+1)} \left\{ D_{l,n,m}\, \partial_x Y_{n,m} - \xi_{l,n,m}\, \partial_y Y_{n,m} \right\} \tag{65}$$

$$v_l = -\sum_{m,n} \frac{a_e^2}{n(n+1)} \left\{ D_{l,n,m}\, \partial_y Y_{n,m} + \xi_{l,n,m}\, \partial_x Y_{n,m} \right\} \tag{66}$$

with ∂_x and ∂_y defined in (10). All horizontal derivatives are computed by using the analytical derivations of the spectral representations (63)-(66).

A set of ordinary differential equations for the spectral amplitudes is obtained by application of Galerkin's method at each full model layer. In order to simplify the resulting integrals we take advantage of the identities

$$\int d\sigma\, Y\, \frac{\partial_\lambda X}{a\cos\phi} = -\int d\sigma\, X\, \frac{\partial_\lambda Y}{a\cos\phi} \tag{67}$$

$$\int d\sigma\, Y\, \frac{\partial_\phi(\cos\phi\, X)}{a\cos\phi} = -\int d\sigma\, X\, \frac{\partial_\phi Y}{a} \tag{68}$$

$$\int d\sigma\, Y\, \nabla^2 X = \int d\sigma\, X\, \nabla^2 Y\,, \tag{69}$$

which are valid for arbitrary spherical functions X and Y. Our final model equations can be written as:

$$\dot{\xi}_{l,n,m} = \int d\sigma \Big(\mathbf{f}_l \times \nabla Y_{n,m} \Big) \cdot \mathbf{e}_z\,,\ l = 1 \ldots n_{lev} \tag{70}$$

$$\dot{D}_{l,n,m} = -\int d\sigma \Big(\mathbf{f}_l \cdot \nabla Y_{n,m} + (\frac{\mathbf{v}^2}{2} + \Phi_l\,)\, \nabla^2 Y_{n,m} \Big)\,,\ l = 1 \ldots n_{lev} \tag{71}$$

$$\dot{T}_{l,n,m} = \int d\sigma \Big(-\vec{v}_l \cdot \nabla T_l - (\dot{\eta}\partial_\eta T)_l + (\frac{RT}{c_p p}\omega)_l + Q_l + \mu_{z\,l} + \mu_{h\,l} + \frac{\epsilon_{z\,l} + \epsilon_{h\,l}}{c_p} \Big) Y_{n,m}\,,\ l = 1 \ldots n_{lev} \tag{72}$$

$$\dot{p}_{s\,n,m} = \int d\sigma\, \partial_t p_s\, Y_{n,m}\,. \tag{73}$$

The integrals on the rhs of (70)-(73) are computed by standard methods, i.e. by Gaussian quadrature with respect to $\sin\phi$ and discrete Fourier transform in longitudinal direction. Traditionally, the distribution of grid points is adapted such that the integrals are exact if the integrants are polynomials of third order in $Y_{n\,m}$. This assumption is equivalent to the condition that the tendency equations are of second order in the prognostic variables. Technically, the number zonal points (or Riemann summands) should be

$$i_{Rie} \geq 3\max(m) + 1 = 3n_z + 1 \tag{74}$$

while the number of (Gauss) latitudes

$$k_{Gau} \geq \frac{3\max(n) + 1}{2} = \frac{3n_z + 1}{2} \tag{75}$$

In each model layer l, the horizontally averaged kinetic energy per unit mass can be written as

$$\frac{1}{4\pi} \int d\sigma\, \frac{1}{2} \mathbf{v}_l^2 = \frac{a_e^2}{8\,\pi} \sum_{n,m} \frac{1}{n(n+1)}\, (\, \xi_{l,m,n}^2 + D_{l,m,n}^2\,)\,.$$

Accordingly, the kinetic energy per unit mass for total wave number n and for a particular model layer l yields

$$K_l(n) := \frac{a_e^2}{8\,\pi\, n(n+1)} \sum_{m=-n}^{n} (\,\xi_{l,m,n}^2 + D_{l,m,n}^2\,)\,. \tag{76}$$

The notation of T42 refers to a truncation wavenumber $n_z = 42$ which corresponds to an approximate grid size of $\Delta y \propto$40000 km/$(2n_z)$ =476 km. For the proper evaluation of integrals, the choice of Riemann summands $i_{Rie} = 128$ and Gauss latitudes $k_{Gau} = 64$ is recommended.

2.4 Time integration

For each spectral mode n, m the final model equations can formally be written as

$$\dot{\mathbf{y}}_{n,m} = \mathbf{T}_{n,m} \tag{77}$$
$$\mathbf{y}_{n,m}^T := (\,\xi_{1,n,m}, \ldots, \xi_{lev,n,m},\ D_{1,n,m}, \ldots, D_{lev,n,m},\ T_{1,n,m}, \ldots, T_{lev,n,m},\ p_{s,n,m}\,) \tag{78}$$

For $n > 0$ the tendency vector $\mathbf{T}_{n\,m}$ can be expanded as

$$\mathbf{T}_{n,m} = \mathsf{A}_n\, \mathbf{y}_{n,m} + \mathbf{T}'_{n,m} \tag{79}$$

$$\text{with} \qquad \mathsf{A}_n := \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & n(n+1)\,\mathsf{a}_1 \\ 0 & \mathsf{a}_2 & 0 \end{pmatrix}. \tag{80}$$

Here, $\mathbf{T}'_{n\,m}$ represents all spectral tendencies owing to Coriolis force, nonlinearities, diffusion, gravity waves, and diabatic heating. The Matrix A_n describes the buoyancy oscillations of a horizontally uniform reference state, i.e. the internal gravity waves in a corresponding linearized model version without Coriolis force and without all so-called physical parameterizations. These gravity modes separate in spectral space and degenerate with respect to the zonal wave number m. $\mathsf{A_n}$ is zero up to the coupling between $(\,D_{1,n,m}, \ldots, D_{n_{lev}\,n,m}\,)$ and $(\,T_{1\,n,m}, \ldots, T_{n_{lev}\,n,m},\, p_{s\,n,m}\,)$. The matrixes a_1 and a_2 depend on the reference state $T_{ref}(p)$ and on the distribution of model layers only (see Simmons & Burridge (1981), appendix). In *KMCM*, we use the reference state shown in Fig. 32a. In *ECHAM*, an isothermal reference state is employed.

Following Hoskins & Simmons (1975), time stepping is performed using the semi-implicit leapfrog scheme. This scheme must be completed by a time filter (Asselin, 1972) in order to damp the computational modes. This so-called semi-implicit method gains its efficiency from 'stretching' the oscillatory terms. This is achieved by application of the implicit leapfrog scheme to the linear terms and the explicit scheme to the remainder. Dropping the wave number indices $n\,m$, introducing the time index i, and denoting the time step by Δt, we get

$$\frac{\mathbf{y}_{i+1} - \mathbf{y}_{i-1}}{2\,\Delta t} = \mathsf{A}\,\frac{\mathbf{y}_{i+1} + \mathbf{y}_{i-1}}{2} + \mathbf{T}'_i\,. \tag{81}$$

Solving (81) for $\mathbf{y}_{i+1}$ and introducing the time filter yields

$$\mathbf{y}_{i+1} = (\mathsf{E} - \Delta t\,\mathsf{A})^{-1}\,(\mathsf{E} + \Delta t\,\mathbf{A})\,\tilde{\mathbf{y}}_{i-1} + 2\,\Delta t\,(\mathsf{E} - \Delta t\,\mathsf{A})^{-1}\,\mathbf{T}'_i \tag{82}$$

$$\tilde{\mathbf{y}}_i = \mathbf{y}_i + \delta\,(\,\tilde{\mathbf{y}}_{i-1} - 2\,\mathbf{y}_i + \mathbf{y}_{i+1}\,)\,. \tag{83}$$

Here, the unit matrix is abbreviated as E, and δ is a filter parameter which is usually set equal to 0.1. The leapfrog scheme requires two foregoing time steps to perform the next. In *KMCM*, the first time step is computed from the initial condition $\tilde{\mathbf{y}}_0$ by performing one Eulerian time step followed by five semi-implicit time steps using $\Delta t/5$.

Application of the explicit leapfrog scheme to diffusion tendencies can cause numerical instabilities that require short time steps in order to be controlled by the time filter. On the other hand, numerical stability of diffusion tendencies would be guaranteed by using an implicit leapfrog time step for these terms. *KMCM* compromises between numerical efficiency and a comprehensible source code by applying an Eulerian step for time integration of all tendencies owing to diffusion, dissipation, and the IGW parameterization. In other words, in (82), $\mathbf{T}'_i$ contains diffusion, dissipation, and IGW tendencies from time $i-1$. The particular time steps used in the T42/L24 and T29/L60 model versions are 15 and 7.30 minutes. Alternatively to semi-implicit time stepping, the model can be integrated by explicit standard schemes like the Runge-Kutta or the Bulirsch-Stoer method for instance. However, these schemes require shorter time steps and are less efficient.

3 Physics

3.1 Planetary boundary layer

The parameterization of boundary layer mixing coefficients follows the local vertical diffusion scheme of Holtslag & Boville (1993). In *KMCM*, a slight modification has been introduced to the Richardson criteria in order to ensure that their derivatives are continuous. The vertical diffusivity consists of three parts

$$K_z = K_{z,mol} + K_{z,stat} + K_{z,dyn} \tag{84}$$

The molecular processes are parameterized as

$$K_{z,mol} = \left(\frac{1}{\nu} + \frac{1}{\nu_{max}}\right)^{-1} \tag{85}$$

including the molecular kinematic viscosity

$$\nu = \frac{\rho_{00}}{\rho} 1.33 \times 10^{-5} \mathrm{m}^2 \ \mathrm{s}^{-1} \tag{86}$$

and a limiting value of $\nu_{max} = 900 \ \mathrm{m}^2 \ \mathrm{s}^{-1}$

The static diffusion term $K_{z,stat}$ is a profile changing from the bottom value $K_{z,bot}$ to the top value $K_{z,top}$ in a certain height range

$$K_{z,stat} = K_{z,bot} + (K_{z,top} - K_{z,bot})\mathrm{Tf}_\eta(\eta_{K,z}, \Delta\eta_{K,z}) \tag{87}$$

with a smooth transition function $\mathrm{Tf} = 0 \rightarrow 1$

$$\mathrm{Tf}_\eta(\eta_*, \Delta\eta_*) = \cos^2\left(\frac{\pi}{2}\frac{Z - Z_*}{\Delta Z_*}\right) \ : \mathrm{Z}_* - \Delta\mathrm{Z}_* < \mathrm{Z} < \mathrm{Z}_* \tag{88}$$

in terms of pressure heights ($Z = -H\log(\eta)$ with $Z_{K,z} = -H\log(\eta_{K,z})$ and $\Delta Z_{K,z} = -H\log(\Delta\eta_{K,z})$

For $l = 1, \ldots, n_{lev}$, that is in the free air, the dynamic diffusion coefficient $K_{z,dyn}$ is defined on half model layers as

$$K_{z,dyn} = \left(\frac{1}{Ka\,z} + \frac{1}{l_{z,asy}}\right)^{-2} S_z F_z(Ri) \tag{89}$$

$$l_{z,asy} = l_{z,bot} + (l_{z,top} - l_{z,bot})\mathrm{Tf}_\eta(\eta_{K,z}, \Delta\eta_{K,z}) \tag{90}$$

$$S_z = \left((\partial_z \mathbf{v})^2 + S_{z,min}^2\right)^{1/2} \tag{91}$$

$$Ri = \frac{g\,(\,\partial_z\,\Theta\,)}{\Theta\ S_z^2} \tag{92}$$

$$F_z(Ri) = \begin{cases} (1 - R_1 Ri)^{1/2} & Ri < 0 \\ (1 + R_1/2Ri + R_2 Ri^2)^{-1} & Ri \geq 0 \end{cases} \tag{93}$$

Here, $Ka = 0.4$ is the van Kárman constant. The asymptotic mixing length $l_{z,asy}$ is allowed to vary over the same height region as the linear background profile. For the sake of numerical stability, a minimum shear $S_{z,min}$ has been introduced. R_1 and R_2 are parameters for the Ri-dependent stability corrections - they are usually set to 18 and 0, respectively.

3.2 Surface layer

For the lowermost model layer ($l = n_{lev}$) the following parameterizations are used: The surface (Rayleigh friction) coefficient C_v appearing in the surface momentum flux

$$(\rho K_z \partial_z \mathbf{v})_0 = C_v \rho_0 \mathbf{v}_0 \tag{94}$$

is composed of three terms

$$C_v = C_{v,ref} + U_0(F_{v,ref} + Cd) \tag{95}$$

The first term is a constant reference value $C_{v,ref}$, the second term a linear dependence on the surface wind

$$U_0 = \left((\mathbf{v}_0)^2 + (S_{z,min}\, z_0)^2\right)^{1/2} \tag{96}$$

in terms of a constant drag coefficient $F_{v,ref}$, and the third term includes a stability-dependent drag coefficient reading

$$Cd = Cd_{neu} F_0(Ri_0)\ , \quad Cd_{neu} := \left(\frac{Ka}{\ln\left((z_0 + z_r)/z_r\right)}\right)^2 \tag{97}$$

$$Ri_0 = \frac{g\, z_0\, (\,\Theta_0 - \Theta_s\,)}{\Theta_{lev}\, U_0^2} \tag{98}$$

$$F_0(Ri_0) = \begin{cases} 1 - R_1/2Ri_0\,/(\,1\,+\,75\,Cd_{neu}\,(|Ri_0|(z_0+z_r)/z_r\,)^{1/2}\,) & Ri_0 < 0 \\ (1 + R_1/2Ri_0 + R_2{Ri_0}^2)^{-1} & Ri_0 \geq 0\,. \end{cases} \tag{99}$$

Here, z_r is the roughness length which has a constant value of 10^{-3} m in the present model experiments. Cd_{neu} is the neutral drag coefficient.

With regard to the surface sensible heat flux

$$(\rho K_z \frac{T}{\Theta} \partial_z \Theta)_0 = C_T \rho_0 (\Theta_0 - \Theta_s) \tag{100}$$

the surface coefficient

$$C_T = C_{T,ref} + U_0(F_{T,ref} + \frac{Cd}{Pr_z}) \tag{101}$$

where a Prandtl number Pr_z has been used. The surface potential temperature (Θ_s) is related to the surface temperature (T_s) via $\Theta_s = T_s(p_{00}/p_s)^{R/c_p}$. An estimate of the surface temperature can be derived from the local balanced heating at the surface (see (28) with sections 3.4 and 3.5)

$$Q(p_s) \propto -\frac{T_s - T_e(p_s)}{\tau} + Q_m(p_s) + Q_c(p_s) \approx 0 \tag{102}$$

reading

$$T_s = T_e(p_s) + C_{T,s}(\tau(Q_m(p_s) + Q_c(p_s))) \tag{103}$$

It includes an empirical surface temperature factor ($C_{T,s}$) and condensational heating rates (Q_m and Q_c).

3.3 Horizontal diffusion

The horizontal diffusion consists of a static, dynamic and filtered part

$$K_h = K_{h,stat} + K_{h,dyn} + K_{h,fil} \tag{104}$$

which have the following properties:

The static horizontal diffusion $K_{h,stat}$ changes from the bottom value $K_{h,bot}$ to the top value $K_{h,top}$ over the height range $[\eta_{K,h}, \eta_{K,h} + \Delta\eta_{K,h}]$ according to

$$K_{h,stat} = K_{h,bot} + (K_{h,top} - K_{h,bot})\mathrm{Tf}_\eta(\eta_{K,h}, \Delta\eta_{K,h}) \tag{105}$$

The dynamic horizontal diffusion $K_{h,dyn}$ includes dependencies on the mixing length l_h, the stress tensor $\mathbf{\Sigma}$ and the Richardson number Ri

$$\begin{aligned} K_{h,dyn} &= l_h^2 \left(||\Sigma_h||^2 + S_{h,min}^2\right)^{1/2} (1 + C_{Ri}F_z(Ri)) && (106) \\ l_h^2 &= l_{h,bot}^2 + (l_{h,top}^2 - l_{h,bot}^2)\mathrm{Tf}_\eta(\eta_{l\,h}, \Delta\eta_{l\,h}) && (107) \\ & && (108) \end{aligned}$$

For scale-selective damping of high-frequency fluctuations a filter function has been set up like

$$K_{h,fil} = K_{fil,bot} + (K_{fil,top} - K_{fil,bot})\mathrm{Tf}_\eta(\eta_{K,fil}, \Delta\eta_{K,fil})\mathrm{Ff}_m(m_{fil,1}, \min(m_{fil,2}, n_z)) \tag{109}$$

It is changing from $K_{fil,bot}$ to $K_{fil,bot}$ above $\eta_{K,fil}$ over $\Delta\eta_{K,fil}$ and acts on zonal wavenumbers m between $m_{fil,1}$ and $m_{fil,2} < n_z$.

3.4 Radiation

As a surrogate for radiative processes, a heating term like

$$Q_{rad} = -\frac{T - T_e}{\tau} \tag{110}$$

is introduced in (28). It mimics the relaxation to a certain reference temperature $T_e(\phi, p)$ on a prescribed time scale.

The zonally symmetric equilibrium temperature

$$T_e(\phi, p) = \Xi(p)\,(B(\phi, p) + \Sigma(\phi, p)) \tag{111}$$

is constructed of three terms: The height-dependent expression

$$\Xi(p) = \Xi_0 + \frac{\Xi_1}{\Psi + p} + \frac{\Xi_2}{(\Psi + p)^2} \tag{112}$$

allows a smooth transition between the atmospheric layers. The technical parameters Ξ_0, Ξ_1, Ξ_2 and Ψ will be specified later in terms of physical parameters.

The other height-and-latitude-dependent term is

$$\begin{aligned} B(\phi,p) &= b_1 + b_2 \arctan\left(\frac{p - p_{jet}}{\Delta p_{jet}}\right) \times \\ &\times \left(\frac{1}{\pi}\arctan\left(\frac{-\sin^2(\phi_{jet})}{\sin^2(\Delta\phi_{jet})}\right) - \frac{1}{\pi}\arctan\left(\frac{\sin^2(\Delta\phi) - \sin^2(\phi_{jet})}{\sin^2(\Delta\phi_{jet})}\right)\right) \times \\ &\times \ \mathrm{Hu}(\phi,p)\, C_\Sigma(p) \end{aligned} \tag{113}$$

$$\sin^2(\Delta\phi) = \left(\sin(\phi) - \frac{p - p_{top}}{p_{bot} - p_{top}}\sin(\phi_{equat})\right)^2 \tag{114}$$

It includes two more technical parameters b_1, b_2 to be specified below and the following features: the tropospheric jet is associated with a vertical temperature gradient over Δp_{jet} at a pressure level of about p_{jet}. Its meridional position is given by ϕ_{jet}, its width by $\Delta\phi_{jet}$. The impact of solar heating can be adjusted with the latitude of the solar equator ϕ_{equat} which influences the jet position at pressure levels between p_{bot} and p_{top}. A "Huppelfunktion"

$$\mathrm{Hu}(\phi,p) = \begin{cases} 1 : p < p_{jet} \\ 1 + \frac{p - p_{jet}}{p_{bot} - p_{jet}} C_{Hu} \left(\frac{\sin(\phi)}{\sin(\phi_{Hu})}\right)^2 \exp\left(1 - \left(\frac{\sin(\phi)}{\sin(\phi_{Hu})}\right)^2\right) : p_{jet} \leq p \end{cases} \tag{115}$$

has been introduced to simulate a secondary temperature maximum at a latitude of about ϕ_{Hu} and a relative amplitude of C_{Hu}.

Summer-winter asymmetries are included with the function

$$\Sigma(\phi,p) = (\Sigma_{sum} - \Sigma_{win})\left(1 - \left(\frac{p_\Sigma}{p}\right)^{1/2}\right)_{>0} + \Sigma_{therm} \tag{116}$$

$$\begin{aligned} \Sigma_{sum}(\phi,p) &= T_{sum}\exp\left(-\frac{\ln^2(p)}{2\ln^2(\Delta p_{sum})}\right) \times \\ &\times \ \exp\left(-\frac{(1+\sin(\phi))^2}{2\sin^2(\Delta\phi_{sum})}\right) \end{aligned} \tag{117}$$

$$\begin{aligned} \Sigma_{win}(\phi,p) &= T_{win}\exp\left(-\frac{p}{\Delta p_{win}}\right) \times \\ &\times \ \left(\frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{\sin(\phi) - \sin(\phi_{win})}{\sin(\Delta\phi_{win})}\right)\right) \end{aligned} \tag{118}$$

$$C_\Sigma(p) = \begin{cases} 1 : \Sigma = 0 \\ 1 - \exp\left(-\frac{p}{p_{trop}}\right) : \text{else} \end{cases} \tag{119}$$

All influence of seasons drops for pressures below p_Σ. The summer season is characterized with a temperature T_{sum} of a layer extending down to p_{sum} where it drops over a range of Δp_{sum}. With respect to latitude, the temperature is increasing over a distance of $\Delta\phi_{sum}$ from the winter pole. The winter temperatures are of order T_{win}, relevant for pressures below Δp_{win} and latitudes crossing ϕ_{win} over $\Delta\phi_{win}$. The modifications in the thermosphere are formulated in terms of log-pressure coordinates $Z = \log(p_{bot}/p)$ as

$$\Sigma_{therm} = \begin{cases} -T_{therm}\, C_{therm}\, \mathrm{Tf}_{therm} \sin^2\left(\frac{\pi}{2}\frac{Z-Z_{MLT}}{Z_{therm}-Z_{MLT}}\right) : Z_{MLT} < Z < Z_{therm} \\ T_{therm}\left(\Delta Z^2 - C_{therm}\,\mathrm{Tf}_{therm}\right) : Z_{therm} < Z < Z_{therm} + \Delta Z_{therm} \\ T_{therm}\left(2\Delta Z_{therm}\Delta Z - \Delta Z_{therm}^2 - C_{therm}\,\mathrm{Tf}_{therm}\right) : Z_{therm} + \Delta Z_{therm} < Z \end{cases} \quad (120)$$

$$\mathrm{Tf}_{therm} = 1 + 0.2\left(\frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{\sin(\phi) - \sin(\phi_{win})}{\sin(\Delta\phi_{win})}\right)\right) \quad (121)$$

$$\Delta Z = Z - Z_{therm} \quad (122)$$

The temperature changes at the scale of T_{therm} start above the level of the mesosphere / lower-thermosphere fixed with $Z_{MLT} = 0.5\, Z_{therm} = 0.5\, \log(p_{bot}/p_{therm})$. The temperature starts to increase above Z_{therm} through a layer of thickness $\Delta Z_{therm} = 0.4\, Z_{therm}$ to a linear height profile. The changes in latitude are similar to those associated with the winter season. The relative impact of these changes can be regulated with the dimensionless parameter C_{therm}.

