

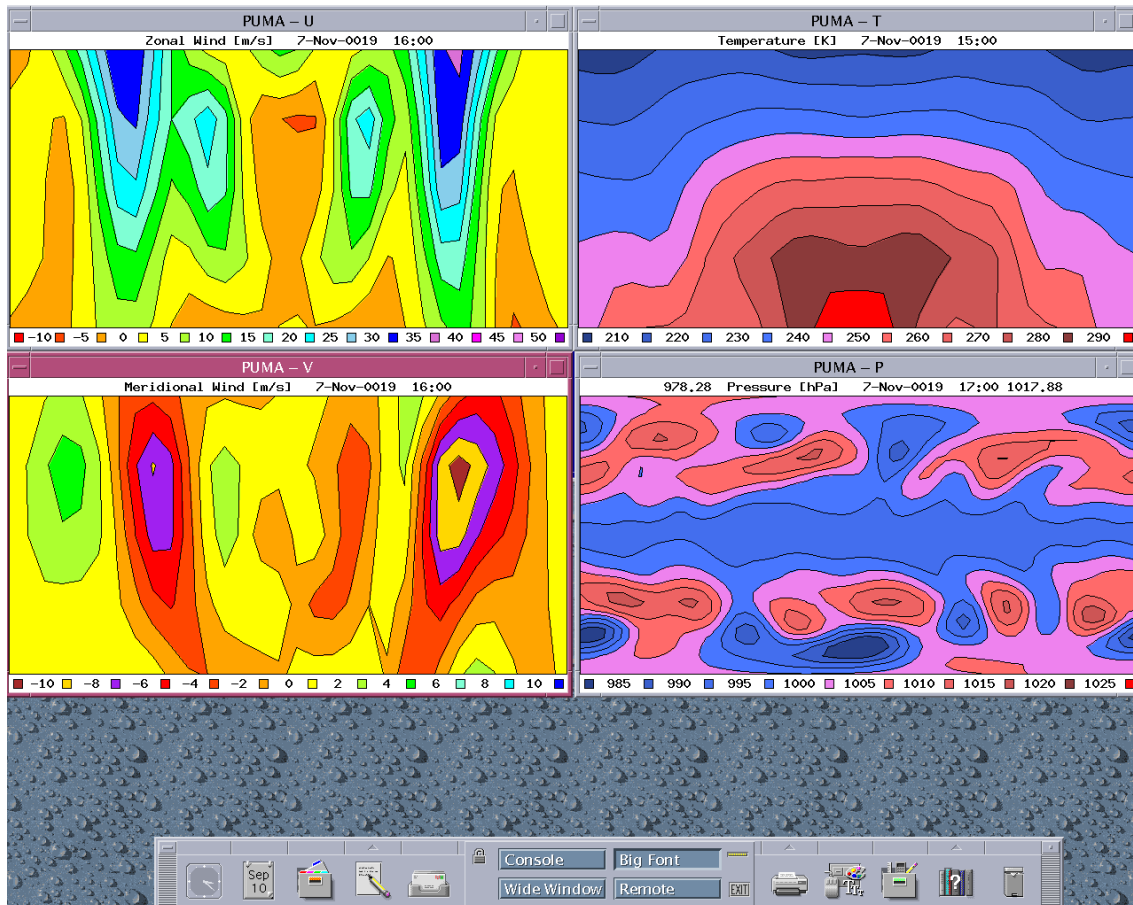


# PUMA



## Portable **U**niversity **M**odel of the **A**tmosphere

Klaus Fraedrich, Edilbert Kirk, Frank Lunkeit



Meteorologisches Institut  
Universität Hamburg  
Bundesstraße 55  
20146 Hamburg



---

**Contents**

<b>1</b>	<b>Introduction</b> .....	<b>4</b>
<b>2</b>	<b>Model Structure</b> .....	<b>7</b>
	Parameterizations .....	9
	Solution .....	10
	Vertical Discretization .....	11
<b>3</b>	<b>Code Structure</b> .....	<b>13</b>
	Initialization .....	14
	Computations .....	16
	Files .....	16
<b>4</b>	<b>User Manual</b> (with sample job) .....	<b>17</b>
<b>5</b>	<b>References</b> .....	<b>20</b>
<b>A</b>	<b>Names of Constants and Variables</b> .....	<b>22</b>
<b>B</b>	<b>Namelist of Input Parameters</b> .....	<b>26</b>
<b>C</b>	<b>PUMA Experiment: A Stormtrack Generated by a Heating Dipole</b> .....	<b>27</b>

## 1 Introduction

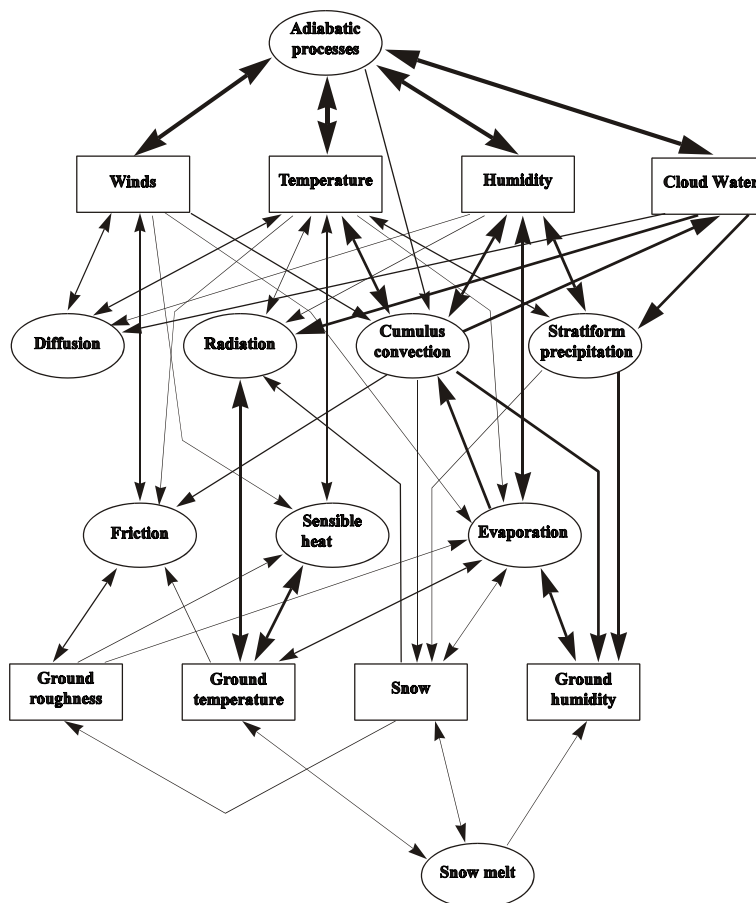
The Portable University Model of the Atmosphere (PUMA) is based on the Reading multi-level spectral model SGCM (Simple Global Circulation Model) described by Hoskins and Simmons (1975) and James and Gray (1986). Originally developed as a numerical prediction model, it was changed to perform as a circulation model. For example, James and Gray (1986) studied the influence of surface friction on the circulation of a baroclinic atmosphere, James and James (1992), and James et al. (1994) investigated ultra-low-frequency variability, and Mole and James (1990) analyzed the baroclinic adjustment in the context of a zonally varying flow. Frisius et al. (1998) simulated an idealized storm track by embedding a dipole structure in a zonally symmetric forcing field and Lunkeit et al. (1998) investigated the sensitivity of GCM (General Circulation Model) scenarios by an adaption technique applicable to SGCMs.

PUMA is introduced with following aims in mind: Training of junior scientists, compatibility with the ECHAM (European Centre - HAMburg) GCM, and scientific applications.

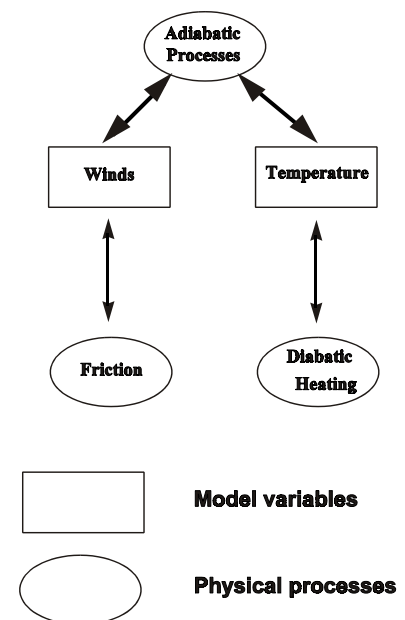
Training of junior scientists and students: PUMA contains only the main processes necessary to simulate the atmosphere. The source code is short and clearly arranged. A student can learn to work with PUMA within a few weeks, whereas full size GCMs require a team of specialists for maintenance, experiment design, and diagnostic.

Compatibility with the ECHAM model: PUMA is designed to be as compatible as possible with the comprehensive GCM ECHAM. Therefore, the same triangular truncation is employed and analogous transformation techniques like the Legendre- and fast Fourier transformation are used. Data from numerical experiments can be processed with the ECHAM postprocessor "afterburner", and all other common tools, statistical, and graphical packages, can be applied on PUMA data in the same manner as on ECHAM data. This allows PUMA to be used as a learning toolkit for potential ECHAM users.

Scientific applications: The PUMA code is the dynamical core of a GCM forced by Newtonian cooling and Rayleigh friction, such as proposed by Held and Suarez (1994) for the evaluation of the dynamical cores of GCMs. It forms the basis for various applications: (a) The code can be utilized to build and test new numerical algorithms (like semi-Lagrangian techniques). (b) Idealised experiments can be performed to analyse nonlinear processes arising from internal atmospheric dynamics (life cycles, etc.). (c) Data assimilation techniques can be incorporated to interpret results from GCM simulations or observations.



**Figure 1:** Processes in ECHAM.



**Figure 2:** Processes in PUMA.

Figure 1 demonstrates the complexity of interactions in a full size climate model, which leads to similarly complex response patterns on small parameter changes. The same diagram for PUMA (Figure 2) shows the direct paths, which allow an easy identification of cause and effect.

Modes of cooperation: The use of PUMA is not restricted to a limited number of research groups. Although it is not a public domain program, anyone interested in atmospheric circulation model experiments may obtain a copy of the source code including this documentation and use it freely. The only thing we ask for, is to give credit to the authors.

PUMA originates from the Simmons and Hoskins SGCM version, but has two major differences: (a) The code is rewritten in portable FORTRAN-90 code, which removes any problems associated with machine-specific properties like word lengths, floating point precision, output, etc. PUMA is a stand alone program, which does not use any external libraries. All necessary routines are in the source code, even the FFT (East Fourier Transform) and the matrix inversion. The model can now be run on a wide range of computer systems ranging from Pentium-PCs to vector/parallel supercomputers with a standard FORTRAN-90 compiler. (b) The truncation scheme is changed from jagged triangular truncation to standard triangular truncation that is compatible to other T-models like ECHAM. The model output data is written in a format compatible to the ECHAM-postprocessor “afterburner”. Thus all other diagnostic software can be used on PUMA data.

The outline of this PUMA documentation is as follows: After a brief description of the dynamics (section 2) the structure of the code is discussed (section 3), followed by the user manual (section 4). Results of a PUMA experiment are displayed in appendix C.

## 2 Model Structure

The model is based on the primitive equations, conveniently formulated in terms of the vertical component of absolute vorticity  $\zeta$  and the horizontal divergence  $D$ . Terrain following  $\sigma$ -coordinates ( $\sigma=p/p_s$ ) are used in the vertical, so that continuity is expressed as a prognostic equation for the logarithm of surface pressure,  $\ln(p_s)$ .

vorticity equation

$$\frac{\partial \zeta}{\partial t} = \frac{1}{1-\mu^2} \frac{\partial}{\partial \lambda} \mathcal{F}_v - \frac{\partial}{\partial \mu} \mathcal{F}_u - \frac{\xi}{\tau_F} - K(-1)^h \nabla^{2h} \zeta \quad (1)$$

divergence equation

$$\frac{\partial D}{\partial t} = \frac{1}{1-\mu^2} \frac{\partial}{\partial \lambda} \mathcal{F}_u + \frac{\partial}{\partial \mu} \mathcal{F}_v - \nabla^2 \left( \frac{U^2+V^2}{2(1-\mu^2)} + \Phi + T_R \ln p_s \right) - \frac{D}{\tau_F} - K(-1)^h \nabla^{2h} D \quad (2)$$

thermodynamic equation

$$\frac{\partial T'}{\partial t} = -\frac{1}{(1-\mu^2)} \frac{\partial}{\partial \lambda} (UT') - \frac{\partial}{\partial \mu} (VT') + DT' - \dot{\sigma} \frac{\partial T}{\partial \sigma} + \kappa \frac{T\omega}{p} + \frac{T_R - T}{\tau_R} - K(-1)^h \nabla^{2h} T' \quad (3)$$

continuity equation

$$\frac{\partial (\ln p_s)}{\partial t} = -\frac{U}{1-\mu^2} \frac{\partial (\ln p_s)}{\partial \lambda} - V \frac{\partial (\ln p_s)}{\partial \mu} - D - \frac{\partial \dot{\sigma}}{\partial \sigma} \quad (4)$$

hydrostatic equation

$$\frac{\partial \Phi}{\partial (\ln \sigma)} = -T \quad (5)$$

with

$$U = u \cos \phi = u \sqrt{1-\mu^2} \quad \mathcal{F}_u = V\zeta - \dot{\sigma} \frac{\partial U}{\partial \sigma} - T' \frac{\partial(\ln p_s)}{\partial \lambda}$$

$$V = v \cos \phi = v \sqrt{1-\mu^2} \quad \mathcal{F}_v = -U\zeta - \dot{\sigma} \frac{\partial V}{\partial \sigma} - T'(1-\mu^2) \frac{\partial(\ln p_s)}{\partial \mu}$$

The temperature  $T$  is split into a reference part,  $T_R(\sigma)$ , usually set to be constant at 250 K on all levels, plus an anomaly,  $T'$ . Further variables are:

$\zeta$	absolute vorticity	$\lambda$	longitude
$\xi$	relative vorticity	$\phi$	latitude
D	divergence	$\mu$	$\sin(\phi)$
$\Phi$	geopotential	$\kappa$	adiabatic coefficient
$\omega$	vertical velocity	$\tau_R$	timescale of Newtonian cooling
$p$	pressure	$\tau_F$	timescale of Rayleigh friction
$p_s$	surface pressure	$\sigma$	vertical coordinate $p/p_s$
K	hyperdiffusion	$\dot{\sigma}$	vertical velocity $d\sigma/dt$
u	zonal wind	v	meridional wind
h	hyperdiffusion order	$T_R$	restoration temperature
T	temperature	$T'$	$T - T_R$

All variables are made non-dimensional using the following characteristic scales:

Variable	Scale	Scale description
Divergence	$\Omega$	$\Omega = \text{angular velocity}$
Vorticity	$\Omega$	$\Omega = \text{angular velocity}$
Temperature	$(a^2 \Omega^2) / R$	$a = \text{planet radius, } R = \text{gas constant}$
Pressure	101325 Pa	PSURF = constant global mean pressure
Orography	$(a^2 \Omega^2) / g$	$g = \text{gravity}$



## 2.1 Parameterizations

Hyperdiffusion for a prognostic variable  $Q$  is represented by  $K(-1)^h \nabla^{2h} Q$ , where  $h$  is an integer (typically 3 or 4) and  $K$  is a diffusion coefficient. The hyperdiffusion is included for computational reasons in the vorticity (1), divergence (2), and temperature equation (3) in order to parameterize the energy cascading into subgrid scales and its subsequent dissipation.

Rayleigh friction is included in the vorticity (1) and the divergence equation (2):

$$\frac{\partial \zeta}{\partial t} + \mathcal{L}_\xi + N_\xi = -\frac{\xi}{\tau_F} \quad (6)$$

$$\frac{\partial D}{\partial t} + \mathcal{L}_D + N_D = -\frac{D}{\tau_F} \quad (7)$$

The timescale  $\tau_F$  can be defined for each horizontal layer; its vertical distribution has a maximum value at the surface decreasing upwards to zero. This is a simple linear approximation for surface drag and turbulent exchange of momentum in the boundary layer.

Newtonian cooling parameterizes the diabatic processes and is included in the thermodynamic equation (3) as a linear term:

$$\frac{\partial T'}{\partial t} + \mathcal{L}_T + N_T = \frac{(T_R - T)}{\tau_R} \quad (8)$$

It represents the process of radiative-convective heating in the free atmosphere. The model is set into motion as the model temperature  $T$  relaxes towards the restoration temperature  $T_R$  which, usually, has a large Equator to Pole gradient;  $\tau_R$  is the e-folding timescale of the Newtonian cooling.

## 2.2 Solution

The equations are solved using the spectral transform method (Orszag 1970, Eliassen et al. 1970). A variable  $Q$  is represented by a truncated series of spectral harmonics:

$$Q(\lambda, \mu) = \sum_{n,m} Q_n^m P_n^m(\mu) e^{im\lambda} \quad (9)$$

where  $n$  is the total wavenumber and  $m$  the zonal wavenumber with  $m \leq n$ ; the  $P_n^m(\mu)$  are the associated Legendre functions. The spectral transformation method relies on a transformation between  $Q$  and the spectral coefficients  $Q_n^m$ , and its inverse. Each timestep involves a transformation of the variables from spectral to gridpoint representation and back again. Linear terms are evaluated in the spectral domain and nonlinear products (such as  $U\zeta$ ) are evaluated in the gridpoint domain. The gridpoint domain provides the opportunity to introduce local parameterizations of radiation, convection adjustments, friction and so on. In PUMA all of these processes are represented by linear terms (Newtonian cooling and Rayleigh friction) and can therefore be solved in the spectral domain.

The Fast Fourier Transformation (FFT) provides an extremely fast transform in the zonal direction. The transforms  $Q^m(\mu)$ , which are the Fourier coefficients at each latitude, are calculated in this manner. From these, the spectral coefficients are obtained by integration with respect to  $\mu$ , using the orthogonality of the  $Q_n^m(\mu)$ :

$$Q_n^m = \int_{-1}^{+1} Q^m(\mu) P_n^m(\mu) d\mu \quad (10)$$

The transformation in  $\mu$  is carried out by Gaussian quadrature.

### 2.3 Vertical Discretization

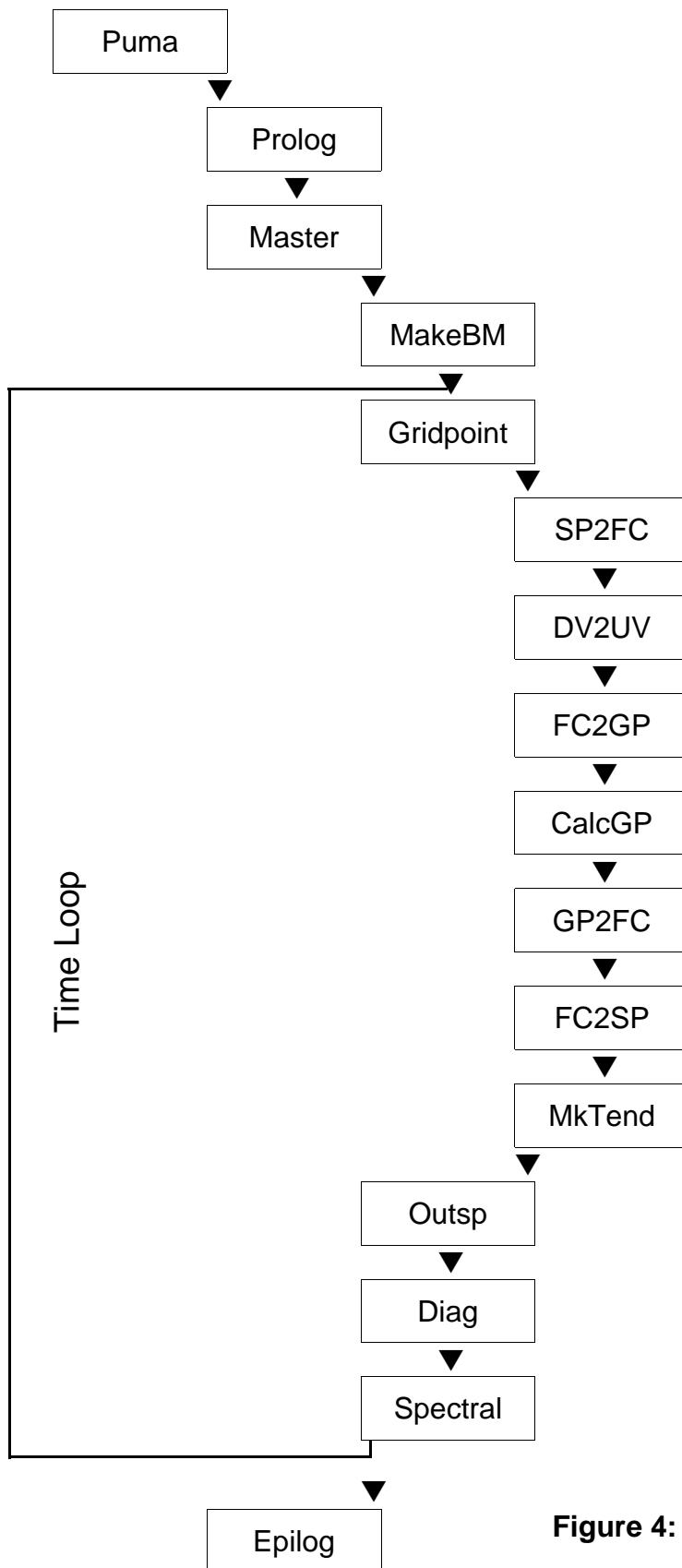
Level	$\sigma$		Variables
0.5	0	-----	$p=0 \quad \dot{\sigma}=0$
1	1	—————	$\zeta, D, T'$
1.5	2	-----	$\dot{\sigma}$
2	3	—————	$\zeta, D, T'$
2.5	4	-----	$\dot{\sigma}$
3	5	—————	$\zeta, D, T'$
3.5	6	-----	$\dot{\sigma}$
4	7	—————	$\zeta, D, T'$
4.5	8	-----	$\dot{\sigma}$
5	9	—————	$\zeta, D, T'$
5.5	10	-----	$p=p_s \quad \dot{\sigma}=0$