The technical parameters b_1 and b_2 are fixed with prescribed surface temperatures at the equator T_{equat} and the pole T_{pole}

$$T_{equat} = B(\phi = \phi_{equat}, p = p_{bot}) \quad (123)$$

$$T_{pole} = B(\phi = \phi_{equat} + \frac{\pi}{2}, p = p_{bot}) \quad (124)$$

The remaining technical parameters (Ξ_0, Ξ_1, Ξ_2 and Ψ) are derived from the following four conditions:

$$\Xi(p_{bot}) = 1 \quad (125)$$

$$\Xi(p_{top})\, B(0, p_{top}; \phi_{equat} = 0) = T_{top} \quad (126)$$

$$\Xi(p_{trop})\, B(0, p_{trop}; \phi_{equat} = 0) = T_{trop} \quad (127)$$

$$\partial_p\left(\Xi(p_{trop})\, B(0, p_{trop})\right) = 0 \quad (128)$$

The first condition (125) normalizes Ξ to unity at the bottom $p = p_{bot}$ in accordance with the settings in (123) and (124). The other conditions (126 and 127) fix the equatorial temperatures at the top and the tropopause with T_{top}, p_{top} and T_{trop}, p_{trop}. The fourth condition (128) expresses the condition of vanishing vertical heat flux at the equatorial tropopause.

The relaxation time scale is of order τ =17 days. It may also vary with height and is composed of three contributions:

$$\tau = \tau_{ref} + \tau_{strat} + \tau_{therm} \tag{129}$$

The reference relaxation time τ_{ref} is changing from a bottom value τ_{bot} to a top value τ_{to} according to

$$\tau_{ref} = \tau_{bot} + (\tau_{top} - \tau_{bot})\,\mathrm{Tf}_\eta(\eta_\tau, \Delta\eta_\tau) \tag{130}$$

The stratospheric slow-down proposed by Dunkerton is implemented as

$$\tau_{strat} = \Delta\tau_{strat}\,\mathrm{Df}_\eta(\eta_{strat}, \Delta\eta_{strat}) \tag{131}$$

The relaxation time increases with a Gauss-like location function around an equivalent height scale $Z = -H\log(\eta)$ like

$$\mathrm{Df}_\eta(\eta_*, \Delta\eta_*) = \exp\left(-\frac{1}{2}(\frac{Z - Z_*}{\Delta Z_*})^2\right) \; : 0 < \mathrm{Z} < \infty \tag{132}$$

A doubling of the relaxation time is assumed in the thermosphere with

$$\tau_{therm} = \tau_{top}\,\mathrm{Tr}_\eta(\eta_{therm}, \Delta\eta_{therm}) \tag{133}$$

The value of η_{therm} has been fixed with 3×10^{-7} corresponding to a pressure of 3×10^{-4} hPa.

3.5 Latent heating

The impact of latent heat release by condensation in the troposphere is modeled with a bulk heating term in (28)

$$Q_{lat} = Q_c + \frac{|\omega|\,\mathrm{He}(-\omega)}{\omega_m} Q_m + Q_{QBO} \tag{134}$$

It consists of a prescribed cumulus heating $Q_c(\lambda, \phi, p)$ in the deep tropics (Hou, 1993) together with a self-induced condensational heating in the mid-latitudes using a heating function $Q_m(\lambda, \phi, p)$ (Mak, 1994) and QBO-related wave-like term. Only uprising air masses are taken into account with the Heaviside function He; a reference pressure velocity ω_m is of the order of 40 hPa/d. The mid-latitude heating is composed of three centers (northern, southern and arbitrary) like

$$Q_m = Q_n + Q_s + Q_x \tag{135}$$

The structure of the single centers is determined with their height ($p_*, \Delta p_*$), longitude ($\lambda_*, \Delta\lambda_*$), latitude ($\phi_*, \Delta\phi_*$), orientation ($\alpha_*$) and a height-dependent shift into this direction ($\Delta\alpha_*$) of the following form

$$
\begin{aligned}
Q_*(\lambda, \phi, p) &= V_*(p) H_*(\lambda, \phi, \eta) && (136)\\
V_*(p) &= Q_* \exp\left(-\frac{1}{2}\left(\frac{p - p_*}{\Delta p_*}\right)^2\right) \\
H_*(\lambda, \phi, \eta) &= \cos^2\left[\frac{\pi}{2}\left(\left(\frac{\cos(\alpha_*)\,\lambda' + \sin(\alpha_*)\,\phi'}{\Delta\lambda_*}\right)^2 + \left(\frac{\cos(\alpha_*)\,\phi' - \sin(\alpha_*)\,\lambda'}{\Delta\phi_*}\right)^2\right)\right] \\
\lambda' &= \lambda - \lambda_* - 2\,\Delta\alpha_*\,(1 - \eta)\cos(\alpha_*) \\
\phi' &= \phi - \phi_* - 2\,\Delta\alpha_*\,(1 - \eta)\sin(\alpha_*)
\end{aligned}
$$

Note, that $\cos[...]$ has to be positive.

The QBO-related heating term

$$
\begin{aligned}
Q_{QBO}(\lambda, \phi, p, t) &= Q_K(\lambda, \phi, p, t) + Q_R(\lambda, \phi, p, t) && (137)\\
Q_K(\lambda, \phi, p, t) &= Q_K \exp\left(-\frac{1}{2}\left(\frac{\ln(p_K) - \ln(p)}{\ln(\Delta p_K)}\right)^2 - \frac{\Omega\, a_e}{c_K}\phi^2\right) \times \\
&\times \sin\,(m_K\,\lambda - \omega_K\,t) && (138)\\
Q_R(\lambda, \phi, p, t) &= Q_R \exp\left(-\frac{1}{2}\left(\frac{\ln(p_R) - \ln(p)}{\ln(\Delta p_R)}\right)^2 - \Omega\, a_e\left(\frac{2\,\Omega}{a_e\,\omega_R^2} + \frac{1}{c_R}\right)\phi^2\right) \times \\
&\times \sin\,(m_R\,\lambda - \omega_R\,t) && (139)
\end{aligned}
$$

allows for a specification of a Kelvin and Rossby wave in terms of their heating rates (Q_K, Q_R), altitude range ($p_K \pm \Delta p_K, p_R \pm \Delta p_R$) along with their zonal wavenumber (m_K, m_R) and frequency (ω_K, ω_R).

4 Usage

Here are described the steps to make the program run and simulate certain test cases. The program flow chart is sketched in fig. 2. In the following it is described, which elements are needed the program setup (`RESOLUTION` and `gau3.data`), runtime parameters (`dc.steer`) and input data (`dc.bini`). The model output (`dc.new`) is described together with several afterburners.

For completeness, the integration procedure and basic variables are postered in fig. 3.

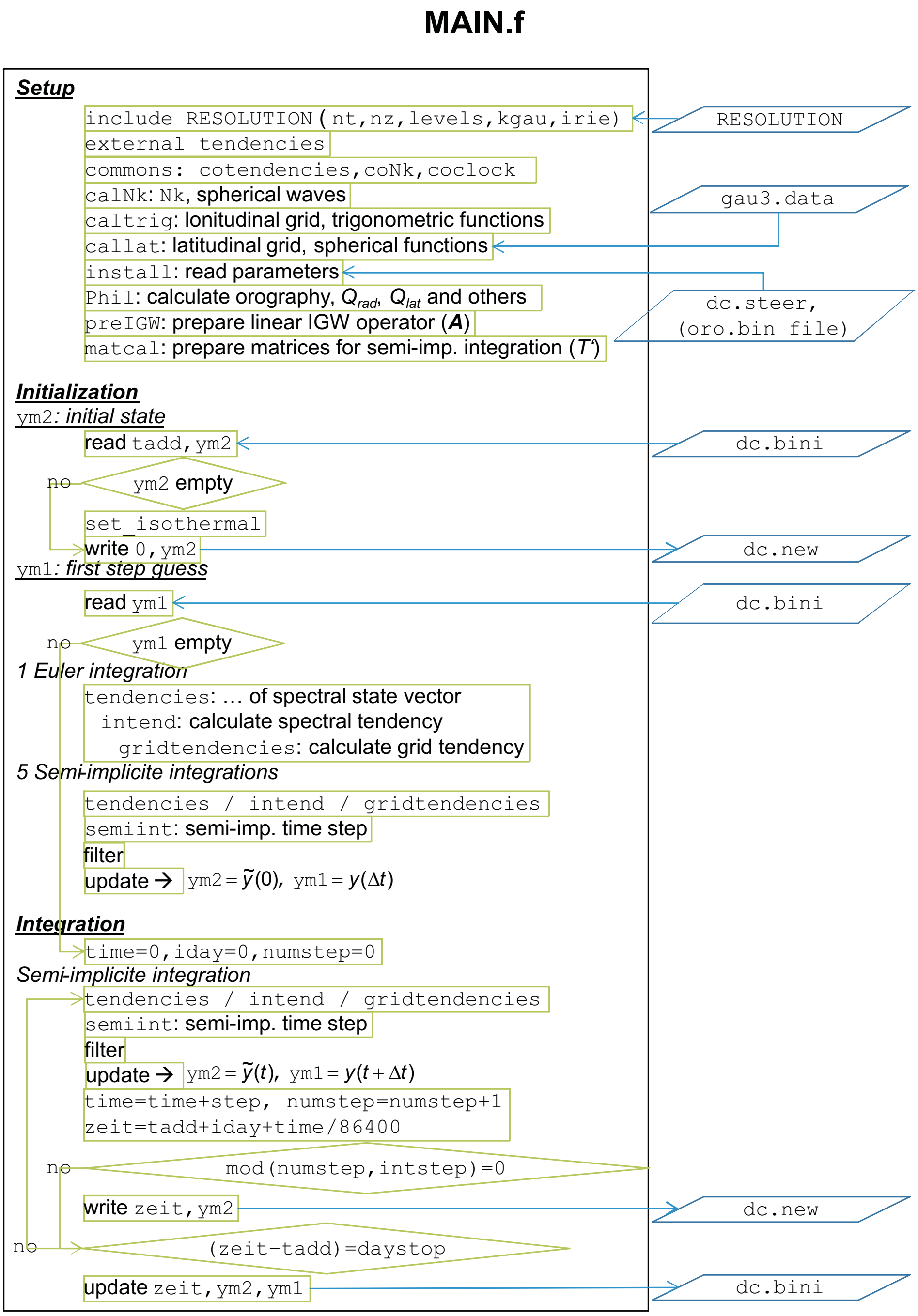

Figure 2: Program flow chart of MAIN.f

Semi-implicite Leapfrog Integration

Model equation $\quad d_t y(t) = T(t) = \mathbf{A}y(t) + T'(t)$

$\mathbf{A}$: linear IGW model $\rightarrow$ implicite leapfrog

T': nonlinear terms $\rightarrow$ explicite leapfrog

Digitized equation $\quad y(t+\Delta t) = (\mathbf{E} - \Delta t\mathbf{A})^{-1}(\mathbf{E} + \Delta t\mathbf{A})\tilde{y}(t-\Delta t) + 2\Delta t(\mathbf{E} - \Delta t\mathbf{A})^{-1}T'(t)$

Asselin filter $\quad \tilde{y}(t) = y(t) + \delta\left(y(t+\Delta t) - 2y(t) + \tilde{y}(t-\Delta t)\right)$

Variables

`daystop`: # days to integrate	`dtime`: time step for initial `ym1`
`dydt`: spectral tendency vector T	`filter`: filter constant (δ)
`iday`: # days integrated so far	`intstep`: # steps between outputs
`irie`: # Riemann summands	`kgau`: # Gauss latitudes
`levels`: # levels	`modr`: # spherical functions
`Nk`: wavenumber pointer	`nrec`: record length (machine-dependent)
`numstep`: # steps integrated so far	`ny`: dimension of state vector (y)
`nz`: # zonal wavenumbers	`step`: time step in seconds
`switch`: control variable for tendencies	`tadd`: previously integrated days
`time`: time of day in seconds	`y`: spectral state vector y(t)
`ym1`: state vector one step ago $y(t-\Delta t)$	`ym2`: state vector two steps ago $y(t-2\Delta t)$
`zeit`: time in days since start	

Figure 3: Integration procedure and major variables in MAIN.f

4.1 Preparing

4.1.1 Set resolution (edit RESOLUTION)

Choose resolution of the horizontal grid and the number of vertical levels, i.e., go to `dclib` and edit the file `RESOLUTION` and change the parameters (see sct. 2.3 and 2.2)

RESOLUTION:

parameter instruction for `nt, nz, levels, irie, gkau`

`nt`: total wave number truncation n_t. Example: `42`

`nz`: maximum zonal wave number $n_z \geq n_t$. Example: `42`

`levels`: number of vertical levels n_{lev}. Example: `30`

`irie`: number of Riemann summands i_{Rie}. Example: `128`

`kgau`: number of Gaussian latitudes k_{Gau}. Example: `64`

For the tutorial example, the resolution file was dumped under the name `kmcm.res` reading

```
        parameter( nz=42, nt=42, levels=30 , irie=128, kgau=64 )
c       parameter( nz=63, nt=42, levels=30 , irie=192, kgau=64 )
c       parameter( nz=63, nt=63, levels=220 , irie=192, kgau=96 )
```

Comments:

- for runs with zonally symmetric atmospheres choose `irie` $= 1$
- `irie` and `kgau` should be preferably multiples of small prime numbers, i.e. multiples of 2 for a fast FFT
- `irie` and `nt` should be typically `irie` $>= 3 *$ `nz` $+ 1$ (see 74)
- `irie` and `kgau` should be typically related via `kgau` = `irie`/2 in 3d-simulations (see 75)

4.1.2 Calculate Gauss latitudes (run gau3cal.exe)

Compile and execute the routine `gau3cal.f` in order to generate the file `gau3.data`. (This step can be skipped if appropriate resolution has been already chosen for the previous run.) From fig. 4 you may take how the program flows. It takes its parameters from `RESOLUTION` and puts its output into `gau3.data`.

GAU3CAL output :

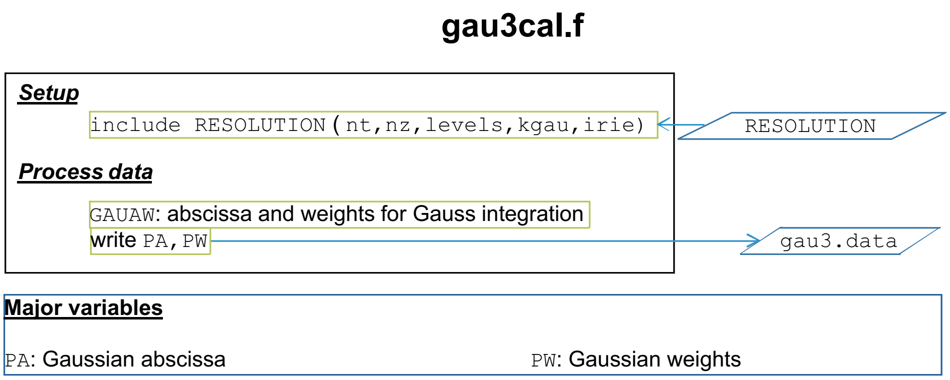

Figure 4: Program flow chart for `gau3cal.f`

`gau3.data` Gaussian latitudes and weights

`data x/pa/`: data block with Gaussian latitudes

`data w/pw/`: data block with Gaussian weights

You may run these steps with the following shell script `kmcm.gau3.csh`:

```
#!/bin/tcsh -f
# kmcm.gau3.csh
#
#####
# gau3cal.csh: running gau3cal (14 Apr 2010 CZ)
#####
#
set runName = 'kmcm'
set outFile = ${runName}.gau3.out
cp ${runName}.res ./dclib/RESOLUTION
#
make gau3cal.exe
#
cd dclib
./gau3cal.exe >&! $outFile
#
mv $outFile ..
cp gau3.data ../${runName}.gau3
#
exit
```

4.1.3 Generate orography (run `worldoro-V.exe`)

For the spectral truncation set with n_z and n_t (see 59) an binary orography file `worldV(nz)-(nt).oro.bin` is necessary. If it should not be present, it can be generated by `worldoro-V.exe`. It goes like shown in chart 5, reads a T106 orography from file `geosp.dat` and requires the following runtime input information:

WORLDORO-V input *(`world.in`)*

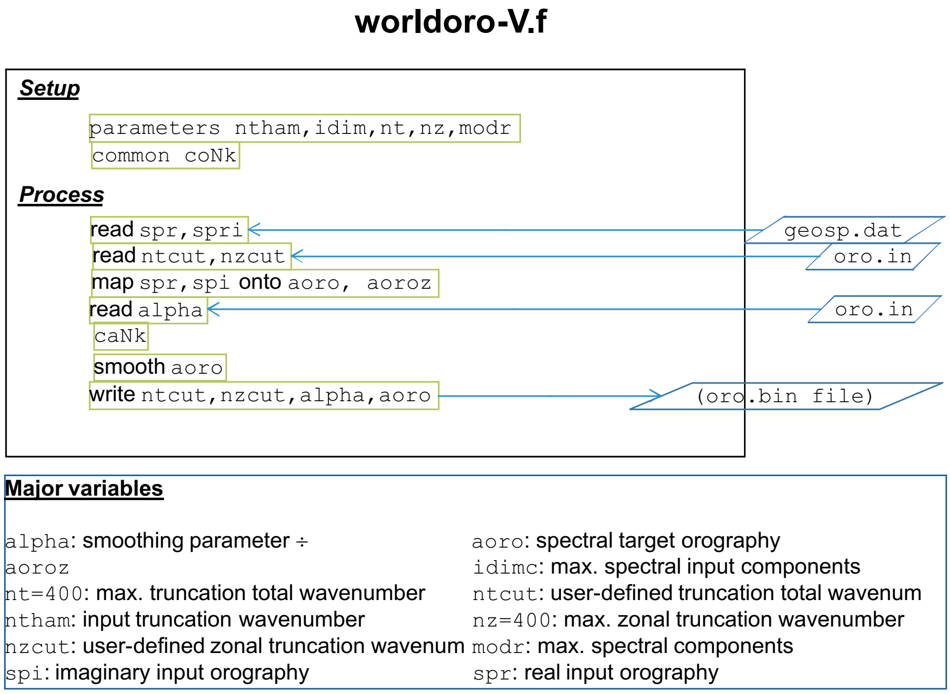

Figure 5: Program flow chart for `worldoro-V.f`

line 1 `ntcut, nzcut`. Cutoff parameters for n and m. Example: `42 42`

line 2 `alpha`: smoothing parameter acting like $\exp\left(-(n/\alpha)^2\right)$. Recommended values are 24 for T31, 30 for T42 and T63, or 35 for T85 and higher resolution, (`alpha < 1` will produce no smoothing). Example: `30`

An example is

```
42 42
30
```

The output has the following structure

WORLDORO-V output (`oro.bin file`)

`ntcut, nzcut, alpha`: dimensional and smoothing information

`aoro`: spectral orography

This procedure is managed by the script `kmcm.oro.csh`

```
#!/bin/tcsh -f
# kmcm.oro.csh
#
#####
```

```
# oro.csh: makes oro bin file (14 Apr 2010 CZ)
#####
#
set runName = 'kmcm'
set oroFile = ${runName}.oro
set inFile = ${oroFile}.in
set outFile = ${oroFile}.out
cp ${runName}.res ./dclib/RESOLUTION
#
make worldoro-V.exe
#
cat >  $inFile << EOF
42 42
30
EOF
#
./worldoro-V.exe < $inFile >&! $outFile
#
exit
```

4.2 Initializing and Running

4.2.1 Specify control parameters (Edit `dc.steer`)

Set control parameters, i.e. edit `dc.steer` in the project directory. All variables shall be given in prescribed order because the input is line-oriented. It is:

KMCM parameters (dc.steer file):

line 1 : The first line is ignored, but must be present. You can make your notes there.

line 2: `step` (integer) : Timestep Δt of the model in seconds. The timestep is a very sensitive parameter. A large value speeds up the model integration, but can lead to a model crash. In this case, the timestep has to be reduced. Then, the output interval has to be adjusted for daily output. Example: `360`

line 3: `daystop` (real) : Length of model integration in days. Example: `how many days (1`

line 4: `intstep (integer) IMEAN (integer)` - blank-separated : Output parameters. Example: `30 0`

`intstep`: time interval in number of time steps between two outputs. Example for daily output at `step` = 720: `intstep` = 120 (720s $*$ 120 = 86400s = 1 day).