**Figure 3:** Vertical geometry of PUMA with associated variables (5 level version).

The model is represented by finite differences in the vertical (Figure 3); the number of  $\sigma$  levels can be preselected. The prognostic variables  $\zeta$ ,  $D$ , and  $T'$  are calculated at full-levels. At the half levels,  $\sigma = 0$  (upper boundary) and  $\sigma = 1$  (lower boundary), the vertical velocity is zero. The vertical advection at level  $r$  is approximated as follows:

$$\left(\overline{\dot{\sigma} \delta_{\sigma} Q_{\sigma}}\right)_r = \frac{1}{2} \left( \dot{\sigma}_{r+\frac{1}{2}} \frac{Q_{r+1} - Q_r}{\Delta \sigma} + \dot{\sigma}_{r-\frac{1}{2}} \frac{Q_r - Q_{r-1}}{\Delta \sigma} \right) \quad (11)$$

The tendencies of temperature, divergence and surface pressure are solved by the semi-implicit scheme with leap-frog time stepping. The vorticity equation is computed by the centred differences in time (Hoskins and Simmons, 1975).



**Figure 4:** Flow diagram of main routines.

### 3 Code Structure

The diagram (Figure 4) shows the route through the main program PUMA with names attached to the most important subroutines.

**PumaMod** is a module, that defines all global parameters, variables and arrays.

**Puma** is the main program. It calls the three subroutines **Prolog**, **Master** and **Epilog**.

**Prolog** does all initialization. It calls the following subroutines:

<b>gauaw</b>	computes Gaussian abscissas and weights.
<b>inilat</b>	initializes some utility arrays like square of cosine of latitude etc.
<b>legpri</b>	prints the arrays of <b>gauaw</b> and <b>inilat</b> .
<b>readnl</b>	reads the namelist from standard input.
<b>initpm</b>	initializes most vertical arrays and some in the spectral domain.
<b>initsi</b>	computes arrays for the semi-implicit scheme.
<b>legini</b>	computes all polynomials needed for the Legendre transform.
<b>restart</b>	starts the model from the restart file of a previous run, if selected.
<b>initfd</b>	initializes spectral arrays.
<b>setzt</b>	sets up the restoration temperature array.
<b>noise</b>	puts a selectable form of noise into $\ln(p_s)$ .
<b>setztex</b>	is a special version of <b>setzt</b> for dipole experiments.

**Master** On initial runs **master** does some initial timesteps, then it runs the time loop for the selected integration time. It calls the following subroutines:

<b>makebm</b>	constructs the array <b>bm</b> .
<b>gridpoint</b>	does all transformations and calculations in grid point domain.
<b>sp2fc</b>	spectral to Fourier coefficients (inverse Legendre transform).
<b>dv2uv</b>	divergence, vorticity to u and v (implies spectral to Fourier).
<b>fc2gp</b>	Fourier coefficients to grid points (fast Fourier transform).
<b>calcgp</b>	calculations in grid point space.
<b>gp2fc</b>	grid points to Fourier coefficients (fast Fourier transform).
<b>fc2sp</b>	Fourier coefficients to spectral (direct Legendre transform).
<b>mktend</b>	make tendencies (implies Fourier to spectral).
<b>spectral</b>	does all calculations in the spectral domain.
<b>outsp</b>	writes spectral fields in physical dimensions on an output file.
<b>diag</b>	writes selected fields and parameters to standard output.

**Epilog** writes the restart file.

### 3.1 Initialization

The model starts either from a restart file or with an atmosphere at rest. The defaults make the initial state a motionless atmosphere with stable stratification. At the start the divergence and the relative vorticity are set to zero (only the vorticity mode(1,0) is set to the planetary vorticity). The temperature is initialized as a horizontally constant field, the vertical distribution is adopted from the restoration temperature of stable stratification. The initialization of the logarithm of the surface pressure is controlled by the namelist variable **kick**: **kick=0** sets all modes to zero; the model runs zonally constant without eddies. **kick=1** generates random white noise, **kick=2** generates random white noise that is symmetric to the Equator. Runs started with **kick=1** or **2** are irreproducible due to the randomization. For reproducible runs with eddies use **kick=3** which initializes only the mode(1,2) of  $\ln(P_s)$  with a small constant. The amplitude of the noise perturbation is normalized to 0.1 hPa, that is 1/10,000 of the mean surface pressure.

A restoration temperature field for the run is set up by **setzt**: First, a global mean restoration temperature profile  $T_R(\sigma)$  is defined. A hyperbolic function of height is used to provide  $T_R$ , as illustrated in Figure 4. With  $z \rightarrow -\infty$  the profile tends to a uniform lapse rate, (**alr**), passing through the temperature (**tgr**) at  $z = 0$ . With  $z \rightarrow +\infty$  the profile becomes isothermal. The transition takes place at the height (**ztrop**). The sharpness of the tropopause is controlled by the parameter (**dttrp**). When (**dttrp** = 0), the lapse rate changes discontinuously at (**ztrop**). For (**dttrp**) small but positive, the transition zone is spread. The hydrostatic relation is used to determine the heights and hence the temperatures of the model levels.

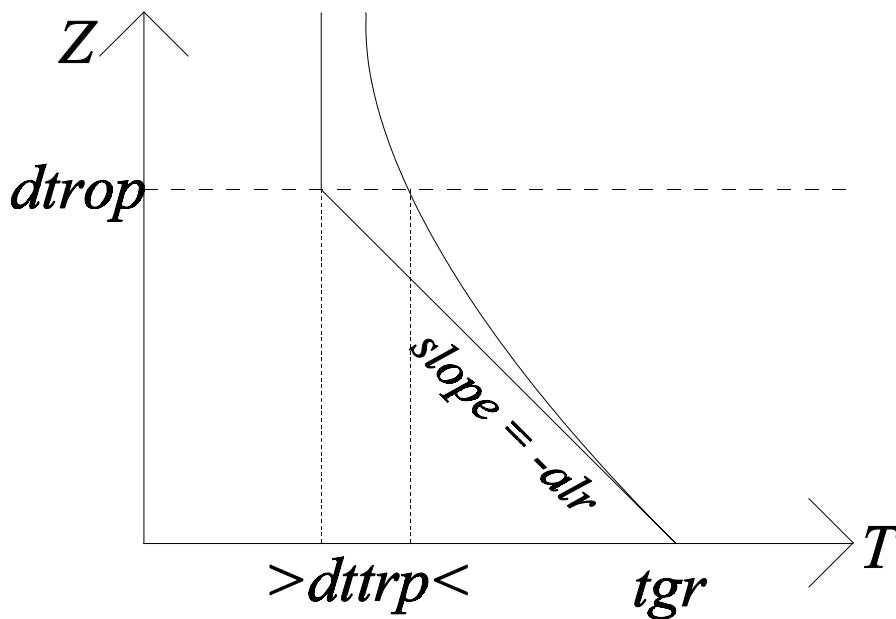
The restoration temperature is set to:

$$T_R(\phi, \sigma) = T_R(\sigma) + f(\sigma) \left( \Delta T_{NS} \frac{\mu}{2} - \Delta T_{EP} \left( \mu^2 - \frac{1}{3} \right) \right) \quad (13)$$

The function  $f(\sigma)$  attains small values near the upper boundary and has the following general form:

$$\begin{aligned} f &= \sin \left( \frac{\pi (\sigma - \sigma_T)}{2 (1 - \sigma_T)} \right) && \text{for } \sigma \geq \sigma_T \\ f &= 0 && \text{for } \sigma < \sigma_T \end{aligned} \quad (14)$$

where  $\sigma_T$  is the value of  $\sigma$  at the height (**ztrop**), calculated within this subroutine. The North Pole - South Pole temperature difference is defined by  $\Delta T_{NS}$ , and  $\Delta T_{EP}$  is the Equator - Pole temperature difference. Note that  $\Delta T_{EP}$  should be positive for the common case where the Poles are colder than the Equator.



**Figure 5:** Specifying the tropopause and the restoration temperature  $T_R$  from input parameters.

### 3.2 Computations

The subroutine **spectral** performs one timestep. Details of the time stepping scheme are given in Hoskins and Simmons (1975). The adiabatic tendencies (advection, etc.) are used. The normal timestep is centered in time and includes a Robert time filter to control time splitting. For the first short initial **nkits** timesteps, a forward timestep is followed by a centred step, each twice as long as its predecessor to initiate a run from data at a single time level. The Robert time filter is not included in the initial steps. The subroutine also calculates the spectral tendencies due to Newtonian cooling, Rayleigh friction, and hyperdiffusion:

The Newtonian cooling is the relaxation of the temperature field **st** towards the restoration temperature ( $T_R$ ) fields **sr**, which is calculated in subroutine **setzt** or **setztex**. The relaxation is driven by the timescale parameter **damp**, set by the namelist parameter **restim**. The Newtonian cooling is added to the spectral temperature tendency **stt**.