`IMEAN`: 0 if output shall be a snapshot of the state vector at the output time step 1 if output shall be the mean over time interval between two outputs

line 5: `ofac` **(real),** `choro` **(string)** : Parameters for orography. Example: `0.1,worldV42-42.oro.bin`

`ofac`: scale of orography, 0.0 for no orography, scale can be used to switch on orography slowly

`choro`: name of the the file containing the orography. The orography file contains the spectral amplitudes of the amplitude and it must have the same spectral resolution as the present model. Different orography files can be found in the directory kmcm/oro, where the several files with the world orography at different resolutions can be found (eg. `worldT21.oro`). The files with the capital `Z` only contain the zonally symmetric component of the world orography (eg. `worldZ21.oro`). The files with the name `buhi` contain only one bump at the position of the Himalayas. The orography can be checked on the longitude-latitude grid with the postprocessor `stdgrads`.

line 6: `IO, HPr, Hshearminqm, IFH` **- comma-separated** : Parameters for horizontal diffusion. Example: `1 , 2.0 , 1e-10 , 1`

`IO`: choice of diffusion model (0: no diffusion at all; 1: traceless stress (23, 24, and 25); 2: nonzero stress trace (17, 19, and 21))

`HPr`: "horizontal Prandtl number" Pr_h (see 27)

`Hshearminqm`: minimum squared shear for horizonal diffusion $S^2_{h,min}$ ($\mathrm{m}^2/\mathrm{s}^2$)

`IFH`: flag to include frictional heating $\epsilon_h + \epsilon_z$ in temperature tendency (6) (0: exclude; 1: include)

line 7: `HLQbo,HLQto, HLQs, HLQds, Rifac` : Parameters for dynamic horizontal diffusion $K_{h,dyn}$ (106). Example: `-1.0e8 , -1.0e8, 8e-2 , 3e-1 , 8.6`

`HLQbo`: bottom value of squared horizontal mixing length $l^2_{h,bot}(\mathrm{m}^2)$

`HLQto`: top value of squared horizontal mixing length $l^2_{h,top}(\mathrm{m}^2)$

`HLQs`: start of transition region for squared horizontal mixing length η_{lh}

`HLQds`: width of transition region for squared horizontal mixing length $\Delta\eta_{lh}$

`Rifac`: Richardson factor C_{Ri}

line 8: `HKbo, HKto, HKs, HKds` : Parameters for static horizontal diffusion $K_{h,stat}$ (105). Example: `0e0 , -6.0e6 , 2e-3 , 1.0e-1`

`HKbo`: bottom value of static horizontal diffusion $K_{h,bot}$(m^2/s)

`HKto`: top value of static horizontal diffusion $K_{h,top}$(m^2/s)

`HKs`: start of transition region for static horizontal diffusion $\eta_{K\,h}$

`HKds`: width of transition region for static horizontal diffusion $\Delta\eta_{K\,h}$

line 9: `HKfbo, HKfto, HKfs, HKfds` : Parameters for the filtered horizontal diffusion $K_{h,fil}$ (109). Example: `-1.6e4, -0.60e4 , 0.9e-1 , 5.4e-1`

`HKfbo`: bottom value of filtered horizontal diffusion $K_{fil,bot}$(m^2/s)

`HKfto`: top value of filtered horizontal diffusion $K_{fil,top}$(m^2/s)

`HKfs`: start of transition region for filtered horizontal diffusion $\eta_{K,fil}$

`HKfds`: width of transition region for filtered horizontal diffusion $\Delta\eta_{K,fil}$

line 10: `N1, N2` : zonal wavenumber range where the filtered horizontal diffusion is switched on (see 109). Example: `30 , 60`

`N1`: start zonal wavenumber $m_{fil,1}$ for switch-on range of filtered horizontal diffusion

`N2`: end zonal wavenumber $m_{fil,2}$ for switch-on range of filtered horizontal diffusion

line 11: `z0, Pr, Tsfac, r1, r2` : Surface layer parameters. Example: `0.0015 , 1 , 0.40 , 18. , 0.`

`z0`: roughness length z_r(m) (see 97)

`Pr`: "vertical Prandtl number" Pr_z (see 101)

`Tsfac`: empirical factor for the estimation of the surface temperature $C_{T,s}$ (see 103)

`r1`: first parameter in surface Ri-parameterization R_1 (see 99)

`r2`: second parameter in surface Ri-parameterization R_2 (see 99)

line 12: `nubo, nuto, Vetas, Vdetas` : Parameters of the static vertical diffusion $K_{z,stat}$(m^2/s) (87). Example: `0.0 , 0.1 , 0.60, 0.80`

`nubo`: bottom value of static vertical diffusion $K_{z,bot}$(m^2/s)

`nuto`: top value of static vertical diffusion $K_{z,top}$(m^2/s)

`Vetas`: start of transition region for static vertical diffusion $\eta_{K,z}$

`Vdetas`: width of transition region for static vertical diffusion $\Delta\eta_{K,z}$

line 13: `vlbo , vlto, Vshearminq` : Parameters for dynamic vertical diffusion $K_{z,dyn}$ (m^2/s)) Example: `30. , 30. , 1e-12`

`vlbo`: bottom value of vertical mixing length $l_{z,bot}$ (m)

`vlto`: top value of vertical mixing length $l_{z,top}$ (m)

`Vshearminq`: minimum squared shear for vertical diffusion $S^2_{z,min}$ ($\mathrm{m}^2/\mathrm{s}^2$)

line 14: `cV [m/s], fV, cT [m/s], fT` : Parameters for surface Rayleigh friction coefficients for momentum and heat (C_v (95)) and C_T (101). Example: `0.000 , 0.0 , 0.0 , 0.0`

`cV`: constant surface velocity Rayleigh coefficient $C_{v,ref}$ (m/s)

`fV`: linear surface velocity Rayleigh coefficient $F_{v,ref}$

`cT`: constant surface temperature Rayleigh coefficient $C_{T,ref}$ (m/s)

`fT`: linear surface temperature Rayleigh coefficient $F_{T,ref}$

line 15: `relaxbo, relaxto [days], etas, detas` : Parameters for the reference relaxation time τ (130). Example: `16.0 , 7.0 , 0.001 , 0.015`

`relaxbo`: bottom value of the reference temperature relaxation time τ_{bot} (days)

`relaxto`: top value of the reference temperature relaxation time τ_{top} (days)

`etas`: start of transition region for reference temperature relaxation time η_τ

`detas`: width of transition region for reference temperature relaxation time $\Delta\eta_\tau$

line 16: `dunknose, dunk, dunks` : Parameters for the stratospheric correction of the temperature relaxation time τ_{strat} (131). Example: `24.0 , 0.095 , 0.75`

`dunknose`: Dunkertons correction of relaxation time $\Delta\tau_{strat}$ (hours)

`strat`: mean level of Dunkertons correction η_{strat}

`dunks`: width of Dunkertons correction $\Delta\eta_{strat}$

line 17: `OPA` : Parameter for the mid-latitude heating Q_m (134). Example: `40d0`

`OPA`: reference pressure velocity ω_m (mb/day)

line 18: `L1, F1, L12` : level distribution parameters containing information about the vertical model half levels in the first floor of the atmosphere. For details see `compost.f` in `dclib` and (31) . Example: `9 , 0.955 , 7`

`L1`: number of levels l_1

`F1`: factor F_1. We have `eta(kk-1) = F1*eta(kk)`. Setting `F1 < 0` equidistant levels will be produced. For `F1 > 1`, a and b values will be read from `ecmwf_lev.data`

`L12`: number of levels to be used for a smooth transition l_{12}

line 19: `L2, F2, L23` : .... for the next floor of the atmosphere. Example: `9 , 0.73 , 5`

`L2`: levels l_2

`F2`: factor F_2

`L23`: transition l_{23}

line 20: `L3, F3, L34` : ... further up. Example: `7 , 0.73 , 7`

`L3`: levels l_3

`F3`: factor F_3

`L34`: transition l_{34}

line 21: `F4` : top level factor F_4. Example: `0.60`

line 22 : (empty line)

line 23 : `Te-Parameter:` : parameters for the equilibrium temperature T_e. The first entry is the mean surface pressure in Pascal `pbo`, which is equivalent to the mass of the atmosphere. The mass of the atmosphere is therefore prescribed and not variable. Further details may be inferred from sct. 3.4 and the Fortran routines `readte`, `pretrelax` and `fte` in the file `relaxation.f` which are called from `Phil` to make `Te(ilon,l,jlat)`.

line 24 : `------------`

line 25 - 49 : `pbo, pto, feq, Teq, Tpo, fje, dsi, fh, dh, Ttr, Tto, ptr, pje, dp, Tsum, Twin, dpsum, dpwin, polni, dfisum, fiwin, dfiwin`

`pbo`: mean surface pressure p_{bot}(hPa). Example: `1013d2` `pto`: top pressure level p_{top}(hPa). Example: `.3d2`

`fequa`: latitude of equatorial temperature maximum $\phi_{equat}(^\circ)$. Example: `-6.0`

`Tequa`: surface temperature at equator T_{equat}(K). Example: `306d0`

`Tpo`: surface temperature at pole T_{pole}(K). Example: `251.0d0`

`fje`: latitude of tropospheric jet $\phi_{jet}(^\circ)$. Example: `36.0`

`dsi`: sharpness of tropospheric jet with respect to a sinus function $\sin^2(\Delta\phi_{jet})$. Example: `0.65`

`fh`: latitude of maximum "Huppelfunction" $\phi_{Hu}(^\circ)$. Example: `15d0`

`dh`: maximum of "Huppelfunktion" C_{Hu}. Example: `.0d0`

`Ttr`: tropopause temperature at equator T_{trop}(K). Example: `202.`

`Tto`: top level temperature at equator T_{top}(K). Example: `240d0`

`ptr`: height of tropopause at equator p_{trop}(hPa). Example: `100d2`

`pje`: height where the sign of $\partial_y T$ changes p_{jet}(hPa). Example: `197d2`

`dp`: associated width Δp_{jet}(hPa). Example: `195d2`

`Tsum`: characteristic summer temperature T_{sum} (K). Example: `55`

`Twin`: characteristic winter temperature T_{win} (K). Example: `97.`

`dpsum`: width of summer layer edge Δp_{sum} (hPa). Example: `75d2`

`dpwin`: width of winter layer edge Δp_{win} (hPa). Example: `130d2`

`polni`: height of seasonal influence p_Σ (hPa). Example: `0.15d2`

`dfisum`: width of summer transition region $\Delta\phi_{sum}$ ($^\circ$). Example: `50.`

`fiwin`: latitude of winter transition region ϕ_{win} ($^\circ$N). Example: `70.`

`dfiwin`: width of winter transition region $\Delta\phi_{win}$ ($^\circ$). Example: `6.`

`pthermo`: pressure level characteristic for thermosphere p_{therm} (hPa). Example: `0.001d2`

`Tthermo`: temperature characteristic for thermosphere T_{therm} (K). Example: `53d0`

`thermfac`: relative impact of thermospheric changes C_{therm}. Example: `0.3`

line 50 : (empty line)

line 51 : `concum-Parameter:` parameters for the latent heating fields Q_c and Q_m (so-called concum parameters). Q_c is denoted `Q0`, while Q_m is separated into two localized heating fields `Qn` and `Qs`. Only the zonal mean of these heating fields is taken into account, when the first longitude of each heating (`x0(1), xn(1), xs(1)`) is set to a negative value. Further details may be inferred from the routine `preqcm` and and others contained in the file `qcm.f` which are called from `Phil` to make `Qm(ilon,l,jlat)` and `Qc(ilon,l,jlat)`.

line 52 : `-----------------`

line 53 - 56 : `Q0, Qn, Qs, Qx`

`Q0`: maximum heating rate in the tropics Q_c(K/d). Example: `1.00`

`Qn`: maximum heating rate in northern midlatitudes Q_n(K/d). Example: `0.70`

`Qs`: maximum heating rate in southern midlatitudes Q_s(K/d). Example: `0.75`

`Qx`: maximum heating rate at arbitrary latitude Q_x(K/d). Example: `0.65`

line 57 : (empty line)

line 58 - 61 : `p0, sp0, f0, sf0`

`p0`: height of tropical maximum p_c(hPa). Example: `490d2`

`sp0`: height width of tropical maximum Δp_c(hPa). Example: `720d2`

`f0`: latitude of tropical maximum $\phi_c(^\circ)$. Example: `-6.0d0`

`sf0`: latitude width of tropical maximum $\Delta\phi_c(^\circ)$. Example: `17d0`

line 62 : (empty line)

line 63 - 66 : `pn, spn, fn, sfn`

`pn`: height of northern maximum p_n(hPa). Example: `960d2`

`spn`: height width of northern maximum Δp_n(hPa). Example: `260d2`

`fn`: latitude of northern maximum $\phi_n(^\circ)$. Example: `42.0`

`sfn`: latitude width of northern maximum $\Delta\phi_n(^\circ)$. Example: `29.`

line 67 : (empty line)

line 68 - 71 : `ps, sps, fs, sfs`

`ps`: height of southern maximum p_s(hPa). Example: `950d2`

`sps`: height width of southern maximum Δp_s(hPa). Example: `238d2`

`fs`: latitude of southern maximum $\phi_s(^\circ)$. Example: `42.5`

`sfs`: latitude width of southern maximum $\Delta\phi_s(^\circ)$. Example: `29.0`

line 72 : (empty line)

line 73 - 76 : `px, spx, fx, sfx`

`px`: height of arbitrary maximum p_x(hPa). Example: `980d2`

`spx`: height width of arbitrary maximum Δp_x(hPa). Example: `260d2`

`fx`: latitude of arbitrary maximum $\phi_x(^\circ)$. Example: `-40.0`

`sfx`: latitude width of arbitrary maximum $\Delta\phi_x(^\circ)$. Example: `26.0`

line 77 : (empty line)

line 78 - 84 : `x0(3), sx0(3), relzo`

`x0(1)`: longitude of first tropical maximum $\lambda_{c,1}(^\circ)$. Example: `42.`

`x0(2)`: longitude of second tropical maximum $\lambda_{c,2}(^\circ)$. Example: `158.`

`x0(3)`: longitude of third tropical maximum $\lambda_{c,3}(^\circ)$. Example: `307.`

`sx0(1)`: longitude width of first tropical maximum $\Delta\lambda_{c,1}(^\circ)$. Example: `47.`

`sx0(2)`: longitude width of second tropical maximum $\Delta\lambda_{c,2}(^\circ)$. Example: `83.`

`sx0(3)`: longitude width of third tropical maximum $\Delta\lambda_{c,3}(^\circ)$. Example: `34.`

`relzo`: relative zonally symmetric heating rate $R_{c,zon}$. Example: `0.14`

line 85 : (empty line)

line 86 - 92 : `xn(2), sxn(2), alfn(2), shitn`

`xn(1)`: longitude of first northern maximum $\lambda_{n,1}(^\circ)$. Example: `164.`

`xn(2)`: longitude of second northern maximum $\lambda_{n,2}(^\circ)$. Example: `164.`

`sxn(1)`: longitude width of first northern maximum $\Delta\lambda_{n,1}(^\circ)$. Example: `52.`

`sxn(2)`: longitude width of second northern maximum $\Delta\lambda_{n,2}(^\circ)$. Example: `52.`

`alfn(1)`: orientation of first northern maximum $\alpha_{n,1}(^\circ)$. Example: `5.`

`alfn(2)`: orientation of second northern maximum $\alpha_{n,2}(^\circ)$. Example: `5.`

`shitn`: height-dependent shift of northern maxima $\Delta\alpha_n(^\circ)$. Example: `40.`

line 93 : (empty line)

line 94 - 100 : `xs(2), sxs(2), alfs(2), shits`

`xs(1)`: longitude of first southern maximum $\lambda_{s,1}(^\circ)$. Example: `309.`

`xs(2)`: longitude of second southern maximum $\lambda_{s,2}(^\circ)$. Example: `309.`

`sxs(1)`: longitude width of first southern maximum $\Delta\lambda_{s,1}(^\circ)$. Example: `56.`

`sxs(2)`: longitude width of second southern maximum $\Delta\lambda_{s,2}(^\circ)$. Example: `56.`

`alfs(1)`: orientation of first southern maximum $\alpha_{s,1}(^\circ)$. Example: `20.`

`alfs(2)`: orientation of second southern maximum $\alpha_{s,2}(^\circ)$. Example: `20.`

`shits`: height-dependent shift of southern maxima $\Delta\alpha_s(^\circ)$. Example: `42.`

line 101 : (empty line)

line 102 - 108 : `xx(2), sxx(2), alfx(2), shitx`. Example:

`xx(1)`: longitude of first arbitrary maximum $\lambda_{x,1}(^\circ)$. Example: `-25.`

`xx(2)`: longitude of second arbitrary maximum $\lambda_{x,2}(^\circ)$. Example: `200.`

`sxx(1)`: longitude width of first arbitrary maximum $\Delta\lambda_{x,1}(^\circ)$. Example: `119.`

`sxx(2)`: longitude width of second arbitrary maximum $\Delta\lambda_{x,2}(^\circ)$. Example: `140.`

`alfx(1)`: orientation of first arbitrary maximum $\alpha_{x,1}(^\circ)$. Example: `-1.`

`alfx(2)`: orientation of second arbitrary maximum $\alpha_{x,2}(^\circ)$. Example: `-1.`

`shitx`: height-dependent shift of arbitrary maxima $\Delta\alpha_x(^\circ)$. Example: `30.`

line 109 : (empty line) Parameters for QBO excitation (see (137)). The forcing of Kelvin and Rossby waves (`AQK(lev,kgau)` and `AQR(lev,kgau)`) is set in `Phil` using routines from `qcm.f`.

line 110 - 111 : `QK, QR`

`QK`: heating rate for Kelvin wave Q_K(K/d). Example: `0.00d0`

`QR`: heating rate for Rossby-gravity wave Q_R(K/d). Example: `0.00d0`

line 112 : (empty line)

lines 113 - 114 : `pK, pR`.

`pK`: pressure level of Kelvin wave exciation p_K(Pa). Example: `480d2`

`pR`: pressure level of Rossby-gravity wave exciation p_R(Pa). Example: `480d2`

line 115 : (empty line)

line 116 - 117 : `spK, spR`. Example:

`spK`: pressure width of Kelvin wave excitation Δp_K(Pa). Example: `720d2`

`spR`: pressure width of Rossby-gravity wave excitation Δp_R(Pa). Example: `720d2`

line 118 : (empty line)

line 119 - 120 : `CK, CR`

`CK`: zonal phase speed of Kelvin wave c_K (m/s). Example: `27d0`

`CR`: zonal phase speed of Rossby-gravity wave c_R (m/s). Example: `-33d0`

line 121 : (empty line)

line 122 - 123 : `MK, MR`

`MK`: dimensionless zonal wavenumber for Kelvin wave m_K. Example: `2`

`MR`: dimensionless zonal wavenumber for Rossby-gravity wave m_R. Example: `4`

The parameters of the equilibrium temperature (T_e), relaxation time (τ) and latent heating fields (Q_c and Q_m) can be easily checked by using the afterburner `zobu` (see 4.4.3) and `GrADS` (see 4.4.1). If you are familiar with these programs, you may take a dummy model data file (`dc.new`) and follow this procedure:

1. Change parameters in `dc.steer`.
2. Run `zobu` with `dc.new` and `dc.steer`.
3. Run `GrADS`, open `dc.new` and display `Te`, `tau` and `qcm`.

Note that the steering file defines the setup of the model. For future reference it is advisable to save the steering file and the output of a certain model run with a special name

```
cp dc.steer (run).steer
cp dc.new (run).new
```

The first line of `dc.steer` can be used for comments.

For the tutorial example, the steering file was named as `kmcm.steer` and looks like

```
15/180, 18/150, 20/135, 22.5/120, 27/100, 30/90, 33.75/80, 36/75, 37.5/72
                 timestep (step) =  360
         how many days (itagstop) =    1
                instep and IMEAN =   30   0
    orography parameter and file =   0.1,worldV42-42.oro.bin
    IO, HPr, Hshearminq, and IFH = 1 , 2.0 , 1e-10 , 1
```

```
 HLQbo,HLQto, HLQs, HLQds, Rifac = -1.0e8 , -1.0e8, 8e-2 , 3e-1 , 8.6
     HKbo, HKto , HKs , HKds     =  0e0    , -6.0e6  , 2e-3 , 1.0e-1
     HKfbo, HKfto , HKfs , HKfds =  -1.6e4, -0.60e4 , 0.9e-1 , 5.4e-1
                       N1 and N2 =  30 , 60
          z0, Pr, Tsfac, r1, r2  =   0.0015 , 1 , 0.40 , 18. , 0.
       nubo, nuto, Vetas, Vdetas  =  0.0 , 0.1  , 0.60, 0.80
         vlbo , vlto, Vshearminq  =   30.  ,  30. , 1e-12
     cV [m/s], fV,  cT [m/s], fT =  0.000 , 0.0  , 0.0 , 0.0
relax_bo&_to [days], etas, detas =  16.0 , 7.0 , 0.001 , 0.015
          dunknose , dunk, dunks =  24.0 , 0.095 , 0.75
    heating parameter OPA [mb/d] =    40d0
   compost parameters: L1,F1,L12 =  9  , 0.955 , 7
                       L2,F2,L23 =  9  , 0.73  , 5
                       L3,F3,L34 =  7  , 0.73 ,  7
                          F4     =       0.60
```

<u>Te-Parameter:</u>
```
-------------
      mittlerer Bodendruck in Pascal  pbo   = 1013d2
                oberstes Druckniveau  pto   =  .3d2
  Breite des Bodentemperaturmaximums  feq   = -6.0
         Bodentemperatur am "Aquator  Tbo   = 306d0
              Bodentemperatur am Pol  Tpo   = 251.0d0
geogr. Breite deseTroposph"aren-Jets  fje   =  36.0
      Sch"arfe des Jets bezgl. sinus  dsi   = 0.65
Lage des Maximums der Huppelfunktion  fh    = 15d0
          Maximum der Huppelfunktion  dh    = .0d0
    Tropopausentemp. "uberm "Aquator  Ttr   = 202.
Temp. am obersten Druckn. "uberm "A.  Tto   = 240d0
 Lage der Tropopause "uberm "Aquator  ptr   = 100d2
         Vorzeichenwechsel von dT/dy  pje   = 197d2
                zugeh"origes Gewicht  dp    = 195d2
                                      Tsum =  55
                                      Twin =  97.
                                     dpsum =  75d2
                                     dpwin = 130d2
                                     polni =  0.15d2
                                    dfisum =  50.
                                    fiwin  = 70.
                                    dfiwin = 6.
                                   pthermo = 0.001d2
                                   Tthermo = 53d0
                                  thermfac = 0.3
```

<u>concum-Parameter:</u>
```
-----------------
       Q0     = 1.00       ! 1 Maximalheizraten in Tropen,
       Qn     = 0.70       ! 2 mittleren Breiten n,
       Qs     = 0.75       ! 3 mittleren Breiten s und
       Qx     = 0.65       ! 4 mittleren Breiten x

       p0     = 490d2      ! 5 H"ohenlage des Tropenmaximums [pa]
       sp0    = 720d2      ! 6 zugeh"orige H"ohenausdehnung [pa]
       f0     = -6.0d0     ! 7 Breite des Tropenmaximums [Grad]
       sf0    = 17d0       ! 8 zugeh"orige Breitenausdehnung [Grad]

       pn     = 960d2      ! 9  H"ohenlage des Nordmaximums [pa]
       spn    = 260d2      ! 10 zugeh"orige H"ohenausdehnung [pa]
```

```
fn     = 42.0        ! 11 Breite des Nordmaximums [Grad]
sfn    = 29.         ! 12 zugeh"orige Breitenausdehnung [Grad]

ps     = 950d2       ! 13 H"ohenlage des Nordmaximums [pa]
sps    = 238d2       ! 14 zugeh"orige H"ohenausdehnung [pa]
fs     =  42.5       ! 15 Breite des Suedmaximums [Grad]
sfs    =  29.0       ! 16 zugeh"orige Breitenausdehnung [Grad]

px     = 980d2       ! 17 H"ohenlage des Xmaximums [pa]
spx    = 260d2       ! 18 zugeh"orige H"ohenausdehnung [pa]
fx     = -40.0       ! 19 Breite des Xmaximums [Grad]
sfx    =  26.0       ! 20 zugeh"orige Breitenausdehnung [Grad]

x0 (1) =  42.        ! 21 geographische L"angen
x0 (2) = 158.        ! 22 und zugehoerige
x0 (3) = 307.        ! 23 Ausdehnungen
sx0(1) =  47.        ! 24 der drei
sx0(2) =  83.        ! 25 tropischen
sx0(3) =  34.        ! 26 Heizmaxima
relzo  =  0.14       ! 27 relative zonalsymmetr. Heizrate

xn (1) = 164.        ! 28 Lagen
xn (2) = 164.        ! 29 und
sxn(1) =  52.        ! 30 Ausdehnungen
sxn(2) =  52.        ! 31 bzgl. geogr. L"ange;
alfn(1)=   5.        ! 32 Drehwinkel
alfn(2)=   5.        ! 33 und
shiftn =  40.        ! 34 Verschiebung bei eta = 0.5

xs (1) = 309.        ! 35 Lagen
xs (2) = 309.        ! 36 und
ssn(1) =  56.        ! 37 Ausdehnungen
ssn(2) =  56.        ! 38 bzgl. geogr. L"ange;
alfs(1)=  20.        ! 39 Drehwinkel
alfs(2)=  20.        ! 40 und
shifts =  42.        ! 41 Verschiebung bei eta = 0.5:1

xx (1) = -25.        ! 42 Lagen
xx (2) = 200.        ! 43 und
ssx(1) = 119.        ! 44 Ausdehnungen
ssx(2) = 140.        ! 45 bzgl. geogr. L"ange;
alfx(1)=  -1.        ! 46 Drehwinkel
alfx(2)=  -1.        ! 47 und
shiftx =  30.        ! 48 Verschiebung bei eta = 0.5:1

QK     = 0.00d0      ! 0.8  Anregungsamplitude der Kelvin-Welle (K/d)
QR     = 0.00d0      ! 0.95 dito fuer Rossby-Schwerewelle (K/d)

pK     = 480d2       ! pressure level of Kelvin wave excitation (pa)
pR     = 480d2       ! dito for Rossby-gravity wave

spK    = 720d2       ! vertical extension of Kelvin level excitation (pa)
spR    = 720d2       ! dito for Rossby-gravity wave

CK     =  27d0       ! zonale Phasengeschw. der Kelvin-Welle (m/s)
CR     = -33d0       ! dito fuer Rossby-Schwerewelle (m/s)

MK     =     2       ! zonale dimenionslose Wellenzahl der Kelvin-Welle
```

```
MR      =     4      ! dito fuer Rossby-Schwerewelle
```

4.2.2 Generate initial conditions (run `inidc.exe`)

If the file `dc.bini` is not available, it may be produced with

$$\texttt{make inidc.exe} \tag{140}$$

and running it. The initial state is a thermally balanced purely zonal flow corresponding to the equilibrium temperature distribution T_e, where all asymmetries about the equator are omitted, i.e. `inidc.f` assumes always $\texttt{fequa} = 0, \texttt{Tsum} = 0$ and $\texttt{Twin} = 0$. The procedure is as follows: The primitive equations provide for zonally symmetric conditions

$$\frac{u_e^2}{a}\tan(\phi) + f u_e = -\partial_y \Phi_e \tag{141}$$

$$-\frac{R\,T_e}{p_e} = -\partial_p \Phi_e \tag{142}$$

which implies a zonal wind solution of

$$u_e(\phi,\, p) = u_e(\phi,\, p_0) + \cos(\phi)\left(-a\Omega + \left(a^2\,\Omega^2 + \int_{p_0}^{p} \frac{dp'}{p'}\, R\, \frac{\partial_{\sin(\phi)}\Phi_e(p')}{\sin(\phi)}\right)^{1/2}\right) \tag{143}$$

where the Coriolis parameter f has been expressed with the rotation frequency of the Earth $2\,\Omega\sin(\phi)$.