The Rayleigh friction is the damping of the spectral divergence **sd** and vorticity **sz** arrays by the timescale **tfric**, a predefined namelist variable. Rayleigh friction is added to the spectral tendencies **sdt** and **szt**.

Hyperdiffusion: The temperature, divergence, and vorticity (**st**, **sd**, **sz**) are multiplied by the hyperdiffusion term; the order is defined by the namelist parameter **ndel**. Hyperdiffusion is added to the according spectral tendencies (**stt**, **sdt**, **szt**).

#### Table of subroutines with input/output:

Subroutine **readnl** reads the namelist from **stdin**.

Subroutine **diag** writes some diagnostics to **stdout**.

Subroutine **restart** reads restart data from **fort.10**.

Subroutine **epilog** writes restart data to **fort.12**, containing **nstep**, **svo**, **sd**, **st**, **spn**, **svomi**, **sdmi**, **stmin**, **spmin**, **sgs** and **stres**.

Subroutine **outsp** writes at selected time intervals data records to **fort.40**, containing the variables divergence, vorticity, temperature and logarithm of surface pressure.



## 4 User Manual

### Compiling

PUMA may be compiled with any standard FORTRAN-90 compiler.

#### Sample compile command:

```
f90 -o puma puma.f
```

Use the "-g" option for debugging and the "-O" option for performance.

#### Sample run command:

```
puma <namelist >diag
```

The namelist is read from file "namelist" the diagnostic output is written to file "diag".

#### Sample CRAY T-3D job: (for the parallelized version of PUMA)

```
1 #QSUB -l mpp_p=16
2 #QSUB -l mpp_t=7200
3 #QSUB -q m3
4 #QSUB -eo
5 #QSUB -o /pf/u/u236001/wait_queue/Pumarun0.out
6 #
7 exp=37000
8 count=`cat $HOME/puma/count|cut -d, -f1`
9 month=`cat $HOME/puma/count|cut -d, -f2`
10 yr=`cat $HOME/puma/count|cut -d, -f3`
11 year=$yr
12 iter=24
13 ARCHIV=/mf/u/u236001/puma
14 HIST=$ARCHIV/${exp}/hist
15 cd $TMPDIR
16 assign -s unblocked -a fort.40 u:40
17 assign -s unblocked -a $HIST u:10
18 assign -s unblocked -a fort.12 u:12
19 puma=/pf/u/u236001/puma/puma
20 namelist=/pf/u/u236001/puma/namelist
21 namelistr=/pf/u/u236001/puma/namelistr
```

```
22  if [ $count -eq 1 ]
23  then
24    if [ $month -lt 10 ]
25    then
26      month=0${month}
27    fi
28    if [ $yr -lt 10 ]
29    then
30      year=0${yr}
31    fi
32    $puma -npes 16 < $namelist
33    mv fort.12 $HIST
34    m2ieeee fort.40 $ARCHIV/${exp}/${exp}_${year}${month}
35    count=`expr $count + 1`
36    month=`expr $month + 1`
37    echo $count,$month,$yr > $HOME/puma/count
38    qsub $HOME/puma/Pumarun0
39    exit
40  elif [ $count -le $iter ]
41  then
42    if [ $month -lt 10 ]
43    then
44      month=0${month}
45    fi
46    if [ $yr -lt 10 ]
47    then
48      year=0${yr}
49    fi
50    $puma -npes 16 < $namelistr
51    mv fort.12 $HIST
52    puma2ieeee fort.40 $ARCHIV/${exp}/${exp}_${year}${month}
53    count=`expr $count + 1`
54    month=`expr $month + 1`
55    if [ $month -eq 13 ]
56    then
57      month=1
58      yr=`expr $yr + 1`
59    fi
60    echo $count,$month,$yr > $HOME/puma/count
61    qsub $HOME/puma/Pumarun0
62    exit
```

63 fi  
64 exit

The model output can be analysed by using the AFTERBURNER. This program is processing model data represented by T21, T42, T63 and T106 resolution in gridpoint and spectral domain.

On the CRAY computers of the DKRZ the path to "afterburner" is: /pf/k/k204004/burn/. The file <mod.doc> contains the online documentation in ASCII-format. The executable program is file <after>.

*Acknowledgments:* The authors wish to thank Ute Luksch, Frank Sielmann, Thomas Frisius, Christoph Raible, Philip Sura, Christian Franzke, Katrin Walter, and Magnus Bornemann for their contributions to this report.

This work was supported by the BMBF through contract BMBF 07 KFT 121/1 "Hochauflösende Langzeitsimulationen der atmosphärischen Zirkulation auf massiv parallelen Rechnerarchitekturen".

## 5 References

- Edmon, H. J., B. J. Hoskins, and M.E. McIntyre, 1980:** Eliassen-Palm cross sections for the troposphere. *J. Atmos. Sci.*, **37**, 2600-2616. Erratum: **38**, 1115.
- Eliassen E., B. Machenhauer, and E. Rasmussen, 1970:** On a numerical method for integration of the hydrodynamical equations with a spectral representation of the horizontal fields. *Inst. of Theor. Met.*, Univ. of Copenhagen, **2**.
- Frisius T., F. Lunkeit, K. Fraedrich, and I. N. James, 1998:** Storm track organization and variability in a simplified atmospheric global circulation model. *Q. J. R. Meteorol. Soc.*, **124**, 119-143.
- Grotjahn, R., 1993:** Global atmospheric circulations. *Oxford University Press*, New York, 430pp.
- Held, I. M. and M. J. Suarez, 1994:** A proposal for the intercomparison of the dynamical cores of atmospheric general circulation models. *Bull. Amer. Meteor. Soc.*, **75**, 1825-1830.
- Hoskins B. J. and A. J. Simmons, 1975:** A multi-layer spectral model and the semi-implicit method. *Q. J. R. Meteorol. Soc.*, **101**, 637-655.
- James I. N. and L. J. Gray, 1986:** Concerning the effect of surface drag on the circulation of a planetary atmosphere. *Q. J. R. Meteorol. Soc.*, **112**, 1231-1250.
- James I. N. and P. M. James, 1992:** Spatial structure of ultra-low-frequency variability of the flow in a simple atmospheric circulation model. *Q. J. R. Meteorol. Soc.*, **118**, 1211-1233.
- James, I. N., 1994:** Introduction to circulating atmospheres. *Cambridge University Press (Cambridge)*, 422 pp.

- James P. M., K. Fraedrich, and I. N. James, 1994:** Wave-zonal-flow interaction and ultra-low-frequency variability in a simplified global circulation model. *Q. J. R. Meteorol. Soc.* **120**, 1045-1067.
- Lunkeit F., K. Fraedrich, and S. E. Bauer, 1998:** Storm tracks in a warmer climate: Sensitivity studies with a simplified global circulation model. *Clim. Dyn.* **14**, in press.
- Mole N. and I. N. James, 1990:** Baroclinic adjustment in a zonally varying flow. *Q. J. R. Meteorol. Soc.* **116**, 247-268.
- Peixoto, J. P. and A. H. Oort, 1992:** Physics of climate. American Institute of Physics, 520 pp.
- Orszag, S. A., 1970:** Transform method for calculation of vector coupled sums. *J. Atmos. Sci.*, **27**, 890-895.
- Roeckner, E., K. Arpe, L. Bengtson, S. Brinkop, L. Dümenil, M. Esch, E. Kirk, F. Lunkeit, M. Ponater, B. Rockel, R. Sausen, U. Schlese, S. Schubert, M. Windelband, 1992:** Simulation of present-day climate with the ECHAM model: Impact of model physics and resolution. *Max-Planck Report* **93**.
- Roeckner, E., K. Arpe, L. Bengtson, M. Christoph, M. Claussen, L. Dümenil, M. Esch, M. Giorgetta, U. Schlese, U. Schulzweida, 1996:** The atmospheric general circulation model ECHAM-4: Model description and simulation of present-day climate. *Max-Planck Report* **218**.
- Ulbrich, U. and M. Ponater, 1992:** Energy cycle diagnosis of two versions of a low resolution GCM. *Meteorol. Atmos. Phys.*, **50**, 197-210.

## A Global Constants and Variables

All global constants and variables are declared in the module PUMAMOD

```

module pumamod

! *****
! * Global Constants *
! *****

parameter(NTRU = 21)           ! Truncation
parameter(NLAT = 32)          ! Latitudes
parameter(NLEV = 5)           ! Number of levels
parameter(NLON = NLAT + NLAT) ! Number of longitudes
parameter(NHOR = NLON * NLAT) ! Horizontal part
parameter(NLEM = NLEV - 1)    ! Levels - 1
parameter(NLEP = NLEV + 1)    ! Levels + 1
parameter(NLSQ = NLEV * NLEV) ! Levels squared
parameter(NTP1 = NTRU + 1)    ! Truncation + 1
parameter(NRSP =(NTRU+1)*(NTRU+2)) ! No of real global modes
parameter(NCSP = NRSP / 2)    ! No of complex global modes
parameter(NVCT = 2 * (NLEV+1)) ! Dim of Vert. Coord. Tab
parameter(NROOT = 0)          ! Master node
parameter(AKAP = 0.286)       ! Kappa
parameter(ALR = 0.0065)       ! Lapse rate
parameter(EZ = 1.63299310207) ! ez = 1 / sqrt(3/8)
parameter(GA = 9.81)          ! Gravity
parameter(GASCON = 287.0)     ! Gas constant
parameter(PI = 3.14159265359) ! Pi
parameter(TWOPI = PI + PI)    ! 2 Pi
parameter(PLARAD = 6371000.0) ! Planet radius
parameter(PNU = 0.02)         ! Time filter
parameter(PNU21 = 1.0 - 2.0*PNU) ! Time filter 2
parameter(PSURF = 101325.0)   ! Surface pressure [Pa]
parameter(WW = 0.00007292)    ! Rotation speed
parameter(CV = PLARAD * WW)    ! velocity scale
parameter(CT = CV*CV/GASCON)  ! temperature scale