The original intension of `inidc.f` was to provide initial conditions for adiabatic life cycle experiments. In order let the baroclinic waves come early into play, wave excitations should be specified according to the "Standardwellenanregung". The initial state can be constructed with three parameters: zonal and total wavenumbers (m_{ini} and n_{ini}) together with a zonal-mean-to-wave-amplitude ratio r_{ini}

$$\xi_{ini}(m = m_{ini}, n = n_{ini}) = \frac{[\xi_e]}{r_{ini}} : \text{vorticity excitation} \tag{144}$$

$$D_{ini}(m = m_{ini}, n = n_{ini}) = \frac{1}{r_{ini}} : \text{divergence excitation} \tag{145}$$

The flow chart of `dcini.f` is in fig. 6.

After you have started `inidc.exe`, reply to the requests - the options are (you may answer all requests by returns):

INIDC input (inidc.in file):

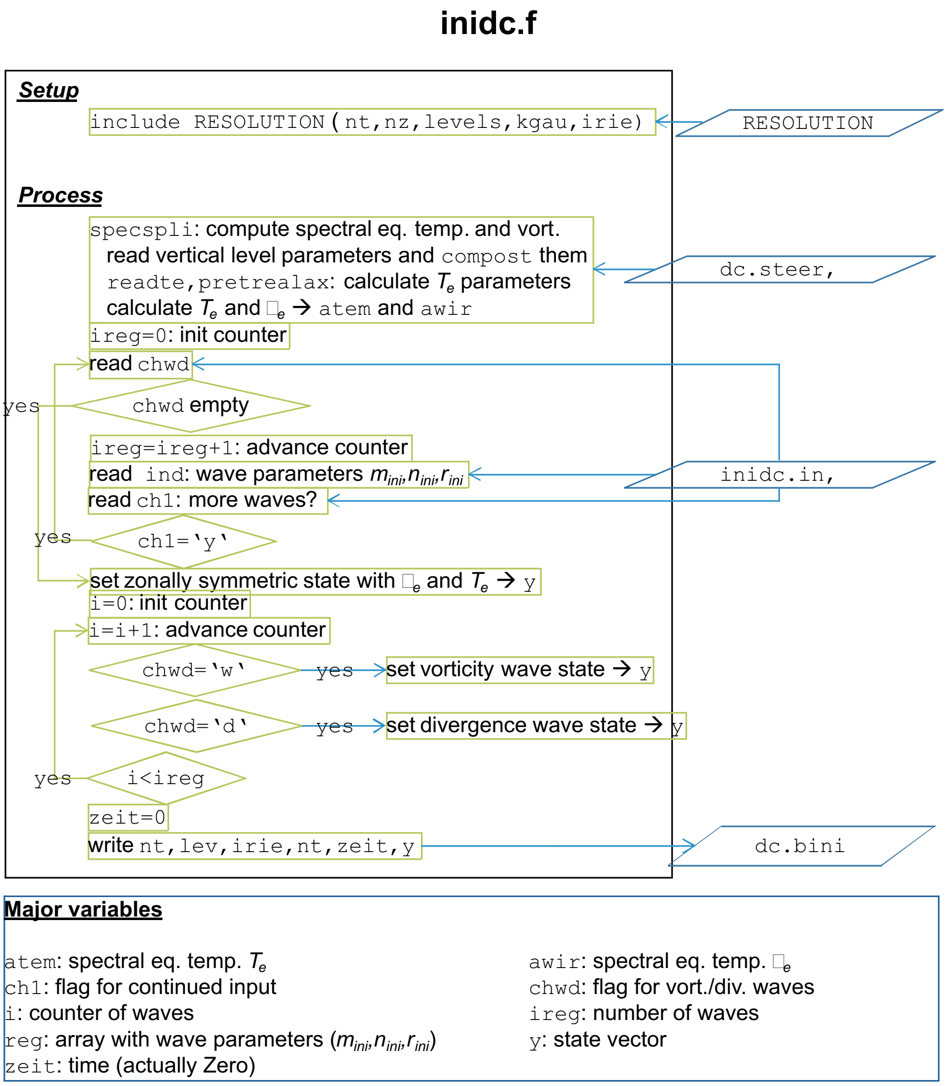

Figure 6: Program flow chart for `inidc.f`

line 1: `chwd` **(character)** : wave excitation (w: vorticity exciation, d: divergence excitation, enter: no wave excitation (default)). Example: `w`

line 2: `ind(1), ind(2), ind(3)` **(real)** : parameters of initial wave. Recommended choice is: `6, 7, 50`. You may provide parameters for up to `nt` waves. Example: `6, 7, 50`

`ind1`: zonal wavenumber of initial wave m_{ini}

`ind2`: total wavenumber of initial wave n_{ini}

`ind3`: zonal-mean-to-wave-amplitude-ratio of initial wave $r_{ini} = [A]/A^*_{ini}$

line 3: `ch1` **(character)** : continue with more wave parameters ('y': yes, enter: discontinue). If 'y' is given, lines 2 and 3 are repeated... until something else is given. Example: `n`

The resulting initial state vector is stored in `dc.bini`:

INIDC output:

`dc.bini` initial state vector for a model start.

`nt, levels, irie, nz`: model resolution

`zeit,y`: sample with time and spectral state vector

If the initial state vector shall be kept for another model run, copy `dc.bini` to a file with a different name (f.e. `(run).bini`), since the model will overwrite `dc.bini` after a completed model run with the state vectors needed for a restart.

All this is done by `kmcm.bini.csh` reading

```
#!/bin/tcsh -f
# kmcm.bini.csh
#
#####
# bini.csh: runs inidc.exe (14 Apr 2010 CZ)
#####
#
set runName = 'kmcm'
cp ${runName}.res ./dclib/RESOLUTION
cp ${runName}.steer ./dc.steer
cp ${runName}.gau3 ./dclib/gau3.data
set biniFile = ${runName}.bini
set inFile = ${biniFile}.in
set outFile = ${biniFile}.out
#
make clean
make inidc.exe >&! make.out
```

```
#
cat >  $inFile << EOF
w
6, 7, 50
n
EOF
#
./inidc.exe < $inFile >&! $outFile
#
mv dc.bini $biniFile
#
exit
```

4.2.3 Integrate the model (run `kmcm.exe`)

Make the model

$$\texttt{make kmcm.exe} \tag{146}$$

and run it. After completion of the run two output files will be produced:

KMCM output:

`dc.bini` containing the state vectors for a model restart in double precision. Important notice! If the restart conditions shall be kept, dc.bini should be copied to a file with a different name, since dc.bini will be overwritten after completion of the next run.

> `nt, levels, irie, nz`: model resolution
>
> `zeit,y`: sample with time and spectral state vector

`dc.new` containing the model output (state vector in single precision, as specified in dc.steer) Important notice! If model output shall be kept, dc.new should be copied to a file with a different name, since dc.new will be overwritten after completion of the next run.

> `nt, levels, irie, nz`: model resolution
>
> {`zeit,y` } : samples with time and spectral state vector

If there are older compilations present, it is advised to `make clean` before. These steps can be managed with `kmcm.new.csh`

```
#!/bin/tcsh -f
# kmcm.new.csh
#
#####
# kmcm.csh: runs kmcm.exe (14 Apr 2010 CZ)
#####
#
# set run name
set runName = 'kmcm'
cp ./${runName}.steer ./dc.steer
cp ./${runName}.res ./dclib/RESOLUTION
cp ./${runName}.bini ./dc.bini
```

```
cp ./${runName}.gau3 ./dclib/gau3.data
set outFile = ${runName}.new.out
#
# clean and make kmcm.exe
make clean
make kmcm.exe >&! make.out
#
# run kmcm.exe
./kmcm.exe >&! $outFile
#
# rename output
mv dc.new ${runName}.new
# mv dc.bini ${runName}.bini
#
exit
```

The program run for this one-day simulation 3 min 53 sec at one CPU of IAP's `atlas`.

A technical note: the standard case includes a discrete Fourier transformation in `intend.f`. If the user would like to use a Fast Fourier Transform as provided by the NAG library, for instance, the file `intend_nag.f` should be copied to `intend.f` and the `Makefile` should be modified to load the NAG library, too.

4.3 Changing

4.3.1 Constants (edit CONSTANTS)

To change model constants edit the file `CONSTANTS` in `dclib`, e.g. the radius of the planet `aerde`. This file contains the following items:

CONSTANTS:

`R`: molar gas constant of dry air R ($m^2/s^2/K$)). Example: `287.04d0`

`cp`: heat capacity at constant pressure c_p ($m^2/s^2/K$). Example: `1004d0`

`g`: gravitational acceleration g (m/s^2). Example: `9.81d0`

`p00`: sea surface pressure p_{00} (hPa). Example: `1013d2`

`aerde`: Earth radius a_e (m). Example: `6.3782d6`

`om`: rotational frequency of the Earth Ω (1/s). Example: `72.92d-6`

`CSUN`: solar constant C_{sun} (W/m^2). Example: `1370d0`

`SIGMA`: Stefan-Boltzmann constant σ ($W/m^2/K^4$). Example: `5.67d-8`

4.3.2 Equilibrium temperature (manipulate `relaxation.f`)

To change the temperature relaxation by new versions of T_e and the vertical profile of relaxation time τ, `relaxation.f` may be completely rewritten. The new T_e-parameters must then be defined in `dc.steer` (similar to the default version, but there may be fewer or more parameters). The routine `pretref.f` must be accordingly adjusted for the new T_e. In particular, the function `fTe.f` must be modified, since it is called by `Phil.f`. In addition the function `calrelaxd.f` for the new τ must be modified or the call to it in `Phil.f` must be canceled. The nice thing about it is, that all these changes, once properly done, are automatically carried forward into the postprocessing routines.

4.4 Analyzing

In the section the analysis of KMCM output is described. The procedure consists of two steps: first, application of an afterburner to obtain the desired data and second, their graphical presentation. The afterburners are written in FORTRAN - three of them are introduced in this section:

- `edtrans`: averaged and integrated data → time series $f(t)$
- `zobutrans`: zonal budgets → latitude-height sections $f(\phi, p, t)$
- `hetrans`: transient height-resolved data → three-dimensional fields $f(\lambda, \phi, p, t)$

All these documentations are supplemented with a practical example which all use the same graphics program. This is introduced first.

4.4.1 Display data with `GrADS`

The easiest way to display the KMCM data is the usage of the Grid Analysis and Display System (GrADS). All output from the model's postprocessors comes in the native GrADS format, which consists of an ASCII-control file (.ctl) and a binary data file (.grads). GrADS supports a wide range of file formats (netCDF, etc.), it is interactive, accepts scripts, can do a lot of standard manipulations like zonal mean, weighted area mean and field manipulations, and GrADS is free. Source code and binaries for many operating systems are available through their homepage (http://grads.iges.org/grads/).

In order to display the KMCM data, an installation of GrADS is recommendable. On SU linux GrADS is preinstalled, so there you only need to add the module ('module add GrADS'). On other linux or windows machines, the respective installation files (binaries and additional data files) have to be downloaded from the GrADS homepage.

Once GrADS is installed, one can get familiar with the program through the GrADS tutorial (http://grads.iges.org/grads/gadoc/tutorial.html) or with KMCM data from the sample run. Useful reference cards of the GrADS-commands and scripting language can be found in the model's documentation directory (GrADS.reference_card.pdf and GrADS.reference_card_scl.pdf). The most important commands will be reviewed here in short:

GrADS commands:

`gradsc` starts GrADS. Graphic output can be chosen either in portrait or landscape mode.

`open filename.ctl` opens the respective GrADS-control file.

`sdfopen filename.nc` opens a netCDF-file, eg. ERA40 reanalysis data.

`q file` (query file) displays the information on the file, like resolution and variables.

`q dims` (query dims) displays the current dimension settings.

`d variable` (display variable) plots the respective variable on the present grid settings. Example: d u

`define newvar= 5*variable1 + variable2` defines a new variable as a function of old variables. This new variable can also be displayed.

`define theta=T*pow(1013/lev,287/1004)` defines the potential temperature. Note that the definition is done only for the current dimension settings and not for the whole data set.

`c` (clear) clears the screen.

`reinit` reinitialize GrADS and restart from the beginning.

`set lon value1 (value2)` sets the longitude coordinate to a single value or to a range of values.

`set lat value1 (value2)` same for latitude

`set lev value1 (value2)` same for levels, where value is given as pressure in hPa.

`set t value1 (value2)` same for time, where values are timesteps of the data file.

`d ave(variable,lon=0,lon=360,-b)` displays the zonal average of a variable. The '-b' excludes double counting of the boundaries.

`d aave(variable,lon=0,lon=360,lat=-90,lat=90)` displays the global area-weighted average of a variable.

`d ave(variable,t=0,t=10)` displays the time average of a variable.

`d vint(ps,variable,top)` performs a mass-weighted vertical integral in hPa pressure coordinates, where *ps* is a map of the surface pressure, *variable* is the quantity which is integrated, and *top* is the constant upper limit of the integration.

`set cint value` sets the contour interval to value.

`set gxout shaded` sets screen output to shaded plots.

`set gxout contour` sets screen output to contour plots.

`draw title text` draws the text as a title of the figure.

`draw xlab text` draws the text as a label of the x-axis.

`draw ylab text` draws the text as a label of the y-axis.

`set vpage x1 x2 y1 y2` sets a virtual page in order to create several plots on a single page. x1, x2 and y1, y2 are the starting and ending coordinates on the GrADS output page. The output page size is 11x8.5 in landscape-mode and 8.5x11 in portrait-mode. Example: set vpage 0 5.5 0 8.5 (only left part of landscape output).

`draw string x y bla` prints the text *bla* at the coordinates x,y of the output page. Example: draw string 3 8.3 here is the top of the page (for landscape output).

`quit` quits GrADS.

`script: white` sets background to white.

`script: cbar` plots a colour bar for a shaded figure.

`script: snapshot` GrADS-script which plots a certain field at the current coordinates for one timestep after the other. After each timestep one can choose between printout, exit and next timestep.

`script: printout` GrADS-script which creates a black-white printout or graphics file (PS/EPS).

`script: printoutc` GrADS-script which creates a colour printout or graphics file (PS/EPS/GIF).

For longer calculations or plot command lists, it is advisable to write one's own GrADS-scripts. This means that you should write all GrADS commands into a file with the extension *.gs*. Each GrADS command needs a leading and trailing single quotation mark, eg. *'set lev 300'*. At the end, `file.gs` can be run in the batch mode, portrait orientation and the run command like

```
gradsc -bpc "run file.gs"
```
(147)

Further information on the GrADS script programming language can be found on the GrADS-homepage or in the reference card (*kmcm/doc*). for the KMCM.

4.4.2 Transient Lorenz Cycle (`edtrans`)

This afterburner is for studies of the Lorenz cycle. The relavant quantities are the total potential energy, the total kinetic energy, the relative and absolute angular momentum:

$$TPE = \{\rho\, c_p\, T\} \tag{148}$$

$$TKE = \left\{\rho\, \frac{\mathbf{v}^2}{2}\right\} \tag{149}$$

$$L_r = \{\rho\, u\, a_e \cos(\phi)\} \tag{150}$$

$$L_a = \left\{\rho\, \Omega\, a_e^2 \cos^2(\phi)\right\} \tag{151}$$

with the volume integral

$$\{...\} = \int \frac{dV}{4\,\pi\, a_e^2} \cdots \tag{152}$$

The available potential energy is calculated as deviations from the global mean temperature

$$APE = \left\{\frac{\rho}{2}\left(\frac{g}{N}\right)^2 \left(\frac{T - <T>_{\lambda\,\phi}}{<T>_{\lambda\,\phi}}\right)^2\right\} \tag{153}$$

$$= \left\{\frac{\rho\, g}{2}\, \frac{\left(T - <T>_{\lambda\,\phi}\right)^2}{<T>_{\lambda\,\phi}\left(g/c_p + \partial_z <T>_{\lambda\,\phi}\right)}\right\}$$

The adiabatic conversion between potential and kinetic energy is taken as

$$C = -\{\rho\, \alpha\, \omega\} \tag{154}$$

and the dissipation rate as

$$D = -\{\rho\,\mathbf{v}\,(\mathbf{H}+\mathbf{Z})\} \tag{155}$$

The different variables are split into a zonal mean and its deviations $\mathbf{v} = [\mathbf{v}]+\mathbf{v}^*$; $T = [T]+T^*$; the related energetic quantities are notated as $TKE = KZ + KE$; $APE = AZ + AE$; $C = CZ + CE$ and $D = DZ + DE$.

The program flow chart is shown in Fig. 7. The following parameters are required:

EDTRANS parameters (edt.in file):

line 1: `ch30` (ASCII): name of the steering file. Needed for `install`. If left empty, `dc.steer` is assumed. Example: dc.steer

line 2: `numfile` : number of input files <= 200. Example: 1

line 3: `file(numfile)` : input file name(s) as specified with `numfile`. Example: kmcm.new

line 4 `ncut m1 m2` (Integers): cutoff, start and end wavenumbers. Example: 63 1 63

line 5 `nsmooth, jump`: how many timesteps to ba averaged and jumped over. (see fig. 8). Example: 8 0

line 6: `ch30` (ASCII): name of the target (`edt`) file without extension. Example: test

The output comprises the following quantities appearing as time series of horizontally averaged and height-integrated values $f(t) = \{f(\lambda, \phi, p, t)\}$ in the form `f( x = 180, y = 1, z = 1, t =` t `)`.

EDTRANS output (`edt.grads file`):

dat(1) `tpe`: total potential energy TPE (J/m^2)

dat(2) `tke`: total kinetic energy TKE (J/m^2)

dat(3) `rl`: relative angular momentum L_r (kg/s)

dat(4) `al`: absolute angular momentum L_a (kg/s)

dat(5) `az`: zonal available potential energy AZ (J/m^2)

dat(6) `ae`: eddy available potential energy AE (J/m^2)

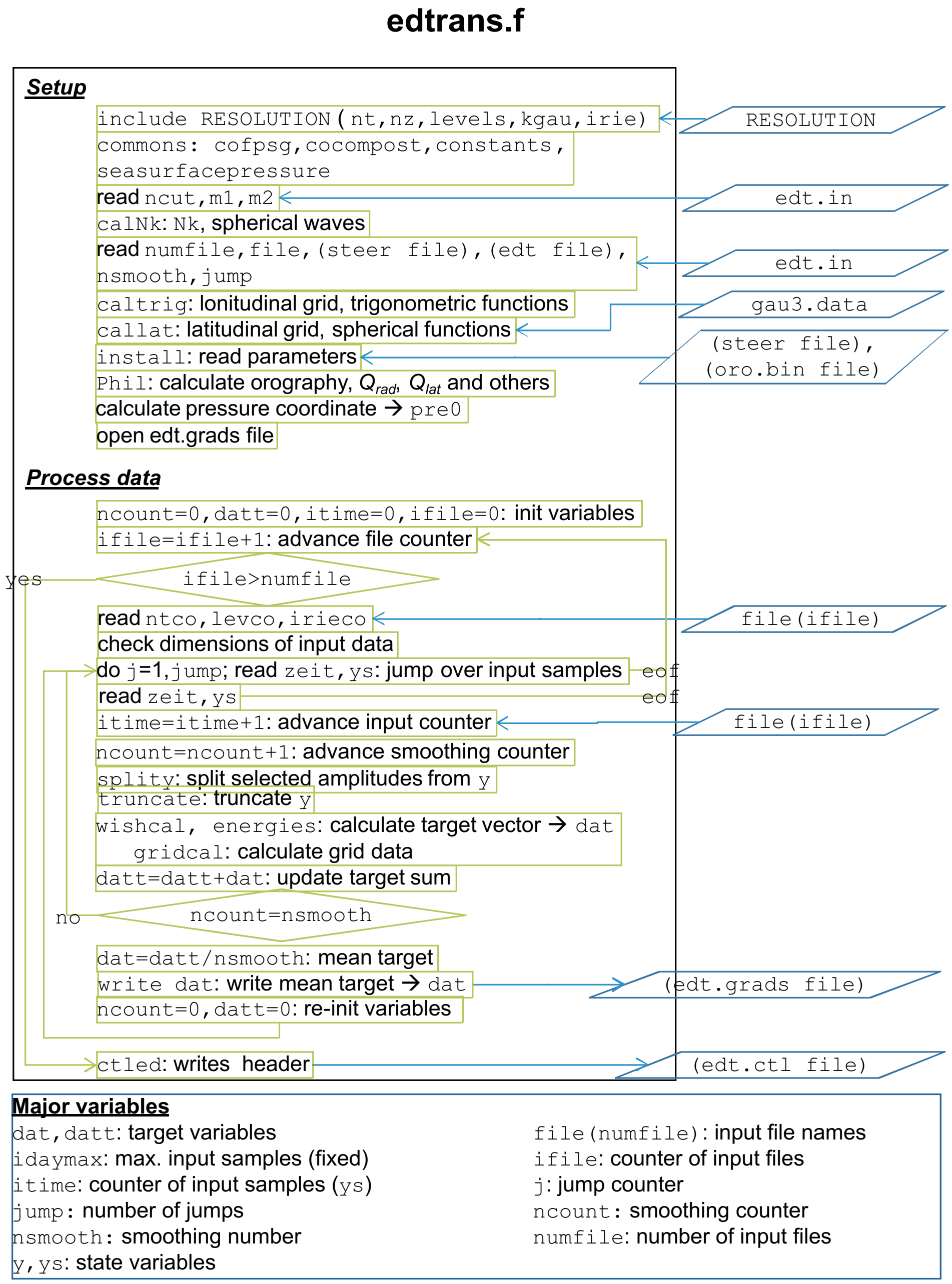

Figure 7: Program flow chart and major parameters of `edtrans.f`

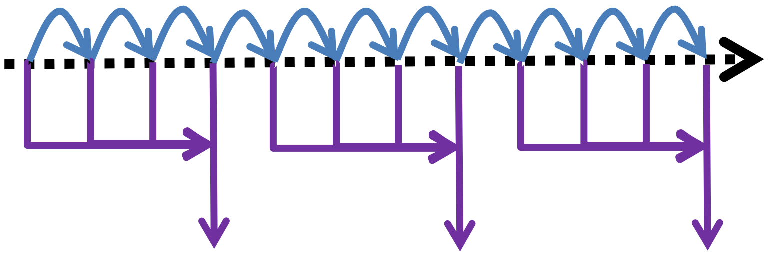

Figure 8: Scetch of the sampling procedure including `jump = 2` and `nsmooth = 4`. This way, one output is saved each `(jump + 1)*nsmooth=12` input time steps.

dat(7) `kz`: zonal kinetic energy KZ (J/m^2)

dat(8) `ke`: eddy kinetic energy KE (J/m^2)

dat(9) `hd`: horizontal dissipation $\{\rho\,\epsilon_h\}$ (W/m^2)

dat(10) `vd`: vertical dissipation $\{\rho\,\epsilon_v\}$ (W/m^2)

dat(11) `he`: d_tTKE owing to horiz. diff. $\{\mathbf{v}\,\mathbf{Z}\}$ (W/m^2)

dat(12) `ve`: d_tTKE owing to vert. diff. $\{\mathbf{v}\,\mathbf{H}\}$ (W/m^2)

dat(13) `ht`: d_tTPE owing to horiz. diff. $\{c_p\,\mu_h\}$ (W/m^2)

dat(14) `Srel`: relative entropy $S_{rel} = \{c_p \ln(T/T_{00}) - R\ln(p/p_{00})\}$ (J/K/m^2)

dat(15) `cz`: zonal adiabatic conversion CZ (W/m^2)

dat(16) `ce`: eddy adiabatic conversion CE (W/m^2)

dat(17) `dz`: zonal dissipation $DZ = -\{[\mathbf{v}]\,[\mathbf{H}+\mathbf{Z}]\}$ (W/m^2)

dat(18) `de`: eddy dissipation $DE = -\{[\mathbf{v}^*\,(\mathbf{H}^*+\mathbf{Z}^*)]\}$ (W/m^2)

A check plot (Fig. 9) is created with the following script:

```
#!/bin/tcsh -f
#  kmcm.edt.plot.csh
#
#####
# kmcm.edt.plot.csh: run edtrans and grads (13 Jul 2010 CZ)
#####
#
set runName = 'kmcm'
cp ${runName}.res ./dclib/RESOLUTION
```

```
cp ${runName}.gau3 ./dclib/gau3.data
set edFile = ${runName}.edt
set inFile = ${edFile}.in
set outFile = ${edFile}.out
set plotFile = ${edFile}.plot
set gsFile = ${plotFile}.gs
#
#
# --- EDTRANS ---
#
make edtrans.exe >&! make.out
#
cat >  $inFile << EOF
42 1 42
1
${runName}.new
${runName}.steer
${runName}
1 0
EOF
#
./edtrans.exe < $inFile >&! $outFile
#
#
# --- GRADS ---
#
cat > $gsFile << EOF
* GrADS script
*
* reinit
'reinit'
* read data
'open '${edFile}
*
* print time series
dummy = openpage()
'set y 1'
'set x 1'
'set t 1 8'
'set LEV 0'
'set vrange -0.025 0.025'
'display hd'
'display he'
'draw title Dissipation'
'draw xlab time'
'draw ylab dissipation [ W / m\`a2\`n ]'
dummy = printpage("hdhe")
*
'quit'
*
function openpage()
  'set vpage 0.0 8.0 0.0 8.0'
  'set parea 1.0 7.5 1.0 7.0'
  'set grads off'
*  'set xyrev on'
return
*
function printpage(pname)
  'set vpage off'
  'set parea off'
```

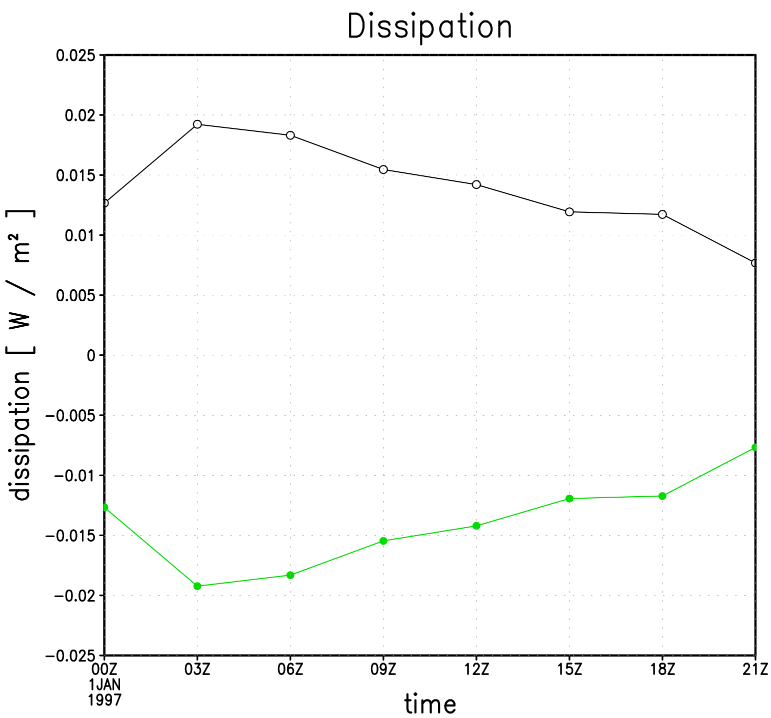

Figure 9: Time series of horizontal dissipation `hd` (empty circles) and related kinetic energy loss `he` (full circles) as generated with GrADS from `edtrans` as `kmcm.edt.plot-hdhe.eps`. Both are appearing with different signs as they should.

```
  'enable print scr.meta'
  'print'
  'disable print'
  '!gxeps -i scr.meta -c -d -o '${plotFile}'-'pname'.eps'
  '!rm scr.meta'
  'clear'
return
EOF
#
gradsc -cbp 'run '$gsFile >&! grads.out
#
exit
```

4.4.3 Transient zonal-mean budgets (`zobutrans`)

This afterburner calculates the transient zonal-mean momentum and heat budgets. Denoting the zonal mean with $[f]$ and the deviations with f^* the basic equations are

$$\partial_t[u] \quad = \quad (f + [\xi])\,[v] - [\omega]\,\partial_p[u] + [R_u] + \tag{156}$$

$$
\begin{aligned}
&+ \quad [\xi^* \, v^*] - \frac{\partial_y}{\cos^2(\phi)} [\cos^2(\phi)\, u^* \, v^*] - \partial_p [u^* \, \omega^*] \\
\partial_t [T] \quad &= \quad -[v]\, \partial_y [T] - [\omega]\, \partial_p [T] + \frac{R}{c_p\, p} [T]\, [\omega] + [Q] \qquad (157) \\
&- \quad \frac{\partial_y}{\cos(\phi)} [\cos(\phi)\, T^* \, v^*] - \partial_p [T^* \, \omega^*] + \frac{R}{c_p\, p} [T^* \, \omega^*]
\end{aligned}
$$

The terms can be rearranged to

$$
\partial_t [u] = (f + [\xi])\, v_{res} - \omega_{res} \partial_p [u] + \frac{\partial_y}{\cos^2(\phi)} EPF_y + \partial_p EPF_p \qquad (158)
$$

with a residual circulation

$$
\begin{aligned}
v_{res} \quad &= \quad [v] - \partial_p \left(\frac{[T^* \, v^*]}{R\,[T]/(c_p\, p) - \partial_p [T]} \right) \qquad (159) \\
\omega_{res} \quad &= \quad [\omega] - \frac{\partial_y}{\cos(\phi)} \left(\frac{\cos(\phi)\, [T^* \, v^*]}{R\,[T]/(c_p\, p) - \partial_p [T]} \right) \qquad (160)
\end{aligned}
$$

and the Eliassen-Palm Flux (EPF) reading

$$
\begin{aligned}
EPF_y \quad &= \quad \cos^2(\phi)\, [u^* \, v^*] + \partial_p [u] \frac{[T^* \, v^*]}{R\,[T]/(c_p\, p) - \partial_p [T]} \qquad (161) \\
EPF_p \quad &= \quad [u^* \, \omega^*] + (f + [\xi])\, \frac{[T^* \, v^*]}{R\,[T]/(c_p\, p) - \partial_p [T]} \qquad (162)
\end{aligned}
$$

The latter contains quasigeostrophic contributions

$$
\begin{aligned}
qgEPF_y \quad &= \quad \cos^2(\phi) [u^* \, v^*] \qquad (163) \\
qgEPF_p \quad &= \quad f \frac{[T^* \, v^*]}{R\,[T]/(c_p\, p) - \partial_p [T]} \qquad (164)
\end{aligned}
$$

The gravity wave drag is defined as

$$
GWD = -\partial_p [u^* \, \omega^*] \qquad (165)
$$

The program flow plan is in Figure 10.

ZOBUTRANS input (`bst.in` file):

line 1 `ncut m1 m2` (Integers): cutoff, start and end wavenumbers. Example: 63 1 63

line 2 `numfile`: number of files <= 200. Example: 1

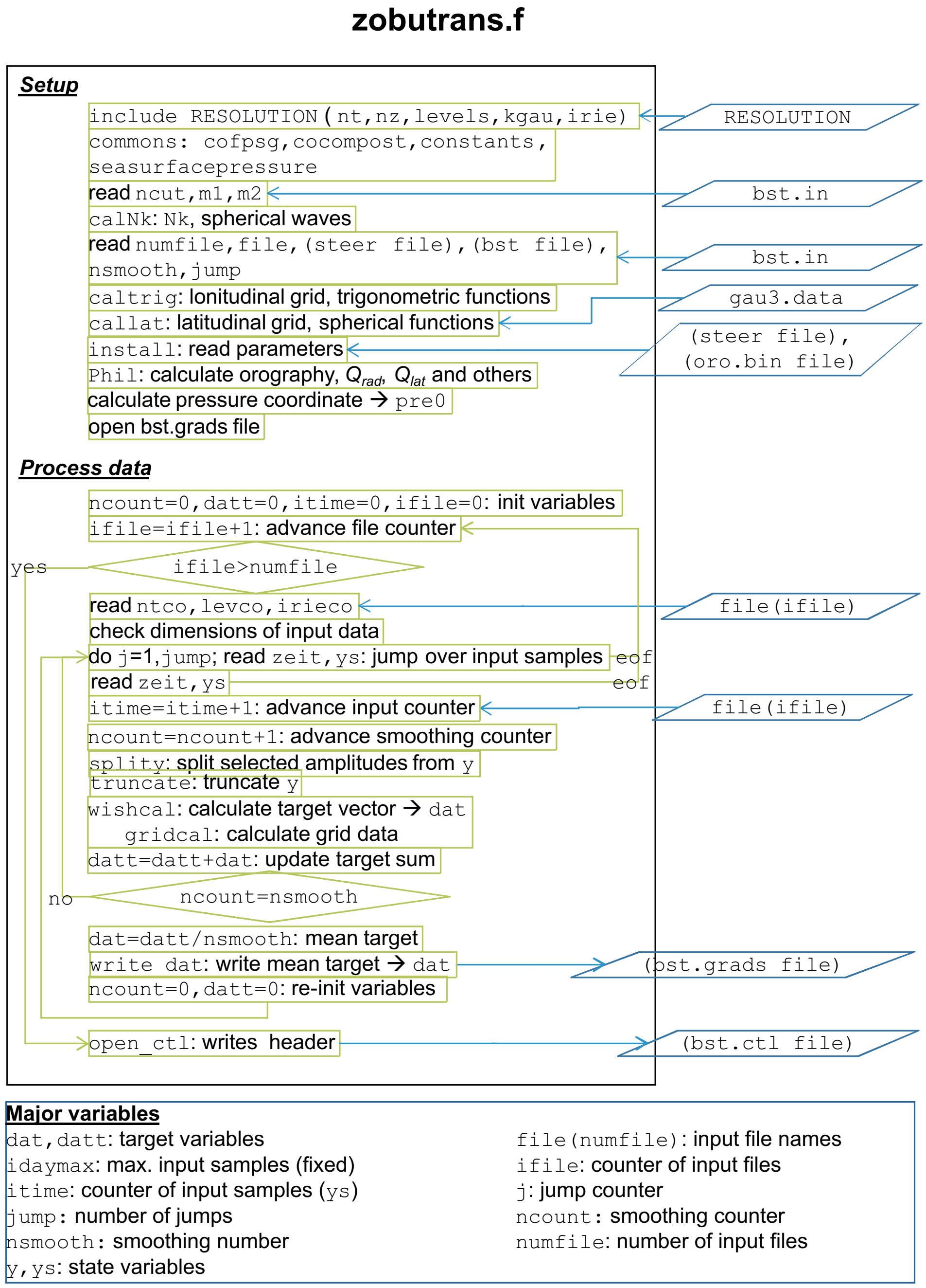

Figure 10: Program flow chart and major parameters of `zobutrans.f`

line 3 `file(numfile)`: file name(s) as specified with `numfile`. Example: test_1

line 4 `ch30` (ASCII): name of the steering file. Needed for `install`. Example: dc.steer

line 5 `ch30`: name of the target (`bst`) file without extension. Example: test

line 6 `nsmooth, jump`: how many timesteps to ba averaged and jumped over. (The selection of samples is determined by `jump`: 0 will jump over none (takes every sample), 1 will jump over one (takes every second) etc. Of these selected samples, `nsmooth` are smoothed: 1 will take the selected sample, 2 will make the mean of two etc. To take the mean of eight samples, set "8 0; to show every eightth sample, set "1 7"). Example: 8 0

The output of `zobutrans` contains the latitude-height-time-dependent quantities $f(\phi, p, t)$ saved as `f( x = 0, y =` ϕ`, z =` p`, t =` t`)`. The parameters are:

ZOBUTRANS output (`bst.grads file`) :

dat(1) `fv`: zonal Coriolis force $f[v]$ (m/s/d)

dat(2) `sxv`: symmetric meridional vorticity flux $[\xi][v]$ (m/s/d)

dat(3) `exv`: eddy meridional flux of vorticity $[\xi^* v^*]$ (m/s/d)

dat(4) `sau`: MMC vertical advection of u $[\omega]\,\partial_p[u]$ (m/s/d)

dat(5) `eau`: eddy vertical advection of u $\partial_p[\omega^*\, u^*]$ (m/s/d)

dat(6) `epx`: [pressure gradient term]* $[\alpha^*\,\partial_x p^*]$ (m/s/d)

dat(7) `du`: [vertical diffusion (u)] Z_x in 20 (m/s/d)

dat(8) `hu1`: [horizontal diff. (u)]_1 H_x(mol, stat) in 19 (m/s/d)

dat(9) `hu2`: [horiz. diff. (u)]_2 H_x(dyn) in 19 (m/s/d)

dat(10) `sac`: MMC vertical advection and conversion $-[\omega]\partial_p[T]+R\,[T]\,[\omega]/(c_p\,p)$ (K/d)

dat(11) `sha`: MMC horizontal advection $-[v]\,\partial_y[T]$ (K/d)

dat(12) `eac`: eddy vertical advection and conversion $-\partial_p[T^* \omega^*] + R\,[T^* \omega^*]/(c_p\, p)$ (K/d)

dat(13) `eha`: eddy horizontal advection $-\partial_y[T^* v^*]$ (K/d)

dat(14) `q`: [relaxation+Qc] radiative and tropical-convection heating $Q_{rad} + Q_c$ (K/d)

dat(15) `qm`: [Qm*f(omega)] self-induced mid-latitude heating $Q_m|\omega|\mathrm{He}(-\omega)/\omega_m$ (K/d)

dat(16) `dt`: [vertical diffusion of heat] μ_z (K/d)

dat(17) `vd`: [vertical dissipation] ϵ_z/c_p (K/d)

dat(18) `ht`: [horizontal diffusion (T)] μ_h (K/d)

dat(19) `hd`: [horizontal dissipation] ϵ_h/c_p (K/d)

dat(20) `vdy`: [dynamic vertical viscosity] $K_{z,dyn}$ (m^2/s) and velocity Rayleigh friction C_v m/s

dat(21) `vst`: [static vertical viscosity] $K_{z,stat}$ (m^2/s)

dat(22) `vmo`: [molecular viscosity] $K_{z,mol}$ (m^2/s)

dat(23) `vlq`: [vertical mixing length squared] l_z^2 (m^2)

dat(24) `ric`: [Richardson number] Ri

dat(25) `rman`: [Richardson criterion] $F_z(Ri)$

dat(26) `bvfq`: [buoyancy frequency squared] N^2 (1/s^2)

dat(27) `Kdy`: [dynamic horizonal viscosity] $K_{h,dyn}$ (m^2/s)

dat(28) `Kst`: [static horizontal viscosity] $K_{h,stat}$ (m^2/s)

dat(29) `vr`: residual meridional wind v_res v_{res} (m/s)

dat(30) `omar`: residual pressure velocity oma_res ω_{res} (Pa/s)

dat(31) `fyqg`: meridional qg EPF $qgEPF_y$ (kg/m/s^2)

dat(32) `fyag`: meridional ag EPF $agEPF_y = EPF_y - qgEPF_y$ (kg/m/s^2)

dat(33) `fzqg`: vertical qg EPF $qgEPF_p$ (kg/m/s^2)

dat(34) `fzag`: vertical ag EPF $agEPF_p = EPF_p - qgEPF_p$ (kg/m/s^2)

dat(35) `dyqg`: meridional div. of qg EPF $\partial_y\, qgEPF_y$ (m/s/d)

dat(36) `dyag`: meridional div. of ag EPF $\partial_y\, agEPF_y$ (m/s/d)

dat(37) `dzqg`: vertical div. of qg EPF $\partial_p\, qgEPF_p$ (m/s/d)

dat(38) `dzag`: vertical div. of ag EPF $\partial_p\, agEPF_p$ (m/s/d)

dat(39) `gwd`: gravity-wave drag GWD (m/s/d)

dat(40) `rau`: vertical advection by oma_r $\omega_{res}\, \partial_p[u]$ (m/s/d)

dat(41) `uo`: $[u^*\omega^*]$ (m Pa/s^2)

dat(42) `vo`: $[v^*\omega^*]$ (m Pa/s^2)

dat(43) `To`: $[T^*\omega^*]$ (K Pa/s)

dat(44) `uv`: $[u^*v^*]$ (m^2/s^2)

dat(45) `Tv`: $[T^*v^*]$ (K m/s)

dat(46) `u`: zonal wind $[u]$ (m/s)

dat(47) `v`: meridional wind $[v]$ (m/s)

dat(48) `oma`: pressure velocity $[\omega]$ (Pa/s)

dat(49) `T`: temperature $[T]$ (K)

dat(50) `p`: real pressure on hybrid surfaces $[p]$ (Pa)

dat(51) `geo`: geopotential height on hybrid surfaces $[Z]$ (km)

dat(52) `f`: Coriolis parameter f (1/s)

dat(53) `xi`: relative vorticity $[\xi]$ (1/s)

dat(54) `sf`: Eulerian streamf. $[\Psi]$ from $\partial_p[\Psi] = [v]$ (10^9 kg/s)

dat(55) `sfr`: residual streamf. Ψ_{res} from $\partial_p\Psi_{res} = v_{res}$ (10^9 kg/s)

dat(56) `pv`: qg potential vorticity qgP (1/s)

dat(57) `pvy`: meridional PV gradient $\partial_y\, qgP$ (1/s/m)

dat(58) `uq`: $[uu]$ (m^2/s^2)

dat(59) `vq`: $[vv]$ ($\mathrm{m}^2/\mathrm{s}^2$)

dat(60) `uagq`: $[u_{ag}u_{ag}]$ ($\mathrm{m}^2/\mathrm{s}^2$)

dat(61) `vagq`: $[v_{ag}v_{ag}]$ ($\mathrm{m}^2/\mathrm{s}^2$)

dat(62) `Tq`: $[TT]$ (K^2)

dat(63) `HLQ`: squared horiz. mixing length l_h^2 (m^2)

dat(64) `tau`: relaxation time τ (days)

dat(65) `Kf`: diff. coeff. for SFHD $K_{h,fil}$ (m^2/s)

dat(66) `Te`: equilibrium temperature $[T_e]$ (K)

dat(67) `sco`: MMC adiabatic conversion $-R/c_p\,[T]\,[\omega]/[p]$ (K/d)

dat(68) `eco`: eddy adiabatic conversion $-R/c_p\;([\,T\,\omega/p\,] - \;[T]\,[\omega]/[p])$ (K/d)

An example for application of the `zobutrans` afterburner is given with the following shell script.

```
#!/bin/tcsh -f
#  kmcm.bst.plot.csh
#
#####
# kmcm.bst.plot.csh: run zobutrans.exe and grads with bst.in (06 Jul 2010 CZ)
#####
#
set runName = 'kmcm'
cp ${runName}.res ./dclib/RESOLUTION
cp ${runName}.gau3 ./dclib/gau3.data
set bstFile = ${runName}.bst
set inFile = ${bstFile}.in
set outFile = ${bstFile}.out
set plotFile = ${bstFile}.plot
set gsFile = ${plotFile}.gs
#
#
# --- ZOBUTRANS ---
#
make zobutrans.exe >&! make.out
#
cat >  $inFile << EOF
42 1 42
1
${runName}.new
${runName}.steer
${runName}
8 0
EOF
#
./zobutrans.exe < $inFile >&! $outFile
```

```
#
#
# --- GRADS ---
#
cat > $gsFile << EOF
* GrADS script
*
* reinit
'reinit'
* read data
'open '$bstFile
*
* print wind squared
'set vpage 0 8 0 10'
'set parea 1 7.5 1 7.5'
'set y 1'
'set lat -90 90'
'set t 1'
'set zlog on'
'set ylevs 1000 300 100 30 10 3 1'
'set grads off'
'set gxout contour'
'display vq'
'draw title zonal mean merdional wind squared (m\`a2\`n/s\`a2\`n)'
'draw xlab latitude'
'draw ylab altitude (hPa)'
dummy = printpage("vq")
*
'quit'
*
function openpage()
  'set vpage 0.0 8.0 0.0 8.0'
  'set parea 1.0 7.5 1.0 7.0'
  'set grads off'
*  'set xyrev on'
return
*
function printpage(pname)
  'set vpage off'
  'set parea off'
  'enable print scr.meta'
  'print'
  'disable print'
  '!gxeps -i scr.meta -c -d -o '${plotFile}'-'pname'.eps'
  '!rm scr.meta'
  'clear'
return
EOF
#
gradsc -cbp 'run '${gsFile} >&! grads.out
#
exit
```

The resulting picture is shown in fig. 11

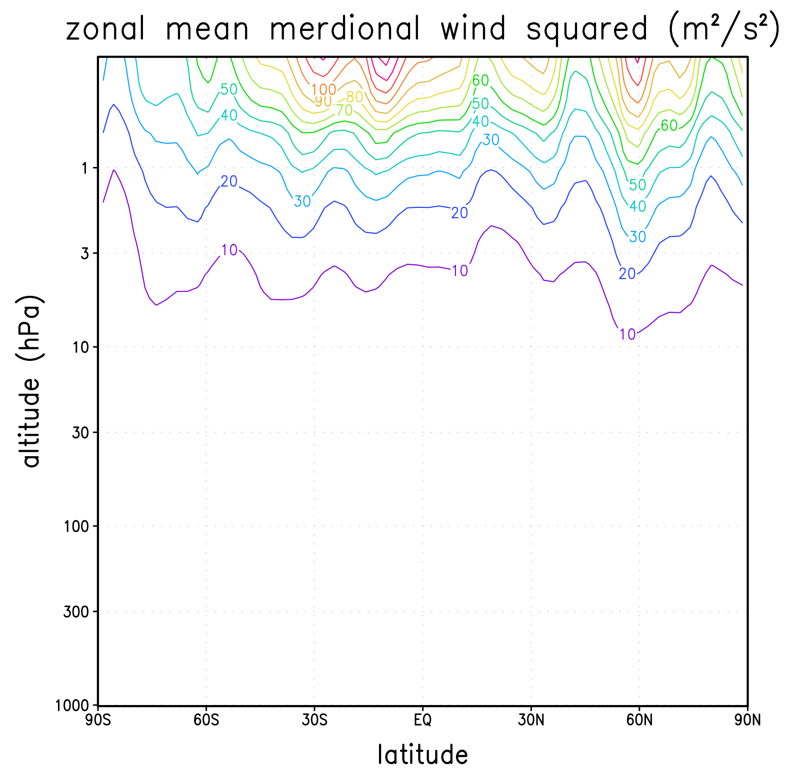

Figure 11: Vertical section of squared meridional wind fluctuations `vq` as generated with GrADS from `zobutrans` as `kmcm.bst.plot-vq.eps`.