```

```
! *****
! * Global Integer Scalars *
! *****

integer :: kick      = 1 ! add noise for kick > 0
integer :: nafter    = 12 ! write data interval
integer :: ncoeff    = 0 ! number of modes to print
integer :: ndel      = 8 ! order of n - diffusion
integer :: ndiag     = 12 ! write diagnostics interval
integer :: nexper    = 0 ! experiment number
integer :: nexper    = 0 ! 1: dipole experiment
integer :: nkits     = 3 ! number of initial timesteps
integer :: nrestart  = 0 ! 1: read restart file 0: initial run
integer :: nrun      = 0 ! if (nstop == 0) nstop = nstep + nrun
integer :: nstep     = 0 ! current timestep
integer :: nstop     = 0 ! finishing timestep
integer :: ntspd     = 24 ! number of timesteps per day

! *****
! * Global Real Scalars *
! *****

real :: amco = 50.0
real :: amwo = 50.0
real :: ascl = 0.0
real :: ascn = 10.0
real :: ascr = 8.0
real :: ascs = 0.0
real :: aswl = 8.0
real :: aswn = 0.0
real :: aswr = 0.0
real :: asws = 10.0
real :: delt      ! 2 pi / ntspd timestep interval
real :: delt2     ! 2 * delt
real :: dtep = 60.0
real :: dtns = 0.0
real :: dtrop = 12000.0
real :: dttrp = 2.0
real :: hco = 10.0
real :: hwo = 10.0
real :: pclam = 145.0
real :: pcphi = 50.0
real :: pwlam = 180.0
real :: pwphi = 40.0
real :: tdiss = 0.25 ! diffusion time scale [days]
real :: tgr = 288.0 ! Temperature ground in mean profile
```

```

real :: wco   =   10.0
real :: wwo   =   10.0

! *****
! * Legendre Polynomials *
! *****

! polix : indirect transformation (spectral to fourier )
! poldx : direct transformation (fourier to spectral)
! x=p : polynomials
! x=d : deviation polynomials

real :: plip(NLAT,NCSP),plid(NLAT,NCSP)
real :: pliu(NCSP,NLAT),pliv(NCSP,NLAT)
real :: pldp(NLAT,NCSP),pldd(NLAT,NCSP)
real :: pldc(NLAT,NCSP),pldq(NLAT,NCSP)
real trig(NLON)

! *****
! * Global Spectral Arrays *
! *****

real :: sd(NRSP,NLEV) = 0.0 ! Spectral Divergence
real :: st(NRSP,NLEV) = 0.0 ! Spectral Temperature
real :: sz(NRSP,NLEV) = 0.0 ! Spectral Vorticity
real :: sp(NRSP)      = 0.0 ! Spectral Pressure (ln Ps)
real :: so(NRSP)      = 0.0 ! Spectral Orography
real :: sr(NRSP,NLEV) = 0.0 ! Spectral Restoration Temperature
real :: sdt(NRSP,NLEV) = 0.0 ! Spectral Divergence Tendency
real :: stt(NRSP,NLEV) = 0.0 ! Spectral Temperature Tendency
real :: szt(NRSP,NLEV) = 0.0 ! Spectral Vorticity Tendency
real :: spt(NRSP)      = 0.0 ! Spectral Pressure Tendency
real :: sdm(NRSP,NLEV) = 0.0 ! Spectral Divergence Minus
real :: stm(NRSP,NLEV) = 0.0 ! Spectral Temperature Minus
real :: szm(NRSP,NLEV) = 0.0 ! Spectral Vorticity Minus
real :: spm(NRSP)      = 0.0 ! Spectral Pressure Minus
real :: sak(NRSP)      = 0.0 !
real :: span(NRSP)     = 0.0 ! Pressure for diagnostics
real :: spnorm(NRSP)  = 0.0 ! Factors for output normalization
integer :: nindex(NRSP) = NTRU ! Holds wavenumber

! *****
! * Global Gridpoint Arrays *
! *****

real gd(NHOR,NLEV),gt(NHOR,NLEV),gz(NHOR,NLEV)
real gu(NHOR,NLEV),gv(NHOR,NLEV),gp(NHOR)

```



```
real rcsq(NHOR)

! *****
! * Diagnostic Arrays *
! *****

integer :: ndl(NLEV) = 0
real csu(NLAT,NLEV), csv(NLAT,NLEV), cst(NLAT,NLEV)

! *****
! * Latitude Arrays *
! *****

character *3 chlat(NLAT)
real csq(NLAT), si(NLAT), gw(NLAT)

! *****
! * Level Arrays *
! *****

real :: restim(NLEV) = 15.0
real :: t0(NLEV) = 250.0
real :: tfrc(NLEV) = (/ (0.0,i=1,NLEM), 1.0 /)
real :: vct(NVCT) = 0.0
real damp(NLEV)
real dsigma(NLEV)
real rdsig(NLEV)
real sigma(NLEV)
real sigmah(NLEV)
real t0ls2(NLEV)
real tkp(NLEV)
real c(NLEV,NLEV)
real g(NLEV,NLEV)
real tau(NLEV,NLEV)
real bml(NLEV,NLEV,NTRU)

end module pumamod
```

## B Namelist

The characteristics of the run can be set by means of a namelist INP which is read in by subroutine **readnl**. Usable defaults have been set for these variables.

Variable	Defaults	Meaning
<b>kick</b>	<b>1</b>	0: no noise initialization ( $p_s = \text{const.}$ ) 1: random white noise 2: Equator symmetric random white noise. 3: mode (1,2) no random initialization
<b>nafter</b>	<b>12</b>	output interval for data [timesteps]
<b>ncoeff</b>	<b>0</b>	number of coefficients to print in <b>wrspam</b>
<b>ndel</b>	<b>8</b>	order of hyperdiffusion ( $2 * h$ , see section 2.1)
<b>ndiag</b>	<b>12</b>	output intervall for diagnostics [timesteps]
<b>nexp</b>	<b>0</b>	experiment identifier
<b>nexper</b>	<b>0</b>	1: dipole added to $T_R$
<b>nkits</b>	<b>3</b>	number of short initial timesteps
<b>nrestart</b>	<b>0</b>	1: perform a restart run
<b>nrun</b>	<b>0</b>	number of timesteps to run (excl. initial ones)
<b>nstep</b>	<b>0</b>	current timestep
<b>nstop</b>	<b>0</b>	stop at timestep <b>nstop</b>
<b>ntspd</b>	<b>24</b>	number of timesteps per day
<b>dtep</b>	<b>60.0</b>	Equator - Pole temperature difference at surface for $T_R$
<b>dtns</b>	<b>0</b>	Pole to Pole temperature difference at surface for $T_R$ can simulate winter / summer hemispheres
<b>dttrp</b>	<b>2.0</b>	a temperature increment which controls the sharpness of the tropopause in $T_R$
	<b>12000.0</b>	height of tropopause in [m]
<b>tdiss</b>	<b>0.25</b>	dissipation time in sidereal days for wavenumber
<b>tgr</b>	<b>288.0</b>	global mean temperature of ground used to set $T_R$
<b>ndl(NLEV)</b>	<b>NLEV * 0</b>	1: activate spectral printouts for this level
<b>restim(NLEV)</b>	<b>NLEV * 15.0</b>	restoration timescale for each level
<b>t0(NLEV)</b>	<b>NLEV * 250.0</b>	reference $T_R$ -temperature profile
<b>tfrc(NLEV)</b>	<b>0,0,0,..,1</b>	Rayleigh friction timescale $\tau_F$ in days for each levels

## C PUMA Experiment: A Stormtrack Generated by a Heating Dipole

This appendix demonstrates the model simulating the dynamics of a single stormtrack, which is generated by a heating dipole anomaly. The mean and eddy fields of basic model variables are shown, as well as some standard analyses of its dynamics.

**Experimental Design:** In PUMA the diabatic heating is parameterized by Newtonian relaxation. The timescales involved are comparable with Held and Suarez (1994) and James et al. (1994):  $\tau_R = 5$  days at  $\sigma = 0.9$ , and  $\tau_R = 10$  days at  $\sigma = 0.7$ , and 30 days elsewhere. The rates in the upper layers reflect the slower radiative relaxation; in the lower layers they characterize turbulent heat exchange with the underlying surface of an aqua planet. Rayleigh friction exists only at the bottom layer with the time scale of  $\tau_F = 1$  day. The simulation of a single storm track requires changes of the zonally symmetric restoration temperature distribution  $T_R$  by superimposing an anomaly dipole (C-1 and C-2). Note that the restoration temperature cannot be interpreted as a climatological mean temperature. For example, a Newtonian cooling term of the order of 5K/day, which is required to balance the temperature advection in lower levels, corresponds to a  $(T_R - T)$  - difference of about 25 K given a 5 day relaxation time. Over the warm (cold) anomaly the magnitude of the vertical  $T_R$  - temperature gradient is increased (reduced) to -9.4 K/km (-5.5 K/km) at  $\sigma = 0.7$ . A tropopause is introduced at 12 km height above which the vertical temperature gradient vanishes. The vanishing temperature difference between North Pole and South Pole (namelist parameter **dtns** = 0, see Section 4) represents spring or autumn conditions, contrasting ECHAM 3 control runs for winter and summer (Roeckner et al. 1992 and 1996). The experiment is performed for 101 years dismissing the first spin-up year. The following climatological analysis is displayed in figures C-1 to C-8.

**First Moments:** The climatological temperature, wind and pressure or height fields are shown in zonally averaged cross-sections (C-1) and in hemispheric distributions (C-2).

The climatological mean *temperature* (C-1b) in the tropics is about 5 K (10 K) too warm at 900 hPa (in the stratosphere); near the Poles it is about 20 K too cold compared with observations (Roeckner et al 1992; figures 6 and 7). The horizontal distribution of the simulated 900 hPa temperature shows a warm anomaly in the south-western sector upstream of the warm part of the dipole (C-2b), which is associated with the diabatic heating (C-2a and C-2c).

The zonally averaged mean *zonal wind* shows maximum intensity of 35 m/s (C-1d), comparable with ECHAM 3 (Roeckner et al. 1992, figure 8 and 9). The zonal winds in the stratosphere are not reduced which is a consequence of the low vertical resolution of 5 levels. The jet lies in the south-eastern sector downstream of the warm pool (C-2e). The zonally averaged mean *surface pressure* (C-1f) shows almost realistic values for the subtropical highs and in the mid-latitudes. The polar troughs do not exist in the model (Roeckner et al. 1992, figure 12), which may be associated with missing gravity wave drag parameterization. The equatorial trough is too deep.

The zonally averaged mean *meridional circulation* is represented by the mass stream function (C-1e). The model reproduces the position and magnitude of the Hadley and the Ferrel cells reasonably well (Peixoto and Oort 1992, figure 7.19). The meridional wind field at 300 hPa (C-2e) reveals a stationary wave pattern with the largest amplitude near the heating regions. Cyclonic (anti-cyclonic) shear of the meridional wind is found close to the cold (warm) anomaly near the surface.