4.4.4 Transient height-resolved data (hetrans)

This afterburner puts out a number of space-time dependent variables in all dimensions (the term "height-resolved" has remained from historical reasons). The program flow chart is shown in Fig. 12 In order to avoid the step-by-step manual input of parameters a namelist can also be provided as `hetrans.exe < hebt.in` which has the following structure:

HETRANS parameters (hebt.in file):

line 1 `ch30` (ASCII): name of the steering file. Needed for `install`. Example: dc.steer

line 2 `ncut m1 m2` (Integers): cutoff, start and end zonal wavenumbers. Example: 63 1 63

line 3 `zone(1) zone(2)`: longitude window (degrees). Example: 0 360

line 4 `intlon`: longitude interval. 1 yields `irie` longitudes, 2 yields `irie`/2, etc. Example: 1

line 5 `nlat`: number of latitudes. If it is less-or-equal 9, this number of latitudes has to be provided in the next line. If `nlat` is in the range $9 < \texttt{nlat} < \texttt{kgau} - 2$, the latitude range has to be provided. Example: 45

line 6 `breite(nlat)`: latitudes (deg). Both `nlat` and `breite` are needed in subroutine `callat`. Example: -90 90

line 7 `levbo levto intlev`: lower and upper level index and interval. Example: 221 1 1

line 8 `numfile`: number of files <= 200. Example: 1

line 9 `file(numfile)`: file name(s) as specified with `numfile`. Example: test_1

line 10 `ch30`: name of the target (`hebt`) file without extension. Example: test

line 11 `nsmooth, jump`: how many timesteps to ba averaged and jumped over. (The selection of samples is determined by `jump`: 0 will jump over none (takes every sample), 1 will jump over one (takes every second) etc. Of these selected samples, `nsmooth` are smoothed: 1 will take the selected sample, 2 will make the mean of two etc. (see fig. 8). To take

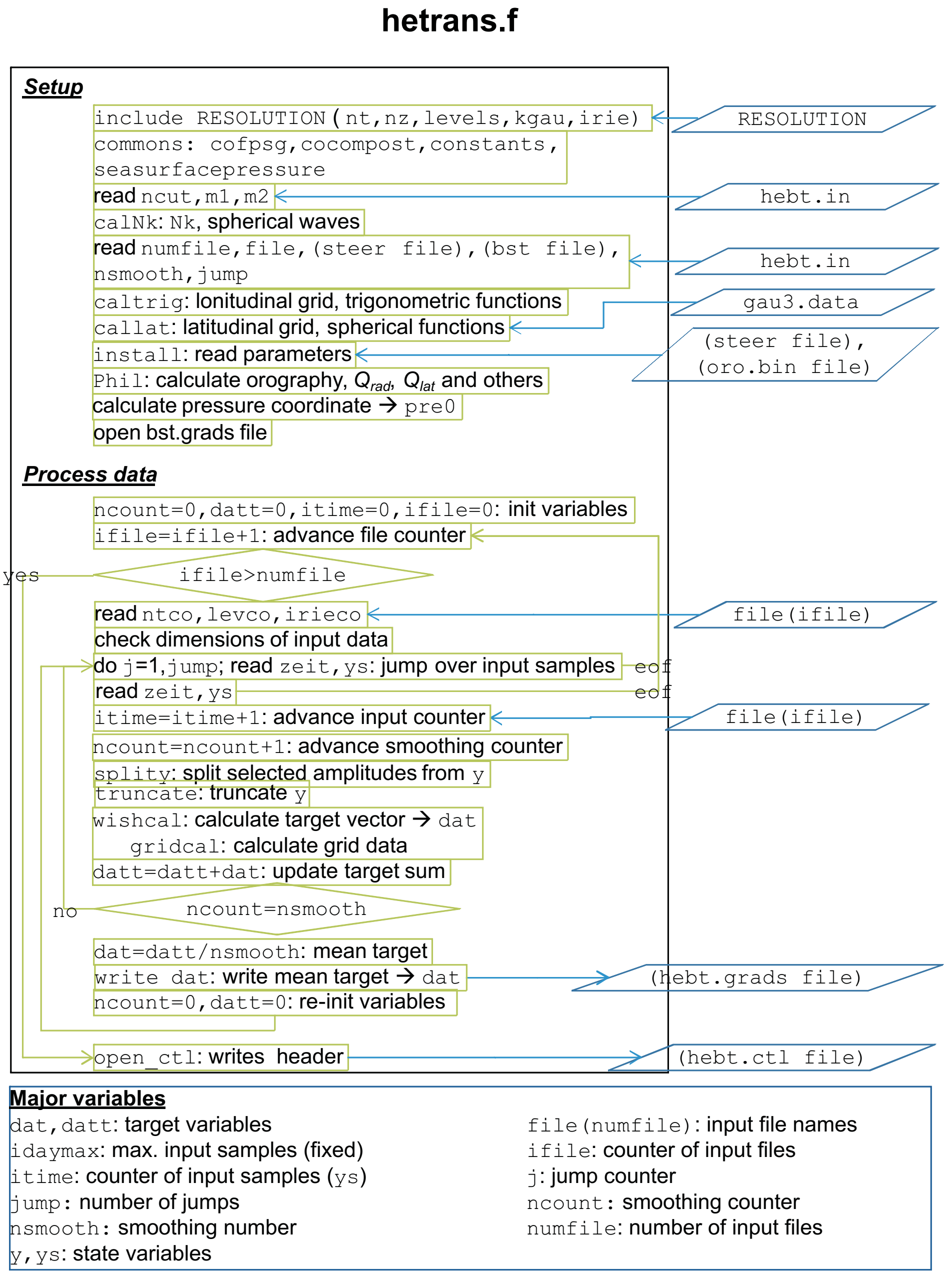

Figure 12: Program flow chart and major parameters of `hetrans.f`

the mean of eight samples, set "8 0; to show every eightth sample, set "1 7"). Example: 8 0

The 4D output variables $f(\lambda, \phi, p, t)$ are stored as `f( x =` λ`, y =` ϕ`, z =` p`, t =` t`)` . In some cases, the surface pressure level $p = 1013$ hPa is reserved for the height-integrated quantities $\int dz\, c_p\, \rho \ldots$. Zonal mean and deviation are used as

$$f = [f] + f^* \tag{166}$$

HETRANS output (`hebt.grads file`):

dat(1) `p` pressure p (mb)

dat(2) `geo` geopotential height Z (km)

dat(3) `u` zonal wind velocity u (m/s)

dat(4) `v` meridional wind velocity v (m/s)

dat(5) `ua` zonal ageostrophic wind velocity u_{ag} (m/s)

dat(6) `va` meridional ageostrophic wind velocity v_{ag} (m/s)

dat(7) `oma` pressure velocity ("OMegA") ω (Pa/s)

dat(8) `t` temperature T (K)

dat(9) `wir` horizontal vorticity "(WIRbelstärke") ξ (1/s)

dat(10) `div` horizontal divergence D (1/s)

dat(11) `psi`: horizontal streamfunction divided by squared Earth radius Ψ/a_e^2 (1/s)

dat(12) `dum`: dummmy

dat(13) `q`: relaxation plus condensational heating ($(T_e - T)/\tau + Q_c$ (K/d)) and vertical integral (W / m^2)

dat(14) `qmi`: self-induced heating Q_m (K/d) and vertical integral (W/m^2)

dat(15) `vdt`: vertical diffusion of heat μ_z (K/d) and vertical integral (W/m^2)

dat(16) `hdt`: horizontal diffusion of heat μ_h (K/d) and vertical integral (W/m^2)

dat(17) `vd`: vertical dissipation ϵ_z/c_p (K/d) and vertical integral (W/m^2)

dat(18) `hd`: horizontal dissipation ϵ_h/c_p (K/d) and vertical integral (W/m^2)

dat(19) `vad`: vertical advection of heat $-\dot{\eta}\partial_\eta T$ (K/d)

dat(20) `had`: horizontal advection of heat $-(\mathbf{v}\cdot\nabla)T$ (K/d)

dat(21) `kzd`: dynamical vertical diffusivity $K_{z,dyn}$ (m^2/s) including momentum Rayleigh friction C_v (m/s)

dat(22) `khd`: dynamical horizontal diffusivity $K_{h,dyn}$ (m^2/s)

dat(23) `bvfq`: squared Brunt Vaisala frequency N^2 (1/s^2)

dat(24) `rman`: Richardson criterion $F_z(Ri)$

dat(25) `uv`: meridional eddy momentum flux u^*v^* (m^2/s^2)

dat(26) `tv`: meridional eddy heat flux T^*v^* (Km/s)

dat(27) `uo`: vertical eddy momentum flux $u^*\omega^*$ (mPa/s^2)

dat(28) `to`: vertical eddy heat flux $T^*\omega^*$ (KPa/s)

dat(29) `ek`: kinetic energy $(u^2+v^2)/2$ (m^2/s^2)

dat(30) `dek`: divergent kinetic energy $(u_{ag}^2+v_{ag}^2)/2$ (m^2/s^2)

dat(31) `te`: equilibrium temperature T_e (K)

dat(32) `qcm`: condensational heating rates Q_c+Q_m (K/d)

dat(33) `ved`: vertical energy dissipation $(\mathbf{v}\cdot\mathbf{Z})/c_p$ (K/d) and vertical integral (W/m^2)

dat(34) `hed`: horizontal energy dissipation $(\mathbf{v}\cdot\mathbf{H})/c_p$ (K/d) and vertical integral (W/m^2)

dat(35) `uq`: zonal wind speed squared u^2 (m^2/s^2)

dat(36) `vq`: meridional wind speed squared v^2 (m^2/s^2)

dat(37) `tq`: temperature squared T^2 (K^2)

dat(38) `eek`: eddy kinetic energy $(u^{*2}+v^{*2})/2$ (m^2/s^2)

dat(39) `rol`: Lagrangian Rossby number $Ro_L = (v_{ag} - u_{ag})/(u_g^2 + v_g^2)^{1/2}$

dat(40) `co`: adiabatic conversion $-R/c_p\, T\, \omega/p$ (K/d) and vertical integral (W/m^2)

A demonstration can be run with the following script:

```
#!/bin/tcsh -f
#  kmcm.hebt.plot.csh
#
#####
# kmcm.hebt.plot.csh: run hetrans.exe and grads with het.in
# (14 Apr 2010 CZ)
#   included reformed hetrans.f (07 Jul 2010 CZ)
#   included PBS section (13 Jul 2010 CZ)
#####
#
#
# --- PBS ---
#
# To run on a machine with PBS queue system like orion, include the
# following lines
#PBS -l mem=8gb
#PBS -q q4xl
cd $PBS_O_WORKDIR
# and submit this like
#  qsub kmcm.hebt.plot.csh
#
#
# --- SETTINGS ---
#
set runName = 'kmcm'
cp ${runName}.res ./dclib/RESOLUTION
cp ${runName}.gau3 ./dclib/gau3.data
set hebtFile = ${runName}.hebt
set inFile = ${hebtFile}.in
set outFile = ${hebtFile}.out
set plotFile = ${hebtFile}.plot
set gsFile = ${plotFile}.gs
#
#
# --- HETRANS ---
#
make hetrans.exe >&! make.out
#
cat >  $inFile << EOF
${runName}.steer
42 1 42
0 360
2
60
-90 90
30 1  1
1
${runName}.new
${runName}
1 0
```

```
EOF
#
./hetrans.exe < $inFile >&! $outFile
#
#
# --- GRADS ---
#
cat > $gsFile << EOF
* GrADS script
*
* reinit
'reinit'
*
* read data:
* 'open kmcm.hebt'
'open '$hebtFile
*
* print section with te and u
dummy = openpage()
'set lev 1000 10'
'set lat -90 90'
'set lon 0'
'set t 1'
'display te'
'display u'
'draw title Te and u'
'draw xlab latitude [ deg ]'
'draw ylab altitude [ hPa ]'
dummy = printpage("teu")
*
* print map with qcm
dummy = openpage()
'set lev 900'
'set lon -180 180'
'set t 8'
'display qcm'
'draw title Q_cm'
'draw ylab latitude [ deg ]'
'draw xlab longitude [ deg ]'
dummy = printpage("qcm")
*
'quit'
*
function openpage()
  'set vpage 0.0 8.0 0.0 8.0'
  'set parea 1.0 7.5 1.0 7.0'
  'set grads off'
return
*
function printpage(pname)
  'set vpage off'
  'set parea off'
  'enable print scr.meta'
  'print'
  'disable print'
  '!gxeps -i scr.meta -c -d -o '${plotFile}'-'pname'.eps'
  '!rm scr.meta'
  'clear'
return
EOF
```

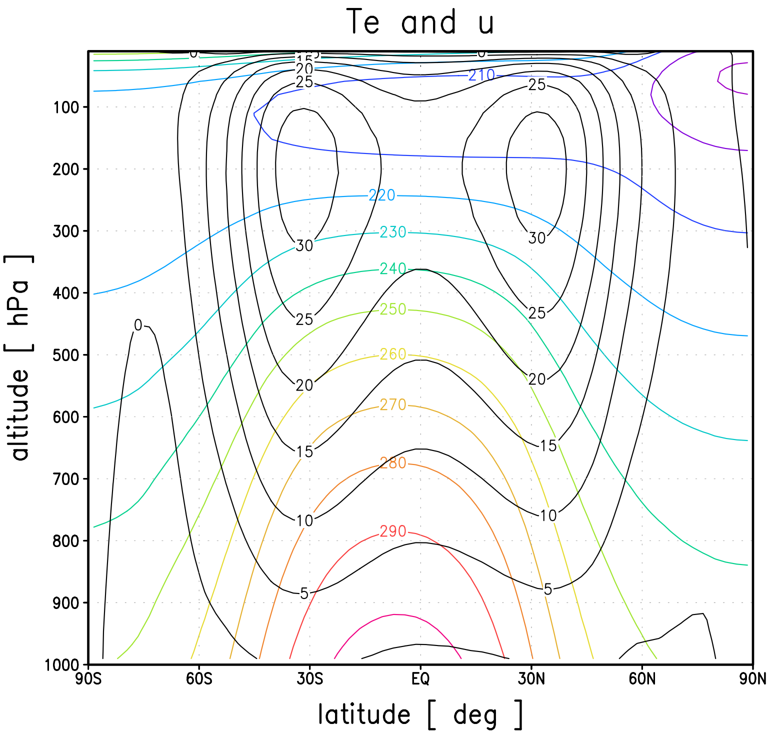

Figure 13: Meridional section at 0 °E of the equilibrium temperature T_e with the zonal wind u as generated with GrADS from `hetrans` as `kmcm.hebt.plot-teu.eps`
.

```
#
gradsc -cbp 'run '${gsFile} >&! grads.out
#
exit
```

The result are the two figures 13 and 14

5 Software

5.1 Fortran programs

Fortran programs:

`AQ_Kelvin( fi, p) in: qcm.f :`
    `compute height-latitude dependence of thermal Kelvin-wave excitation.`
    `EB / 12.02.2003 )`

`AQ_Rossby( fi, p) in: qcm.f :`
    `compute height-latitude dependence of thermal Rossby-gravity-wave`

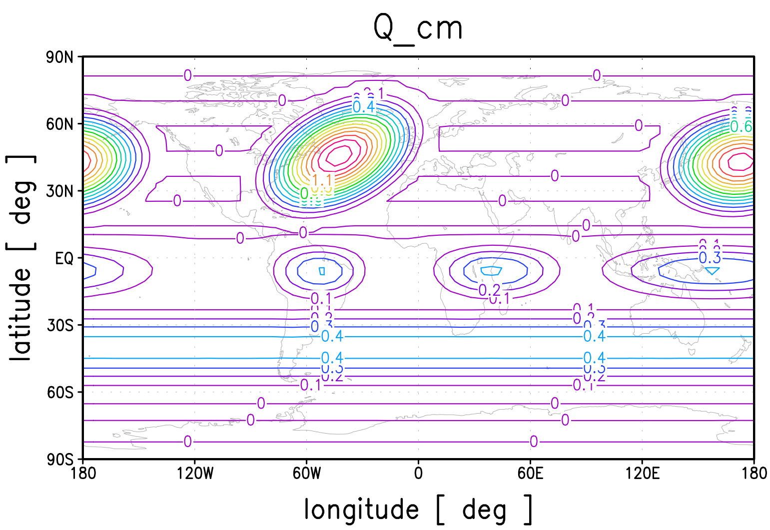

Figure 14: Horizontal map at 900 hPa of the latent heating functions $Q_c + Q_m$ as generated with GrADS from `hetrans` as `kmcm.hebt.plot-qcm.eps`
.

    excitation. (EB / 30.01.2003)

`BSSLZR( PBES, KNUM) in:  gau3cal.f :`

    BSSLZR RETURNS *KNUM* ZEROS, OR IF *KNUM>50,* *KNUM* APPROXIMATE
    ZEROS OF THE *BESSEL FUNCTION J0. letzte Aenderungen, EB /
    10.09.98

`calNk :`

    calculate total wavenumber N as a function of the spherical
    harmonics index k -> Nk(2:modr) (EB / 22.12.94 - 09.12.08 (CoRA)
)

This function inverts eq. 60 as $n = n(k, m)$.

`calNk( ntcut, nzcut) in:  worldoro-V.f :`

    calculate total wavenumber N as a function of the spherical
    harmonics index k -> Nk(2:modr). (EB / 14.10.2009)

`callat( latgrid, breite) in:  hogrid.f :`

    set latitudinal grid and compute spherical harmonics (EB /
    03.12.2003 - 11.12.2008 (CoRA)). The number of latitudes is
    latgrid. If 9 < latgrid < kgau , the latitudes are equidistant
    between breite(1) and breite(2). Prescribed latitudes provided

by the field breite are used if latgrid <= 9. Gaussian latitudes are assumed for latgrid=kgau.

Called from `MAIN` as `callat( kgau, breite)` fills the common blocks `colatitude` and `colatitude_FFT` with the Gauss latitudes read from `gau3.data`.

`calrelaxd(relaxd) in: relaxation.f` :

```
compute relaxation time constants (relaxd). A possible dependence
on latitude has not been used yet. A peak of the relaxation
time in the lowermost stratosphere was inspired by a paper of
Tim Dunkerton. ( EB / 02.09.97 - May 2006 )
```

`caltrig( longrid, zone) in: hogrid.f` :

```
set longitudinal grid and compute trigonometric functions. The
number of longitudes are equidistant between zone(1) and zone(2).
longrid must be equal irie to avoid nonsense. ( Erich Becker
/ 11.12.2008 (CoRA)
```

This procedure is called from `MAIN.f` as `caltrig(irie,zone)` with `zone` set to $[0, 360]$, hence `longrid = irie`, `zone(1) = 0` and `zone(2) = 360`. As a result, the common block `cotrig` is filled with several differentiations of $\sin(m\lambda)$ and $\cos(m\lambda)$ at each longitude grid.

`compost( pref, F1, F2, F3, F4, L1, L12, L2, L23, L3, L34)` :

```
Set the distribution of hybrid levels, i.e. calculate the discrete
half level values the vertical hybrid coordinate eta and compute
the corresponding values of the functions a(eta) and b(eta)
with respect to the sea level reference pressure pref. The
parameters F1,..,L34 specify 4 domains consisting of L1, L2,
L3, and LEVELS-(L1+L2+L3) numbers of half levels where each
half level coordinate eta(l) is computed from the adjacent lower
level as ta(l-1)=Fi*eta(l), i=1,2,3,4. The remaining parameters
L12, L23, and L34 specify the number levels used to make a "smooth"
transition between these different domains of vertical resolution.
Setting F1<0 or (L1+L2+L3)>LEVELS results in equidistant level
spacing with respect to eta. ( Erich Becker / 26.04.2004 )
Adding the option to read a and b from a "ecmwf_lev.data" for
F1>1 (CZ/20.08.2010)
```

`ctled( ch30, itime, dday ) in: edtrans.f` :

```
CTLED: makes a CTL file (hk)
rearranged time-height sorting ( CZ / 12..07.2010 )
```

```
added adiabatic conversion (cz,ce) (CZ/11.08.2010)
added dissipation rates (dz,de) (CZ/17.08.2010)
```

called by `edtrans`

`edtrans` :

```
EDTRANS: Energies of the Lorenz cycle (EB, HK / Sep 2009)
Reformed "edward" including jump/nsmooth sampling, ncut option
and correct grads storage (CZ / 13.07.2010 )
Allowed long input data files ( CZ / 27.07.2010 )
Included adabatic conversion (ce,cz) (CZ/10.08.2010)
included dissipation (de,dz) (CZ/17.08.2010)
```

See sct. 4.4.2

`energies( AZ, AE, KZ, KE, CZ, CE, DZ, DE ) in: edtrans.f` :

```
ENERGIES: energies for the Edward- N.- Lorenz- cycle ( EB /
12.10.95 - 04.03.99 )
added adiabatic conversion (CZ,CE) (CZ/10.08.2010)
added dissipation (dz,de) (CZ/17.08.2010)
```

`fpsg( p ) in psgcal.f` :

```
Nullstelle liefert psg(geopot).  ( Erich Becker / 26.09.96 )
pref durch p00 ersetzt am 21.12.2001
```

`fpsi( xpsi ) in: relaxation.f` :

```
This function is zero for xpsi=psi.  ( EB / 24.09.98 )
```

`fTe( sinfi, pre) in: relaxation.f` :

```
compute equilibrium temperature Te up to the orbit of the moon.
( Erich Becker / 09.04.99 - 10.10.2000 - 19.12.2002 )
```

`gau32( f, n, a, b, result)` :

```
Gau3-Integrator mit Nullstellen und Gewichtsfaktoren von P_32.
( Erich Becker / 22.12.94 )
```

`gau3cal` :

```
Compute gridpoints and weights for Gaussian intgration and provide
a corresponding data instruction.  The subroutines below are
adopted from ECHAM3/ECMWF. ( EB / 10.09.98 )
```

`GAUAW( PA, PW, K) in: gau3cal.f` :

```
COMPUTE ABSCISSAS AND WEIGHTS FOR *GAUSSIAN INTEGRATION.
PA - ARRAY, LENGTH AT LEAST *K,* TO RECEIVE ABSCISSAS.
PW - ARRAY, LENGTH AT LEAST *K,* TO RECEIVE WEIGHTS.
```

```
THE ZEROS OF THE *BESSEL FUNCTIONS ARE USED AS STARTING APPROXIMATIONS
FOR THE ABSCISSAS. NEWTON ITERATION IS USED TO IMPROVE THE VALUES
TO WITHIN A TOLERANCE OF *EPS.*
some modifications introduced ( EB / 10.09.98 )
```

`gridcal ( jlat, sinfico, cosfi, f,fy, ps, psx, psy, psi, wir, div, wiry, divy, u, v, T, Ty, TL, wirf, divf, wiryf, divyf, uf, vf, Tyf, TLf, wirx, divx, ux, vx, Tx, wirxf, divxf, Txf, uy, vy, uyf, vyf, uyx, uyy, vyx, vyy, ug, vg, uag, vag ):`

```
GRIDCAL: grid fields for budget calculations, directly taken
from subroutine intend.f ( Erich Becker / 30.03.2005 - 26.02.2009
)
```

The spectral amplitudes from `/coamplis/` and `/coamplisf/` are projected onto a grid. The values are passed back as parameters.

`gridtendencies ( jlat, f, ps, psx, psy, wir, div, wiry, divy, u, v, T, Ty, TL, wirf, divf, wiryf, divyf, uf, vf, Tyf, TLf, wirx, divx, ux, vx, Tx, wirxf, divxf, Txf, uy, vy, uyf, vyf, uyx, uyy, vyx, vyy, u_t, v_t, T_t, ps_t, bracket) :`

```
compute and collect the grid-space representations of all tendencies
of horizontal wind, temperature (enthalpy), and surface pressure.
The dynamic terms follow the vertical differencing method of
Simmons & Burridge (1981, MWR) and Simmons & Chen (19.., QJRMS),
using a new reference temperature according to Becker (2003,
Habilitationsschrift, Appendix A). Radiative and latent heating
is mimicked by temperature relaxation towards Te, prescribed
tropical heating Qc and self-induced heating with respect to
Qm.  All diffusion and sensible heating tendencies, as well
as the dissipation are provided by the subroutine symdiff.  (
Erich Becker / 09.09.1997 - 05.01.2009 )
```

`hetrans :`

```
HETRANS: 4d afterburner for KMCM ( EB / 10.08.2006 )
Adjusted input w/o V truncation; calculate DDAY instead of reading;
replaced zeitmax/nsmooth with jump/nsmooth sampling ( CZ / 07.07.2010
)
added vertical integral of adiabatic conversion to vad (EB/09.07.2010)
allowed long input file names (CZ/27.07.2010)
included ad.  conv.  (ac) into dat (CZ&ER/10.08.2010)
```

Transient height-resolved data for GrADS (see sct. 4.4.4)

Hm_n(l, x, fi, sinfi) in: qcm.f :
provide longitude-latitude dependence of Qn at full model layer l. (EB / 02.12.05)

Hm_s(l, x, fi, sinfi) in: qcm.f :
provide longitude-latitude dependence of Qs at full model layer l. (EB / 02.12.05)

Hm_x(l, x, fi, sinfi) in: qcm.f :
provide longitude-latitude dependence of Qx at full model layer l. (EB / 02.12.05)

Hmean(fi, shift) in: qcm.f :
compute zonal mean of horizontal dependence of Qm for given shift. (EB / 02.12.05)

Hmeas(fi, shift) in: qcm.f :
compute zonal mean of horizontal dependence of Qs for given shift. (EB / 02.12.05)

Hmeax(fi, shift) in: qcm.f :
compute zonal mean of horizontal dependence of Qx for given shift. (EB / 02.12.05)

iapsg(modco,sinfico,apsg) in: psgcal.f :
provide vector whose integration over sin(fi) yields the spectral representation of the reference surface pressure. (EB / 06.01.97 - 22.09.98)

igwumlesen(ch1, y, ZA) in: semiint :
ch1 = 'f'orward: writes prognostic variables from state vector y onto ZA = (divergence, temperature, surface pressure); ch1 = 'b'ack: writes ZA onto state vector y (EB / 29.08.2002)

inidc :
zonal-mean initial conditions that fulfills graident wind balance. no topopgraphy, ps=p00, and symmetry about the equator are implied. some waves can be excited in order to simulate a baroclinic life cycle. (Erich Becker / 20.03.2010)

install(step, daystop, intstep, IMEAN, ch30) :
read variable model parameters from dc.steer and manage all calculations prior to time integration. (EB / 20.11.2005)

`intend( jlat, sinfico, dydt ) :`

`compute Fourier transforms of the grid-space representations of wind, temperature, and surface pressure tendencies and multiply these by the Legendre functions such that further integration about sin(fi) gives the tendency of the spectral state vector. The present subroutine takes advantage of FFT. To see what's going on look into intend_std.f! (Erich Becker and Joerg Hertzer (NEC) / December 2003) (with spectrally filtered fields: EB / 04.01.09) (extended for variable spectral zonal resolution: EB (25.07.09))`

`Lc( x ) in: qcm.f :`

`longitude dependence of Qc. ( EB / 09.09.97 - 02.12.05 )`

`MAIN :`

`Driver of the "Kuhlungsborn Mechanistiv general Circulation Model" (KMCM) with variable spectral zonal resolution ( Erich Becker / 28.11.2004 - 24.07.2009 )`

`matcal( step ) in: IGW.f :`

`compute matrices for semi-implicit time stepping. ( Erich Becker / 16.06.94 - 09.10.96 )`

`NULLD( F, X1, X2, X0, EPS, M) in: ../elibs/. :`

`ILLINOIS-ALGORITHMUS ZUR BESTIMMUNG DER NULLSTELLE X0 DER FUNKTION F(X) IM INTERVALL X1..X2 MIT DER ABSOLUTEN GENAUIGKEIT EPS. M IST MAXIMALE ITERATIONSZAHL. (23.11.88; AUF DOUBLE PRECISION 22.01.94)`

`open_ctl ( ch30, itime, dday, intlon, nlat, zone, breite, ldruck, druck) in: hetrans.f:`

`open grads and ctl file ( Erich Becker / 19.01.2005 )`
`allowed HR in TDEF ( CZ / 08.07.2010 )`
`included ad. conv. (co) (CZ&ER/10.08.2010)`

Called by `hetrans`: Writes header information (dimensions, names, descriptions) to a CTL file for GrADS. It's name is passed with the variable `ch30`.

`open_ctl ( ch30, itime, dday, nlat, pre0 ) in: zobutrans.f:`

`open grads and ctl file ( Erich Becker / 19.01.2005 )`
`allowed HR in TDEF ( CZ / 08.07.2010 )`
`added ad. conv. (sco, eco), omitted dat(0) (CZ/10.08.2010)`

Called by `zobutrans`

`Phil( longrid, latgrid) :`

`compute orography, surface temperature, relaxation time and relaxation temperature, latent heating functions, roughness length, and reference surface pressure for the 3d-grid defined by orography, compost parameters, as well as longrid and latgrid ( Erich Becker / June 2007 - 26.07.2009 )`

`preIGW in: IGW.f` :

`provide coefficients to compute the linear GW terms according to the reference state. ( Erich Becker / (09.10.96 )`

`preqcm in: qcm.f` :

`read all parameters for latent heating functions from *.steer and make all necessary preparations for computation of Qc, Qn, Qs, Qx, and thermal excitations of equatorial waves. ( Erich Becker / 18.09.97 - 02.12.2005 )`

`presymdiff in: symdiff.f` :

`Purpose: read all parameters necessary to define the vertical and horizontal diffusion coefficients, as well as the surface coefficients; provide the vertical profiles of the mixing lengths, the offset Richardson numberi, and the prescribed diffusion coefficients. ( Erich Becker / 26.05.2008 )`

`preTref` :

`provide parameters for reference temperature ( Erich Becker 06.01.97 )`

`pretrelax in: relaxation.f` :

`prepare all parameters for computation of equilibrium temperature Te and relaxation time (tau). ( Erich Becker 24.09.98 )`

`profile( lev, etaq, prof, bo, to, s, ds) in: symdiff.f` :

`provides standard profiles for horizontal diffusion (HK, HKf, and HLQ)`

`psgcal` :

`calculate the spectral representation apsg of the surface pressure psg associated with the reference temperature Tref(p) and the orography as already read by install.f; update pref which represents the mass of the atmosphere. ( Erich Becker / 06.01.97 )`

Qcz(fi, p) in: qcm.f :
latitude-pressure dependence of Qc (EB / July 1995 - 02.12.05)

readte in: relaxation.f :
read Te parameters from dc*.steer (unit=15). (EB / 28.12.07)

semiint(ym2, ym1, y, dydt, step) :
performs semi-inplicit time step (EB / 19.11.1996 - march 2003)

specspli(awir, atem) in: inidc.f :
Spectral decomposition onto the spherical functions normalized to Unity (EB / 09.10.95 - 17.12.96)

splity(y) :
split spectral state vector into individual spectral amplitudes of horizontal vorticity and streamfunction, horizontal divergence and velocity potential, temperature, and surface pressure. (EB / 27.06.94 - CoRA, 12.07.2008)

These amplitudes are stored in the common blocks /coamplis/ and /coamplisf/.

symdiff (jlat, alth, preh, pre, dipre, ps, psx, psy, wir, div, wiry, divy, u, v, T, Ty, TL, wirf, divf, wiryf, divyf, uf, vf, Tyf, TLf, wirx, divx, ux, vx, Tx, wirxf, divxf, Txf, uy, vy, uyf, vyf, uyx, uyy, vyx, vyy, diuh, divh, diTh, udif, vdif, Tdif):
compute wind and temperature tendencies owing to vertical diffusion (turbulent and molecular) and horizontal diffusion, including the frictional heating. A local mixing length based PBL scheme is used as in NCAR-CCM1 (Holtslag and Boville, J. Climate 6, 1825-1842, 1993). Both K_z and C can be supplemented by an additional prescribed background diffusion. The profile of this background diffusion coeff. is also applied to the offset of the Richardson number, as well as to the asymptotic vertical mixing length. The no-slip condition is used to compute the frictional heating associated with vertical momentum, ensuring energy conservation for each column (Becker, 2003, MWR). Horizontal momentum diffusion is based on the symmetric stress tensor formulations as given in Becker (2001, JAS; 2003, MWR). A nonlinear diffusion

according to Becker & Burkhardt (2007, MWR) can be chosen. The formulation fully accounts for spherical geometry consistent with Smagorinsky (1993). In order to ensure energy and angular momentum conservation on hybrid surfaces, the momentum and temperature diffusion tendencies contain a special correction involving the surface pressure gradient. Adjustment to dynamic instability of gravity waves in the middle atmosphere is realized by the scaling the nonlinear horizontal diffusion coefficient by the same Richardson criterion that is applied in the vertical diffusion scheme (Becker, 2008, JAS, submitted).(Erich Becker / 09.02.2005 - 20.05.2008)

`Taical( ntco, pre, y) in: inidc.f :`
y(n) = atem(n, pressure = pre) / p . (EB / 23.11.94)

`tendencies( time, y, dydt ) :`
computes full tendencies dydt(n), n=1...ny of the spectral state vector components y(n), n=1,...,ny. For semi-implicit integration (switch=.false.), horizontal and vertical diffusion and frictional heating tendencies are taken from the previous time step. (Erich Becker / 17.09.96 - 18.02.2009)

`Tespec( n1, sinfi, y) in: inidc.f :`
y(n) = Te * PN(n-1), PN = ordinary Legendre-Polynom, n1 = nt+1. (EB / 22.11.94)

`truncate( ncut, m1, m2 ) :`
TRUNCATE: truncate spectral amplitudes at total wavenumber ncut and select zonal wavenumbers from m1 to m2. (EB / 26.02.09)

`umlesen( y ) :`
split spectral state vector into individual spectral amplitudes of horizontal vorticity and streamfunction, horizontal divergence and velocity potential, temperature, and surface pressure; provide also spectrally filtered amplitudes for use in the diffusion scheme. (EB / 27.06.94 - 04.01.2009)

`VEK( F, N, XMIN, XMAX, A, IE, EPS) in: ../elibs/. :`
BULI-INTEGRATOR ZUR BERECHNUNG DES VEKTORS
A(I) = int(x=xmin...xmax) F(I,X) DX , I=1..N
EPS : MAXIMALE RELATIVE ABWEICHUNG ZWISCHEN DEN BEIDEN LETZTEN EXTRAPOLATIONSSCHRITTEN

```
IE : ZAHL DER EXTRAPOLATIONEN
SUBROUTINE F(N,X,V) IST SO ZU SCHREIBEN, DASS V(1:N) DIE N FUNKTIONSWERT
DER VEKTORFUNKTION F ENTHAELT.
N DARF NICHT GROESSER SEIN ALS MULTI (06.12.90)
```

`Vm_n( p ) in: qcm.f` :

```
vertical profile of Qn ( EB / 06.01.05 )
```

`Vm_s( p ) in: qcm.f` :

```
vertical profile of Qs ( EB / 06.01.05 )
```

`Vm_x( p ) in: qcm.f` :

```
vertical profile of Qx ( EB / 06.01.05 )
```

`wishcal( relL, ruhL, TKE, TPE, Srel, hd, vd, he, ve, ht ) in: edtrans.f` :

```
WISHCAL: compute 3D grid-space representations ( EB / 28.06.2006
)
added prerequisites for adabatic conversion (CZ/10.08.2010)
dito for dissipation (CZ/17.08.2010)
```

Called by `edtrans`.

`wishcal( nlat, dat ) in: hetrans.f` :

```
WISHCAL: compute 3D grid-space representations ( EB / 28.06.2006
)
added ad. conv (co) (CZ/ER / 10 Aug 2010)
```

Called by `hetrans`.

`wishcal( nlat, dat ) in: zobutrans.f` :

```
WISHCAL: compute zonal mean budgets from 3D grid-space representations
( Erich Becker / 26.08.2004 - 01.07.2010 )
included ad. conversion (sco, eco) (CZ/10.08.2010)
```

Called by `zobutrans`.

`worldoro-V.f in: ../.` :

```
extract spectral representation of the world orography from
the given T106 orographie with an adjustable spectral filter
for the new V-truncation ( EB / 14.10.2009 )
```

`xispec( ntco, sinfi, y) in: inidc.f` :

```
c y(n) = ue(fi,o) * DPN(n-1) * cosfi, PN = gew"ohnliches Legendre-Polyno
( EB / 22.11.94 )
```

```
zobutrans :
    ZOBUTRANS: Special afterburner for spectral of KMCM: zonal momentum
    and heat budget in the temporal and zonal mean on the model's
    hybrid layers.  ( EB / 30.06.2010 )
    calculated DDAY instead of reading; read both conventional and
    V-shaped triangulated data ( CZ / 06.07.2010 )
    added ad.  conversion (sco, eco) (CZ / 10.08.2010)
```

Transient zonal-mean budgets for GrADS (see sct. 4.4.3)

5.2 Common blocks

Common blocks:

```
/coAmeazon/Amean(0:npol,lev1), Ameas(0:npol,lev1), Ameax(0:npol,lev1)

/coamplis/awir(levels,2:modr), astr(levels,2:modr), adiv(levels,2:modr),
    apot(levels,2:modr), atem(levels,modr), aps(2:modr)

/coapsg/apsg(2:modr)

/coclock/day, clock, dayclock

/cocompost/a(0:levels), b(0:levels), bq(levels), dibq(levels), dib(levels-1),
    eta(0:levels)

/coconum/cocu(npar) --> Q0, Qn, Qs, Qx, p0, sp0, f0, sf0, pn, spn,
    fn, sfn, ps, sps, fs, sfs, px, spx, fx, sfx, x0(3), sx0(3),
    relzo, xn(2), sxn(2), alfn(2), shitn, xs(2), sxs(2), alfs(2),
    shits, xx(2), sxx(2), alfx(2), shitx

/codunki/dunknose, dunk, dunks

/cofpsi/xito, xitr, Btr, Btr_p

/cohogrid/ilon, jlat

/colatitude/ fi(kgau), sinfi(kgau), P(0:nz,0:nz,kgau), PX(0:nz,0:nz,kgau),
    PY(0:nz,0:nz,kgau), PL(0:nz,0:nz,kgau)

 /colatitude_FFT/QX(0:nz,0:nz,kgau), AY(0:nz,0:nz,kgau), AL(0:nz,0:nz,kgau)

/coNk/Nk(2:modr)

/constants/R, cp, cpd, Rcpd, g, pref, aerde, om, pi
```

```
/coOPA/OPA

/coorography/aoro(modr), ofac

/coPhil/oro(irie,kgau), Ts(irie,kgau), psg(irie,kgau), Qc(irie,lev1,kgau),
    Qm(irie,lev1,kgau), Te(irie,lev1,kgau), z0mask(irie,kgau), relaxd(lev)

/coQBO1/QK, QR, pK, pR, spK, spR, CK, CR, OK, OR, MK, MR

/coQBO2/AQK(lev,kgau), AQR(lev,kgau)

/corelaxt/relaxbo, relaxto, etas, detas

/coshift/shiftn(lev1), shifts(lev1), shiftx(lev1)

/cosuper/Teq_t, Tpo_t, b1_t, b2_t

/coTe1/x(nte) --> pbo, pto, feq, Teq, Tpo, fje, dsi, fh, dh, Ttr,
    Tto, ptr, pje, dp, Tsum, Twin, dpsum, dpwin, polni, dfisum,
    fiwin, dfiwin

/coTe2/b1, b2, s1, s2, s3, pid, xi0, xi1, xi2, psi, sequa

/coTe3/ pthermo, Tthermo, thermfac

/cotendencies/switch

/coTref/ pbo, ptr, pto, tbo, ttr, tto, xi0, xi1, xi2, psi

/cotrig/ trig(irie,0:2*nz), trigx(irie,0:2*nz), xlam(irie)

/seasurfacepressure/ p00

/stat_symdiff/ udif(irie,lev,kgau), vdif(irie,lev,kgau), Tdif(irie,lev,kgau)
```

5.3 Data files

Some data appear in files with fixed names:

Data files:

`(case).bst.(ctl,grads)` : GrADS input from `zobutrans` (see sct. 4.4.3)

`(case).edt.(ctl,grads)` : GrADS input from `edtrans` (see sct. 4.4.2)

`(case).hebt.(ctl,grads)` : GrADS input from `hetrans` (see sct. 4.4.4) `dc.bini` : : `kmcm` initial data for `MAIN`. This file may be generated with `inidc` or can be taken from a previous `kmcm` run.

```
nt,levels,irie,nt, zeit, y
```

`dc.new` : `kmcm` time series data from `kmcm`

```
nt,levels,irie,nt, { zeit, y }
```

`gau3.data` : Gauss latitudes from `gau3cal` for `kmcm`

```
data x/pa/; data w/pw/
```

`(file).oro.bin` : orography file

5.4 Parameter files

These are line-oriented fix-named input files functioning like namelists:

Parameter files:

`CONSTANTS` : `kmcm` constants (see sct. 4.3.1)

`dc.steer` : `gau3, kmcm` namelist (see sct. 4.2.1)

`RESOLUTION` : `kmcm` resolution parameters (see sct. 4.1.1)

6 Applications

Here are briefly described several KMCM applications along with publications from the user community.

6.1 Scientific publications

In temporal order of their publication:

6.1.1 The feedback of midlatitude waves onto the Hadley cell in a simple general circulation model. (Becker et al., 1997)

We present self-consistent comparisons of axissymmetric and 3-dimensional simulations of the tropospheric circulation under idealized physical conditions. These reveal a feedback of transient eddies onto the Hadley circulation which, first, strongly depends on equatorial heating and, second, is in the case of realistic heating quite different from prescribed eddy forcing. A quantitative estimation for the eddy-induced mixing of vorticity into the poleward flow of the Hadley cell is given. The proposed relation is in accordance with observations. It is derived from the computational result that eddy momentum flux convergence is of the same order as the equatorward flux of relative vorticity generated by the Hadley circulation. Evaluation of the local budgets of sensible heat gives rise to a clear picture of how the poleward transports due to Hadley circulation and transient eddies interlock. This mechanism is found to be essential for an interpretation of the eddy feedback.