**Second Moments:** Further climatological analysis extends to standard deviations of the geopotential height and the meridional transports of sensible heat and momentum. Unfiltered, band-pass (between 2.5 and 6 days) and low-pass (greater than 10 days) filtered data are presented. The climatology of the *geopotential height fluctuations* (C-3a), which includes the corresponding filtered fields (C-5a and d), can be compared with the ECHAM control run (Roeckner et al. 1992, figure 23 to 25). Only the band-pass filtered part is underestimated which may be attributed to the lack of moist processes in PUMA.

The *meridional transport of sensible heat* near the surface is in the right order of magnitude; the secondary maximum near the tropopause is missing due to the low vertical resolution (C-3b, see Roeckner et al. 1992, figure 35 to 37). The *meridional transport of westerly momentum* by transient eddies is underestimated (C-3c, see Roeckner et al. 1992, figure 31 to 33). High eddy activity can be localized downstream of the dipole; in particular the band-pass filtered field (C-6a to c) shows the observed characteristics of the northern hemisphere storm tracks. Northward heat flux attains its largest value upstream of the pronounced height variance maximum; further downstream the pattern of westerly momentum transport reflects the barotropic decay of the synoptic eddies. The low-pass filtered data (C-6d to f) indicate variability on time scales longer than the typical life of a synoptic system. Part of this low-frequency

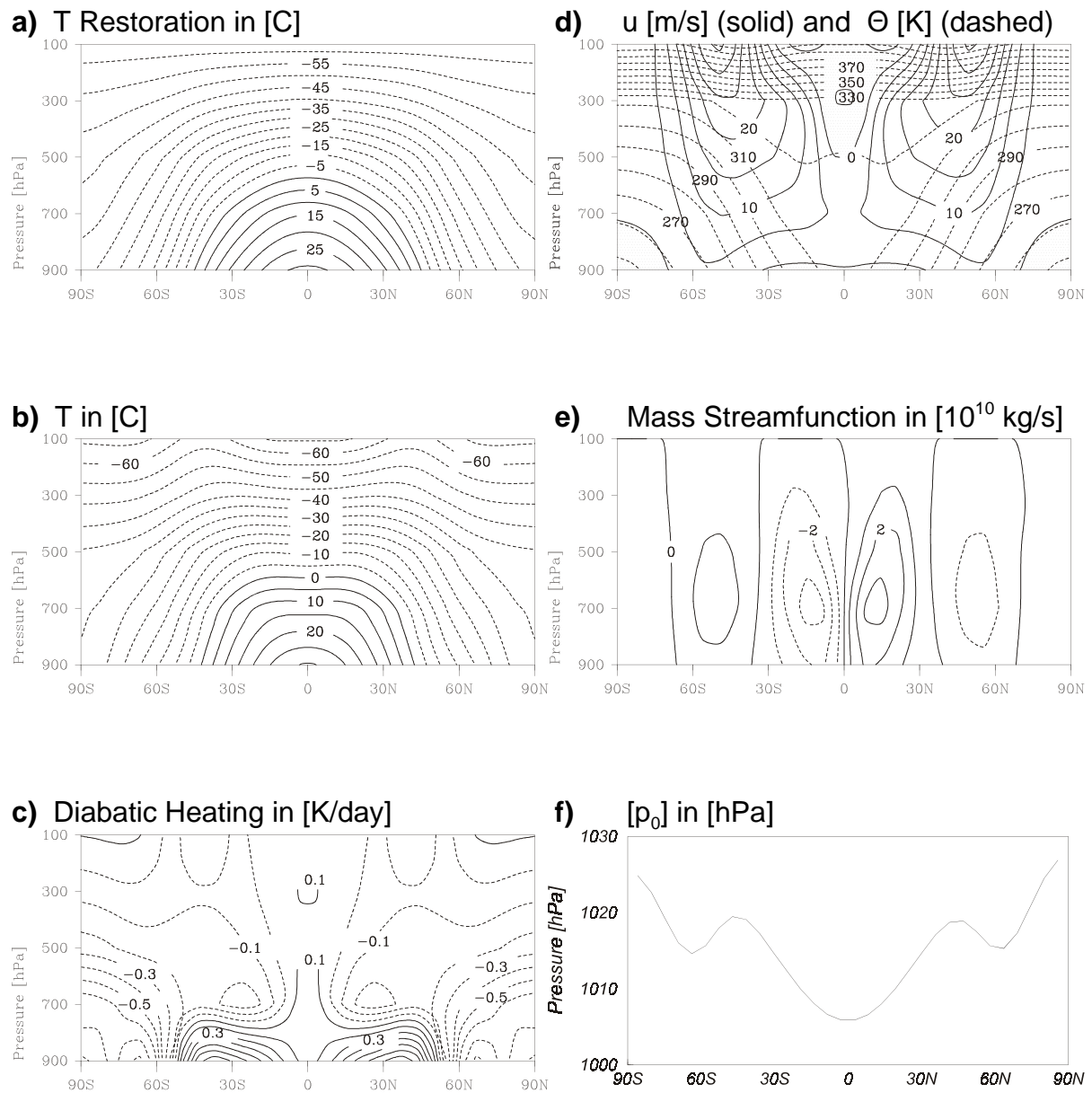
variability is associated with retrogressive large scale waves (Frisius et al. 1998) related to the growth and decay of a blocking anticyclone developing downstream of the storm track.

**Processes:** The Lorenz energy cycle, the Eliassen-Palm flux, and the transformed Eulerian mean circulation comprise various aspects of atmospheric dynamics:

The *Lorenz energy cycle* (C-7) provides the global mean energy reservoirs with their generations and conversions. The eddy contributions are separated into stationary and transient parts of the ultra-long wavenumbers (1 to 3) and the synoptic scale waves (4 to 10). Comparison with ECHAM (Ulbrich and Ponater 1992, figure 1 and 2) shows similar magnitudes for the available potential energy, but the eddy kinetic energy and the energy conversion between stationary and transient eddies are too small. Note, however, that estimates of the energy conversions tend to differ considerably (Grotjahn 1993, figure 4.26). Some energy conversions vary by more than a factor three. To a lesser extent, this also holds for energy reservoirs.

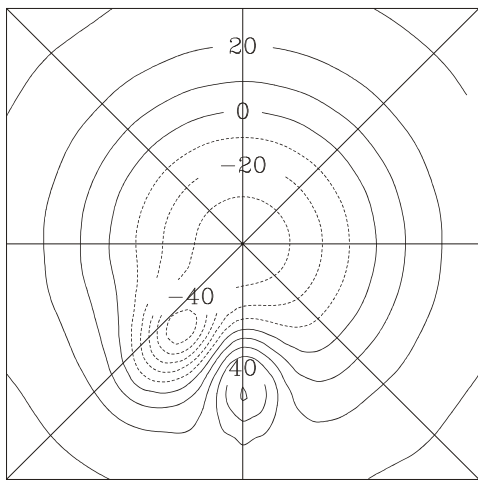
The *Eliassen-Palm (EP) flux* and its divergence are further standard diagnostic tools used to document the model performance (C-8). In the lower troposphere the components of the EP-vector reveal poleward temperature fluxes and equatorward momentum fluxes in the upper troposphere. The eddy fluxes of heat and momentum are larger than the stationary ones. The distribution of the EP-flux divergence resembles that of the linear Charney mode of baroclinic instability with maximum temperature fluxes at the bottom surface (Edmon et al., 1980, figure 2). While the EP-flux cross-sections appear to agree reasonably well with observations (Peixoto and Oort 1992, figure 14.9), major disagreement lies in the absence of a near-surface flux divergence (C-8a and b) which, presumably, is a consequence of the coarse vertical resolution and strong damping in PUMA.

The *transformed Eulerian mean circulation* (C-8c) is a residual mean meridional mass stream function and solely a response to the gradients of the heating, because the Eulerian mean circulation due to eddy fluxes has been removed by the transformation. As in observations (Edmon et al 1980, figure 6) the heating induces a single thermally direct overturning in both hemispheres. In the mid-latitudes this circulation appears to be too strong compared to observations, which is due to the large poleward temperature fluxes near the surface.