6.1.2 The role of orographically and thermally forced stationary waves in the causation of the residual circulation. (Becker & Schmitz, 1999)

Several experiments performed with an idealized troposphere-stratosphere GCM are compared to estimate the impact of orography and prescribed local heat sources on the residual circulation in the northern winter stratosphere. It is found that only the combined action of orographic and thermal wave forcing in northern midlatitudes is capable of inducing a residual circulation reaching continuously from tropical to polar latitudes at stratospheric altitudes. Intensifications of the residual circulation in response to modified forcing of stationary waves are generally associated with, firstly, a reduced polar night jet accompanied by enhanced easterlies in low and summer hemispheric latitudes and, secondly, substantial warming of the polar night stratosphere completely compensated by cooling in the tropics and subtropics.

6.1.3 Symmetric stress tensor formulation of horizontal momentum diffusion in global models of atmospheric circulation. (Becker, 2001)

In climate and weather forecast models, small-scale turbulence in the free atmosphere is usually parameterized by horizontal diffusion of horizontal momentum. This study proposes a formulation that is based on a symmetric stress tensor. The advantage over conventional methods is twofold. First, the Eulerian law of angular momentum conservation is fulfilled. Second, a self-consistent formulation of the momentum and thermodynamic equations of motion becomes possible due to incorporation of the local frictional heating rate, that is, the proper dissipation. The importance of these issues is demonstrated by numerical experiments performed with a simple general circulation model. For example, the new scheme precisely accounts for the irreversible increase of total potential energy during the decay of a baroclinic life cycle. Also the stress generated by horizontal momentum diffusion is found to be significant in the angular momentum budget of multiple life cycle experiments.

6.1.4 Interaction between extratropical stationary waves and the zonal mean circulation. (Becker & Schmitz, 2001)

A troposphere-stratosphere simple GCM that simulates the wintertime general circulation with considerable accuracy is presented. The model is driven by temperature relaxation, additional prescribed tropical heating, self-induced heating in midlatitudes, and boundary layer mixing. The idealizations in diabatic heating are used to study the basic impacts of orographic and midlatitude thermal forcing of stationary waves on the zonally averaged northern winter climatology in the troposphere.

It is found that large-scale mountains in the winter hemispheric midlatitudes do generally lead to enhanced eddy feedback onto the Hadley circulation; that is, the tropical streamfunction maximum is reduced and the surface near equatorward flow is enhanced due to enhanced eddy heat flux from the subtropics into midlatitudes. In addition, the subtropical jet is reduced and shifted poleward due to enhanced eddy deceleration in the poleward branch of the Hadley cell.

A prescribed longitude dependence of self-induced heating in the winter extratropics gives rise to quite an opposite response with regard to Hadley cell, subtropical Eliassen-Palm flux divergence, and eddy heat flux from the subtropics into midlatitudes. The subtropical jet is displaced equatorward and the overall wave activity shifts poleward. These effects depend on the nonlinear nature of self-induced midlatitude heating.

6.1.5 Energy deposition and turbulent dissipation owing to gravity waves in the mesosphere. (Becker & Schmitz, 2002)

An attempt is made to define the thermodynamics of internal gravity waves breaking in the middle atmosphere on the basis of the energy conservation law for finite fluid volumes. Consistent with established turbulence theory, this method ultimately determines the turbulent dissipation to be equivalent to the frictional heating owing to the Reynolds stress tensor. The dynamic heating due to nonconservative wave propagation, that is, the energy deposition, consists of two terms: namely, convergence of the wave pressure flux and a residual work term that is due to the wave momentum flux and the vertical shear of the mean flow. Only if both heating terms are taken into account does the energy deposition vanish, by definition, for conservative quasi-linear wave propagation. The present form of the energy deposition can also be deduced from earlier studies of Hines and Reddy and from Lindzen.

The role of the axiomatically defined heating rates in the heat budget of the middle atmosphere is estimated by numerical experiments using a simple general circulation model (SGCM). The authors employ the theory of Lindzen to parameterize saturation of internal gravity waves, including the first theorem of Eliassen and Palm, to define the wave pressure flux. It is found that in the climatological zonal mean, the residual work and the simulated turbulent dissipation give maximum cooling/heating rates of 25 day^{-1} and 12.5 day^{-1} in the summer mesosphere/lower thermosphere. These values are small but not negligible against the major contributions to the heat budget.

Filtering out gravity wave perturbations in the thermodynamic equation reveals that the simulated dissipation, which is due to the shear of the planetary-scale flow, does not generally represent the total dissipation. The latter turns out to be dominated by the shear associated with gravity wave perturbations. Taking advantage of Lindzen, the total frictional heating can be calculated and is quantitatively consistent with in situ measurements of Lübken.

6.1.6 Frictional heating in global climate models. (Becker, 2003)

A new finite-difference formulation of the frictional heating associated with atmospheric vertical momentum diffusion is proposed. It is derived from the requirement that, according to the no-slip condition, the sum of internal and kinetic energy of a fluid column is not changed by surface friction. The present form is designed to be implemented along with the hybrid-coordinate differencing scheme of Simmons and Burridge.

The effects of incorporating frictional heating in general circulation models (GCMs) of the atmosphere are assessed by analyzing representative long-term January simulations performed with an idealized GCM. The model employs the proposed discretization of vertical terms as well as recently derived horizontal diffusion and dissipation forms. For the conventional definition of a GCM with no frictional heating, the climatological global energy budget yields a spurious thermal forcing of about 2 Wm^{-2}. In the equivalent new model definition, this shortcoming is reduced by two orders of magnitude. Moreover, the long-term global mean of the simulated frictional heating yields approximately 1.9 Wm^{-2}. This value is in agreement with both the residuum in the conventional case as well as with existing estimates of the net dissipation owing to synoptic and planetary waves.

6.1.7 Climatological effects of orography and land-sea heating contrasts on the gravity wave-driven circulation of the mesosphere. (Becker & Schmitz, 2003)

On the basis of permanent January simulations performed with an idealized general circulation model for the troposphere and middle atmosphere, the sensibility of the general circulation to orographic and thermal forcing of large-scale stationary waves is assessed. Gravity waves are parameterized following Lindzen's saturation theory. Up to the stratopause, present model results coincide with earlier estimates, confirming that the boreal winter zonal-mean climate does crucially depend on the combined action of orography and land-sea heating contrasts. Since, in turn, the propagation and breakdown of internal gravity waves is strongly modulated by the background horizontal winds, the mesospheric response to stationary wave forcing turns out be substantial as well. It is found that in the climatological zonal mean, a warmer polar night stratosphere is accompanied by lower temperatures in the mesosphere up to about 80 km. The temperature signal induced by stationary wave forcing changes sign again in the upper mesosphere/lower thermosphere, which, except for the polar night region, is globally heated up by $10 - 20$ K. This heating is weaker if the assumed Prandtl number for gravity wave-induced vertical diffusion is raised from 3 to 6.

The thermal effects in the mesosphere are interpreted in terms of a global weakening of the summer-to-winterpole residual circulation that occurs along with strongly diminished gravity wave drag, turbulent diffusion, and energy deposition in the northern winter mesosphere. The weakening of gravity wave effects in the presence of quasi-stationary planetary waves is dominated by reduced efficiency of gravity wave saturation in the mesosphere. That is, due to the more variable and, on average, reduced planetary-scale horizontal winds, gravity wave saturation is distributed over a greater depth and drops in altitude. On the other hand, enhanced critical level absorption of gravity waves in the lower stratosphere plays at most a secondary role. Furthermore, present model results suggest that the winter-summer asymmetry in gravity wave breakdown, which is well known from the northern mesosphere, may be absent or even reversed in the southern mesosphere.

6.1.8 Dependence of the annular mode in the troposphere and stratosphere on orography and land-sea contrasts. (Körnich et al., 2003)

Using an idealized general circulation model, we explore the influence of orography and land-sea heating contrasts on the development of Annular Modes (AM) under perpetual January conditions. In the troposphere, model experiments with different stationary wave forcing all show an internally induced AM, but with significantly different amplitudes. Only the combined forcing of orography and land-sea heating contrast leads to a localization of the AM similar to the North Atlantic Oscillation, and additionally, to a stratospheric AM and its downward propagation. A tropospheric feedback mechanism between the anomalies of zonal-mean wind and stationary waves applies in the simulation with the combined forcing only, where the coupling of orographically and thermally forced waves reproduces the observed stationary waves, and where the simulated diabatic heating supports the feedback mechanism. In contrast, transient forcing dominates the variability in the experiments without orography, comparable to the Southern AM.

6.1.9 Direct heating rates associated with gravity wave saturation. (Becker, 2004)

Analysis of filtering out subscale motions is applied for internal gravity waves. This leads to a new perspective of the planetary-scale sensible heat budget of the upper mesosphere/lower thermosphere. In line with previous results of Becker and Schmitz, the present paper recapitulates that the dissipation of gravity wave kinetic energy and the local adiabatic conversion of mean enthalpy into gravity wave kinetic energy cannot be neglected, and that the net effect of both cools the upper mesosphere/lower thermosphere. In addition, the importance of the wave entropy flux - an effect which is ignored in customary gravity wave parameterizations for global circulation models - is stressed. We show that, when evaluated on the basis of Lindzen's saturation assumption, the wave entropy flux convergence behaves like a vertical diffusion of the mean strati1cation, where the wave-induced diffusion coeffcient is involved with a Prandtl number of 2. This result imposes an upper bound of 2 for the effective Prandtl number which scales the combined entropy flux owing to turbulence and gravity waves. The direct heating rates generated by gravity wave saturation are assessed quantitatively, using an idealized general circulation model completed by a Lindzen-type gravity wave parameterization.

6.1.10 High Rossby-wave activity in austral winter 2002: Modulations of the general circulation of the MLT during the MaCWAVE/MIDAS northern summer program. (Becker et al., 2004)

The seasonally and zonally averaged effects of the anomalously strong Rossby-wave activity in austral winter 2002 are estimated by a sensitivity experiment using an idealized general circulation model. The approach focuses on the modulation of gravity-wave saturation via the altered mean winds in the mesosphere and MLT. In the northern summer MLT the effects consist of increased vertical gradients of the mean zonal wind and temperature, as well as downward shifts of the residual circulation and the gravity wave-induced frictional heating. All these signals agree reasonably with observations obtained during the MaCWAVE/MIDAS program 2002.

6.1.11 The role of stationary waves in the maintenance of the Northern Annular Mode as deduced from model experiments. (Körnich et al., 2006)

The influence of stationary waves on the maintenance of the tropospheric annular mode (AM) is examined in a simple global circulation model with perpetual January conditions. The presented model experiments vary in the configurations of stationary wave forcing by orography and land-sea heating contrasts. All simulations display an AM-like pattern in the lower troposphere. The zonal momentum budget shows that the feedback between eddies with periods less than 10 days and the zonal-mean zonal wind is generally the dominating process that maintains the AM. The kinetic energy of the high-frequency eddies depends on the stationary wave forcing, where orographic forcing reduces and thermal forcing enhances it. The AMs in the model experiments differ in the superposed anomalous stationary waves and in the strength of the zonally symmetric component. If only orographic stationary wave forcing is taken into account, the mountain torque decelerates the barotropic wind anomaly, and thus acts to weaken the AM. However, the combined forcing of orography and land-sea heating contrasts produces a feedback between the anomalous stationary waves and the AM that compensates for the mountain torque. The different behavior of the model experiments results from the fact that only the thermal forcing changes the character of the anomalous stationary waves from external Rossby waves for orographic forcing alone to vertically propagating waves that enable the feedback process through wave-mean flow interaction. Only with this feedback, which is shown to be due to linear zonal-eddy coupling, does the model display a strong AM with centers of action over the oceans. The main conclusions are that this process is necessary to simulate a realistic northern AM, and that it distinguishes the northern from the southern AM.

6.1.12 Nonlinear horizontal diffusion for GCMs. (Becker & Burkhardt, 2007)

The mixing-length-based parameterization of horizontal diffusion, which was originally proposed by Smagorinsky, is revisited. The complete tendencies of horizontal momentum diffusion, the associated frictional heating, and horizontal diffusion of sensible heat in spherical geometry are derived. The formulations are modified for the terrain-following vertical-hybrid-coordinate system in a way that ensures energy and angular momentum conservation at each layer. Test simulations with a simple general circulation model, run at T42 horizontal resolution and for permanent January conditions, confirm the conservation properties and highlight the enhancement of nonlinear horizontal diffusion in areas of high baroclinic activity. The simulated internal variability is dependent on the nature of the horizontal diffusion, with high-frequency variability being enhanced over the northern continents and low-frequency variability being increased (decreased) over the Pacific (Atlantic) Ocean when using nonlinear rather than linear diffusion. Locally reduced horizontal dissipation over Europe is compensated by increased dissipation owing to vertical diffusion, indicating the potential importance of nonlinear horizontal diffusion for gravity wave-resolving simulations. Inspection of the spectral energy reveals that the scheme needs to be modified in order to damp unbalanced ageostrophic motions at the smallest resolved scales more efficiently. A corresponding empirical modification is proposed and proves to work properly.

6.1.13 Sensitivity of the upper mesosphere to the Lorenz energy cycle of the troposphere. (Becker, 2009).

The concept of a mechanistic general circulation model that explicitly simulates the gravity wave drag in the extratropical upper mesosphere in a self-consistent fashion is proposed. The methodology consists of 1) a standard spectral dynamical core with high resolution, 2) idealized formulations of radiative and latent heating, and 3) a hydrodynamically consistent turbulent diffusion scheme with the diffusion coefficients based on Smagorinskyś generalized mixing-length formulation and scaled by the Richardson criterion. The model reproduces various mean and variable features of the wave-driven general circulation from the boundary layer to the mesopause region during January.

The dissipation of mesoscale kinetic energy (defined as the frictional heating due to the mesoscale flow) in the extratropical troposphere is found to indicate the tropospheric gravity wave sources relevant for the mesosphere/lower thermosphere. This motivates a sensitivity experiment in which the large-scale differential heating is perturbed such that the Lorenz energy cycle as measured by the globally integrated frictional heating becomes stronger. As a result, both the resolved gravity wave activity and the dissipation of mesoscale kinetic energy in the extratropical troposphere are amplified. These changes have strong remote effects in the summer mesopause region, where the gravity wave drag, the residual meridional wind, and the frictional heating shift to lower altitudes. Furthermore, temperatures decrease below the summer mesopause and increase farther up, which is accompanied by an anomalous eastward wind component around the mesopause.

6.1.14 Consistent scale interaction of gravity waves in the Doppler spread parameterization. (Becker & McLandress, 2009)

The standard Doppler spread parameterization of gravity waves, which was proposed by C.-O. Hines and has been applied in a number of middle atmosphere general circulation models, is extended by the inclusion of all effects associated with vertical diffusion. Here the Wentzel-Kramers-Brillouin (WKB) approximation is employed to calculate the vertical propagation of the wave spectrum in the presence of wave damping. According to the scale interaction between quasi-stationary turbulence and the larger nonturbulent flow, all vertical diffusion applied to the resolved flow should damp the parameterized gravity waves as well. Hence, the unobliterated part of the gravity wave spectrum is subject to diffusive damping by the following processes: 1) the background diffusion derived from the model's boundary layer vertical diffusion scheme, which may extend into the middle atmosphere, 2) molecular diffusion, and 3) the turbulent diffusion resulting from the truncation of the gravity wave spectrum by Doppler spreading, which thus feeds back on the unobliterated gravity waves. The extended Doppler spread parameterization is examined using perpetual July simulations performed with a mechanistic general circulation model. For reasonable parameter settings, the convergence of the potential temperature flux cannot be neglected in the sensible heat budget, especially in the thermosphere. Less gravity wave flux enters the model thermosphere when vertical diffusion is included, thus avoiding the need for artificial means to control the parameterized gravity waves in the upper atmosphere. The zonal wind in the tropical middle and upper atmosphere is found to be especially sensitive to gravity wave damping by diffusion.

7 Updates

(Version 1, Release 1, Ch. Zülicke, 12 October 2009) : first compilation of Erich Beckers habilitation (Becker, 2003a), Stefan Lossows webpages and Heiner Körnichs manual (file: kmcm1r1.tex)

(Version 1, Release 1, Update 1, Ch. Zülicke, 04 March 2010) : updated bibliography, included running header, completed sections (file: kmcm1r1u1.tex)

(Version 1, Release 1, Update 2, Ch. Zülicke, 12 March 2010) : included figures, added `edward` and `zobu` (file: kmcm1r1u2.tex)

(Version 1, Release 1, Update 3, Ch. Zülicke, 16 March 2010) : completed cross-references and definitions (file: kmcm1r1u3.tex)

(Version 1, Release 1, Update 4, Ch. Zülicke, 18 March 2010) : minor corrections (file: kmcm1r1u4.tex)

(Version 1, Release 1, Update 5, Ch. Zülicke, 28 June 2010) : included `inidc` and `worldoro-V`; completed description of radiation and latent heating; included common blocks in flow charts (file: kmcm1r1u5.tex)

(Version 1, Release 1, Update 6, Ch. Zülicke, 07 July 2010) : included `zobutrans` for `zobu` (file: kmcm1r1u6.tex)

(Version 1, Release 1, Update 7, Ch. Zülicke, 08 July 2010) : updated `hetrans` (file: kmcm1r1u7.tex)

(Version 1, Release 1, Update 8, Ch. Zülicke, 13 July 2010)

replaced `edward` with `edtrans` including nsmooth/jump sampling, ncut option and correct grads storage (changed: `edtrans.f`)

extended `hetrans` for adiabatic conversion term (changed: `hetrans.f`)

(Version 1, Release 1, Update 9, Ch. Zülicke, 17 August 2010)

minor correction in input file loop (changed: `hetrans.f`)

remark on PBS queueing system (changed: `kmcm.hebt.plot.csh`)

allowed for long input file names (changed: `edtrans.f, hetrans.f`)

included `edtrans.exe` in `Makefile` (changed: `Makefile`)

completed Fortran program documentation (added: `iapsg, fpsg in psgcal.f`)

included adiabatic conversion in all afterburners (changed: `edtrans.f, zobutrans.f, hetrans.f`)

unified use of `gridcal` and `truncate` (changed: `Makefile, edtrans.f, gridcal.f, hetrans.f, truncate.f, zobutrans.f`)

included zonal and eddy diffusion (changed: `edtrans.f`)

(Version 1, Release 1, Update 10, Ch. Zülicke, 02 Sep 2010)

included option to read hybrid level parameters a and b from the file `ecmwf_lev.data` (changed: `compost.f`)

References

Asselin, R., 1972: Frequency filter for time integrations. Mon. Wea. Rev. 100: 487 - 490.

Becker, E., G. Schmitz & R. Geprägs, 1997: The feedback of midlatitude waves onto the Hadley cell in a simple general circulation model. Tellus 49A: 182 - 199

Becker, E. & G. Schmitz, 1999: The role of orographically and thermally forced stationary waves in the causation of the residual circulation. Tellus 51A: 902 - 913.

Becker, E., 2000: Description of the Kühlungsborn Mechanistic general Circulation Model KMCM. Inofficial document. Available in the KMCM-package (kmcm/doc/kmcm_manual.pdf and kmcm_manual_fig.pdf) or from Heiner Körnich (heiner@misu.su.se).

Becker, E., 2001: Symmetric stress tensor formulation of horizontal momentum diffusion in global models of atmospheric circulation. J. Atmos. Sci. 58: 269 - 282.

Becker, E. & G. Schmitz, 2001: Interaction between extratropical stationary waves and the zonal mean circulation. J. Atmos. Sci. 58: 462 - 480.

Becker, E. & G. Schmitz, 2002: Energy deposition and turbulent dissipation owing to gravity waves in the mesosphere. J. Atmos. Sci. 59: 54 - 68.

Becker, E., 2003: Frictional heating in global climate models. Mon. Wea. Rev. 131: 508 - 520.

Becker, E., 2003a: Energetics of turbulent momentum diffusion and gravity wave breakdown in general circulation models of the atmosphere. Habilitation at the Leibniz Institute of Atmospheric Physics Kühlungsborn at Rostock University. ISSN 1615-8083.

Becker, E. & G. Schmitz, 2003: Climatological effects of orography and land-sea heating contrasts on the gravity wave-driven circulation of the mesosphere. J. Atmos. Sci. 60: 103 - 118.

Becker, E., 2004: Direct heating rates associated with gravity wave saturation. J. Atm. Solar-Terr. Phys. 66: 683 - 696.

Becker, E., A. Müllemann, F.-J. Lübken, H. Körnich, P. Hoffmann & M. Rapp, 2004: High Rossby-wave activity in austral winter 2002: Modulations of the general circulation of the MLT during the MaCWAVE/MIDAS northern summer program. Geophys. Res. Lett. 31: L24S03. doi:10.1029/2004GL019615.

Becker, E. & U. Burkhardt, 2007: Nonlinear horizontal diffusion for GCMs. Mon. Wea. Rev. 135: 1439 - 1454.

Becker, E., 2009: Sensitivity of the upper mesosphere to the Lorenz energy cycle of the troposphere. J. Atmos. Sci. 66: 647 - 666.

Becker, E. & C. McLandress, 2009: Consistent scale interaction of gravity waves in the Doppler spread parameterization. J. Atmos. Sci. 66: 1434 - 1449.

DKRZ (Deutsches Klimarechenzentrum), 1992: The ECHAM3 atmospheric general circulation model. Technical Report No. 6, DKRZ, Hamburg.

Holton, J. R., 1982: The role of gravity wave induced drag and diffusion in the momentum budget of the mesosphere. J. Atmos. Sci. 39: 791 - 799.

Holtslag, A. A. M. & B. A. Boville, 1993: Local versus nonlocal boundary-layer diffusion in a global climate model. J. Climate 6: 1825 - 1842.

Hoskins, B. J. & A. J. Simmons, 1975: A multi-layer spectral model and the semi-implicit method. Q. J. Roy. Met. Soc. 101: 637 - 655.

Hou, A. Y., 1993: The influence of tropical heating displacements on the extratropical climate. J. Atmos. Sci. 50: 3553 - 3570.

Körnich, H., G. Schmitz & E. Becker, 2003: Dependence of the annular mode in the troposphere and stratosphere on orography and land-sea contrasts. Geophys. Res. Lett. 30, 5: 1265, doi:1210.1029GL016327.

Körnich, H., G. Schmitz & E. Becker, 2006: The Role of Stationary Waves in the Maintenance of the Northern Annular Mode as Deduced from Model Experiments J. Atmos. Sci. 63, 11: 2931 - 2947.

Lindzen, R. S., 1981: Turbulence and stress owing to gravity wave and tidal breakdown. J. Geophys. Res. 86: 9707 - 9714.

Machenhauer, B. & E. Rasmussen, 1972: On the integration of the spectral hydrodynamical equations by a transform method. Rep. No. 3, Inst. Theor. Meteorol., Univ. Copenhagen.

Mak, M., 1994: Cyclogenesis in a conditionally unstable moist baroclinic atmosphere. Tellus 46A: 14 - 33.

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 J. Chen, 1991: The calculation of geopotential and the pressure gradient in the ECMWF atmospheric model: Influence on the simulation of the polar atmosphere and on temperature analyses. Q. J. R. Met. Soc. 117: 29 - 58.