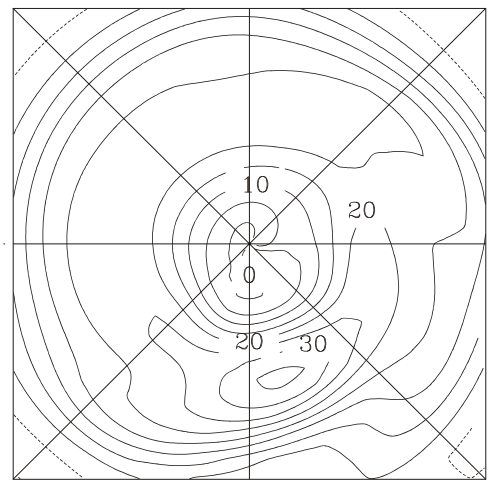


**Figure C-1:** Zonal mean cross-sections of climatological averages.

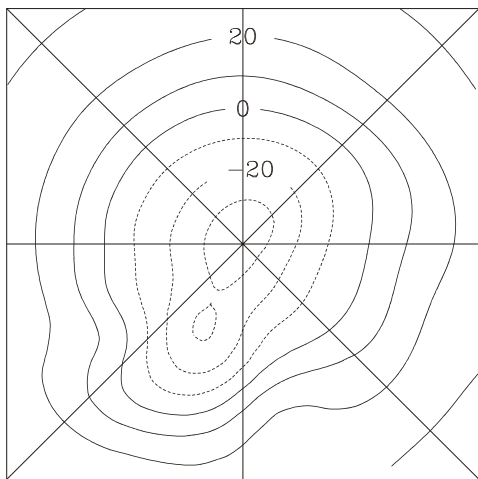
**a) T Restoration 900 hPa in [C]**



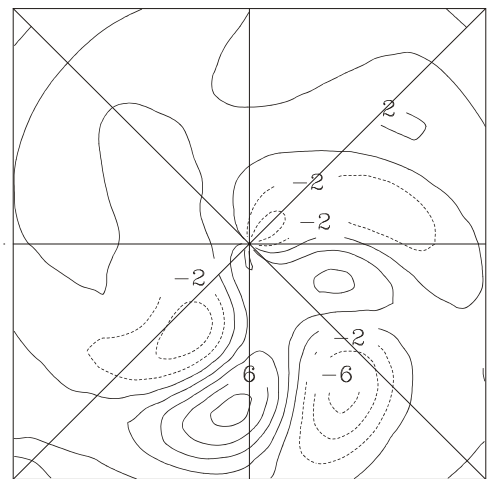
**d) u 300 hPa in [m/s]**



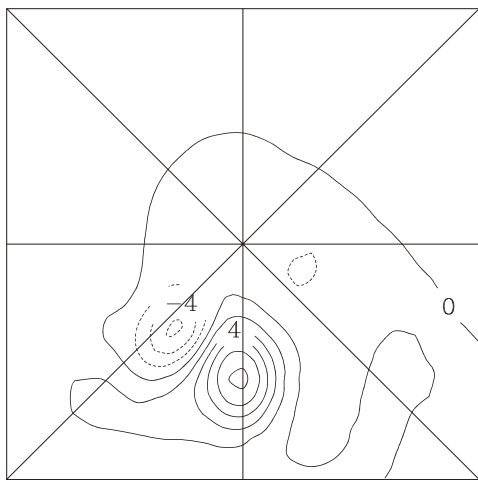
**b) T 900 hPa in [C]**



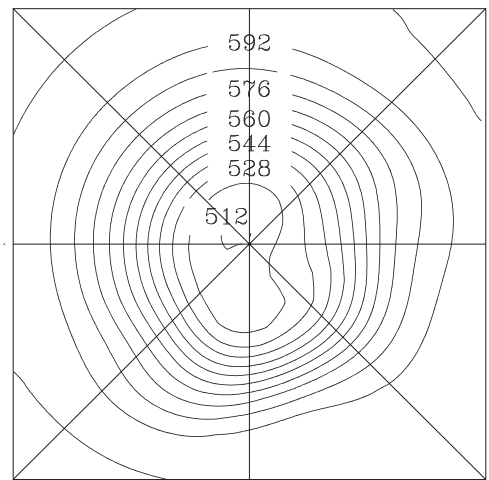
**e) v 300 hPa in [m/s]**



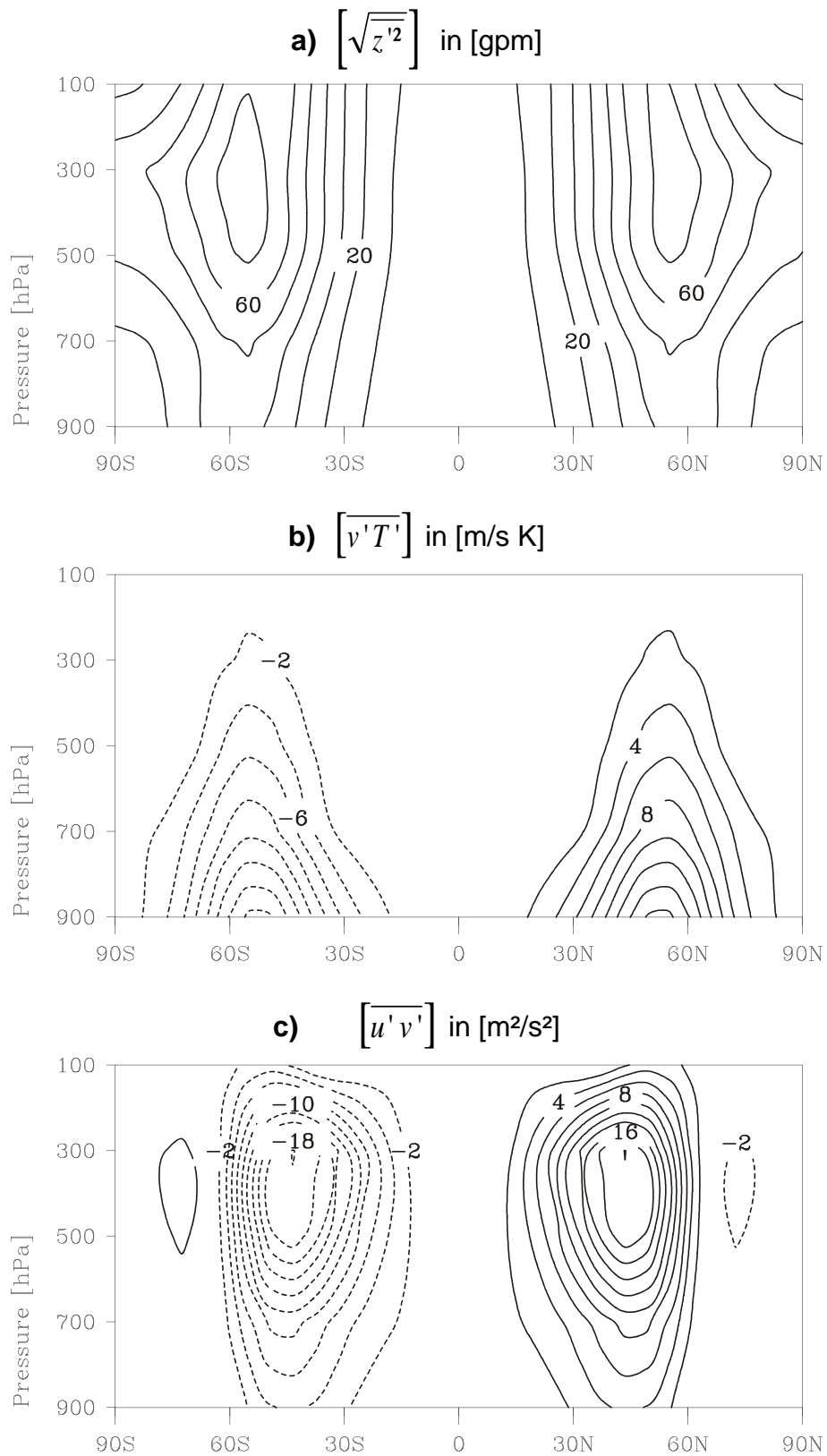
**c) Diabatic heating 900 hPa in [K/day]**



**f) z 500 hPa in [gpdam]**

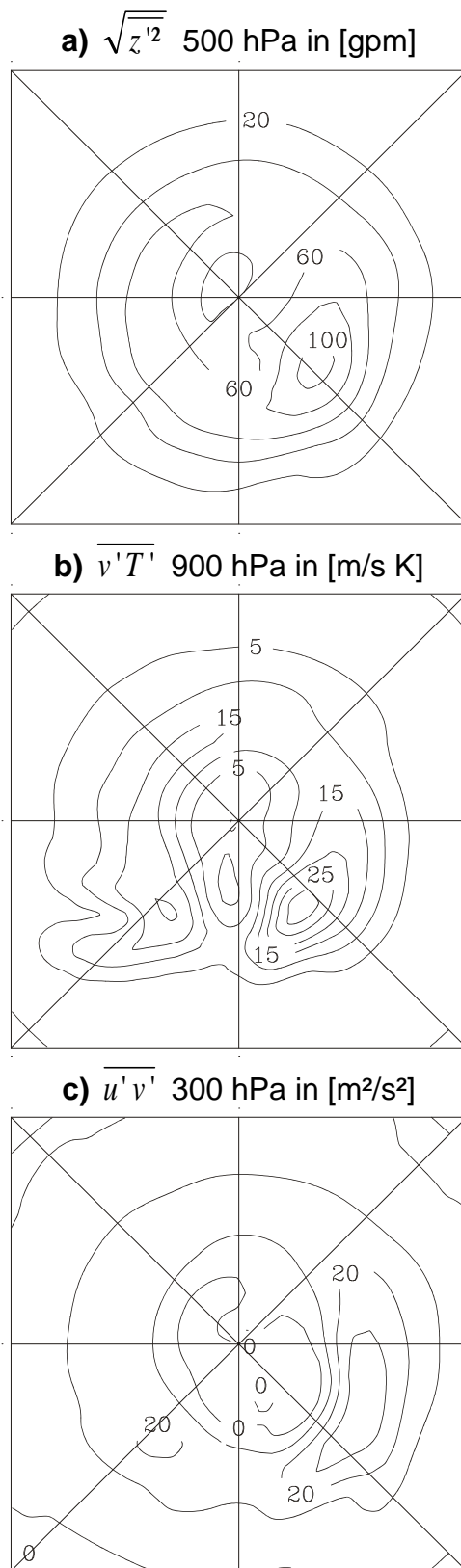


**Figure C-2: Hemispheric fields of climatological averages.**

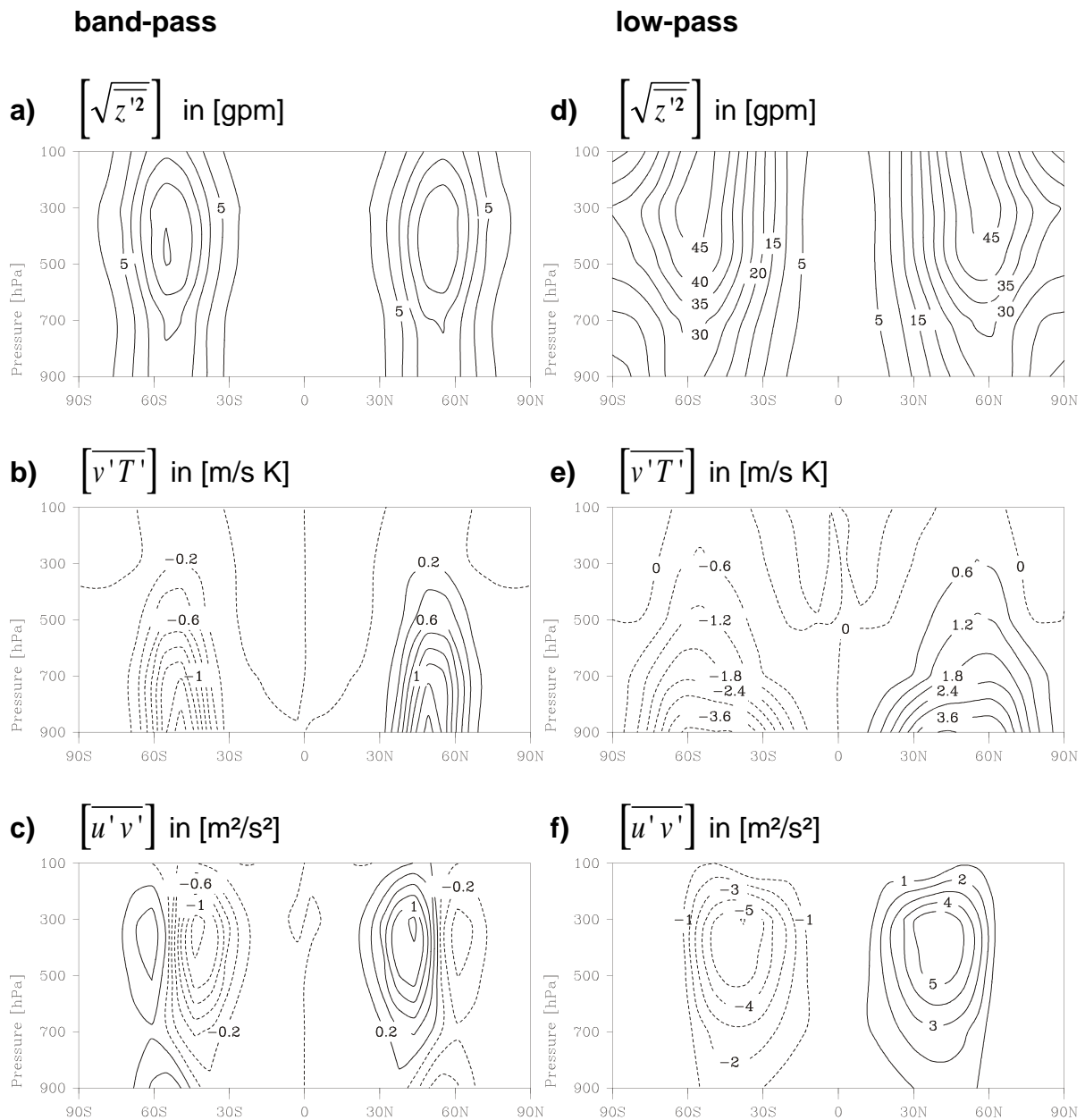


**Figure C-3:** Zonal and time mean cross-sections of transient eddies





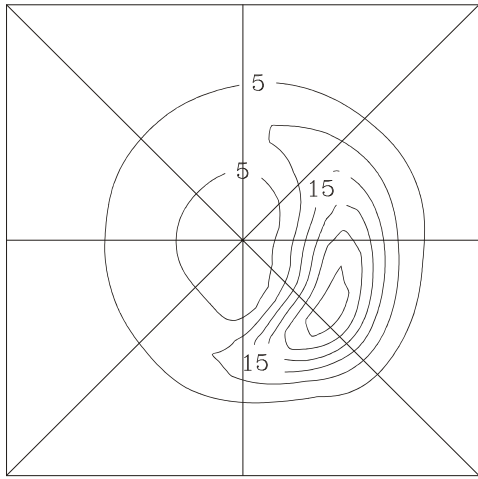
**Figure C-4:** Hemispheric fields of climatological averages of transient eddies.



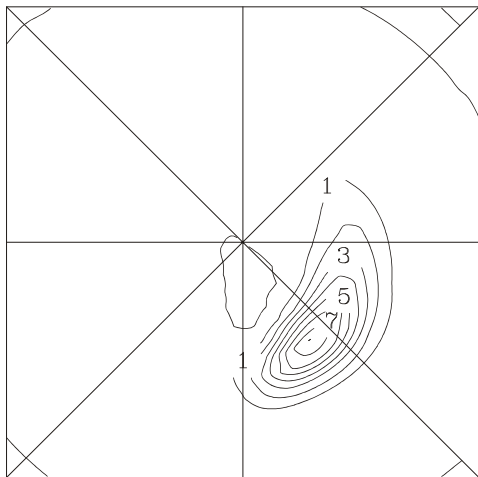
**Figure C-5:** Zonal and time mean cross-sections of band-pass (left column) and low-pass (right column) filtered transient eddies (Blackmon filter).

**band-pass**

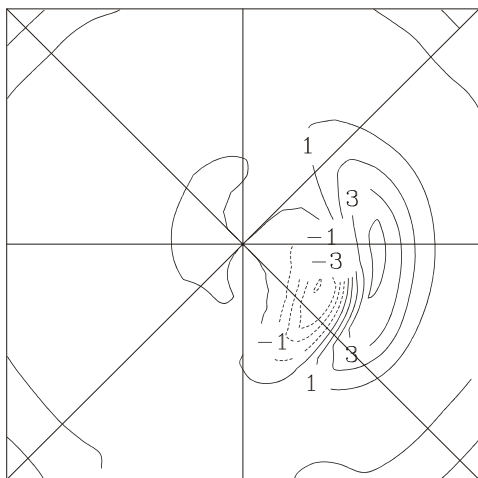
**a)**  $\sqrt{z'^2}$  500 hPa in [gpm]



**b)**  $\overline{v'T'}$  900 hPa in [m/s K]

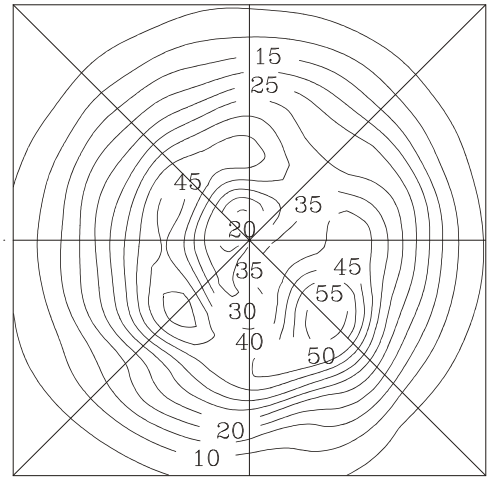


**c)**  $\overline{u'v'}$  300 hPa in [m<sup>2</sup>/s<sup>2</sup>]

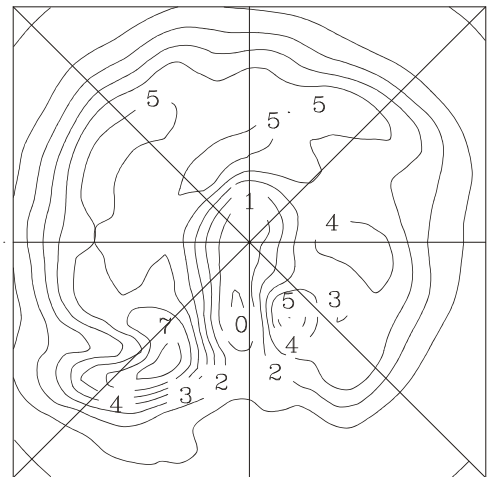


**low pass**

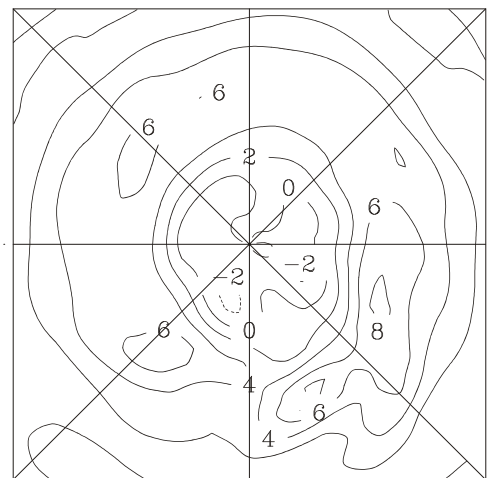
**d)**  $\sqrt{z'^2}$  500 hPa in [gpm]



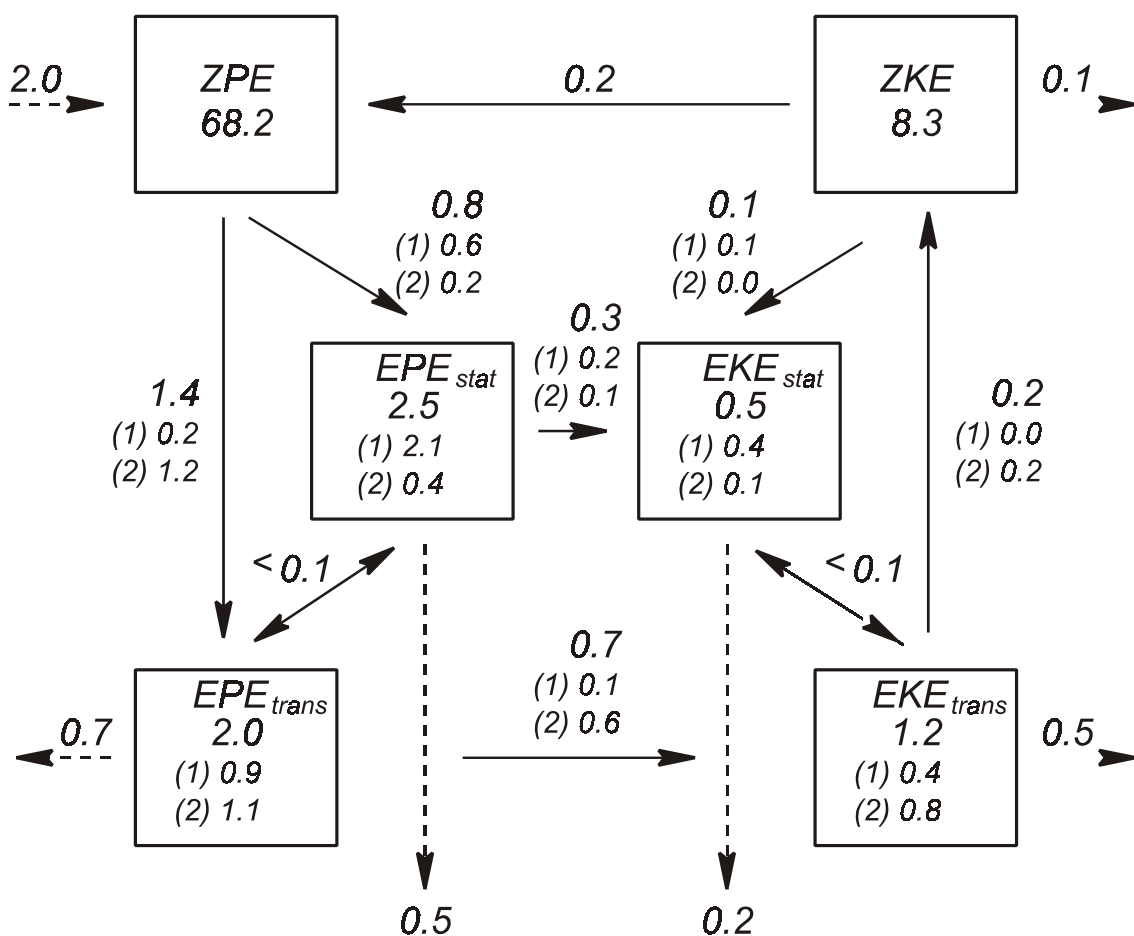
**e)**  $\overline{v'T'}$  900 hPa in [m/s K]



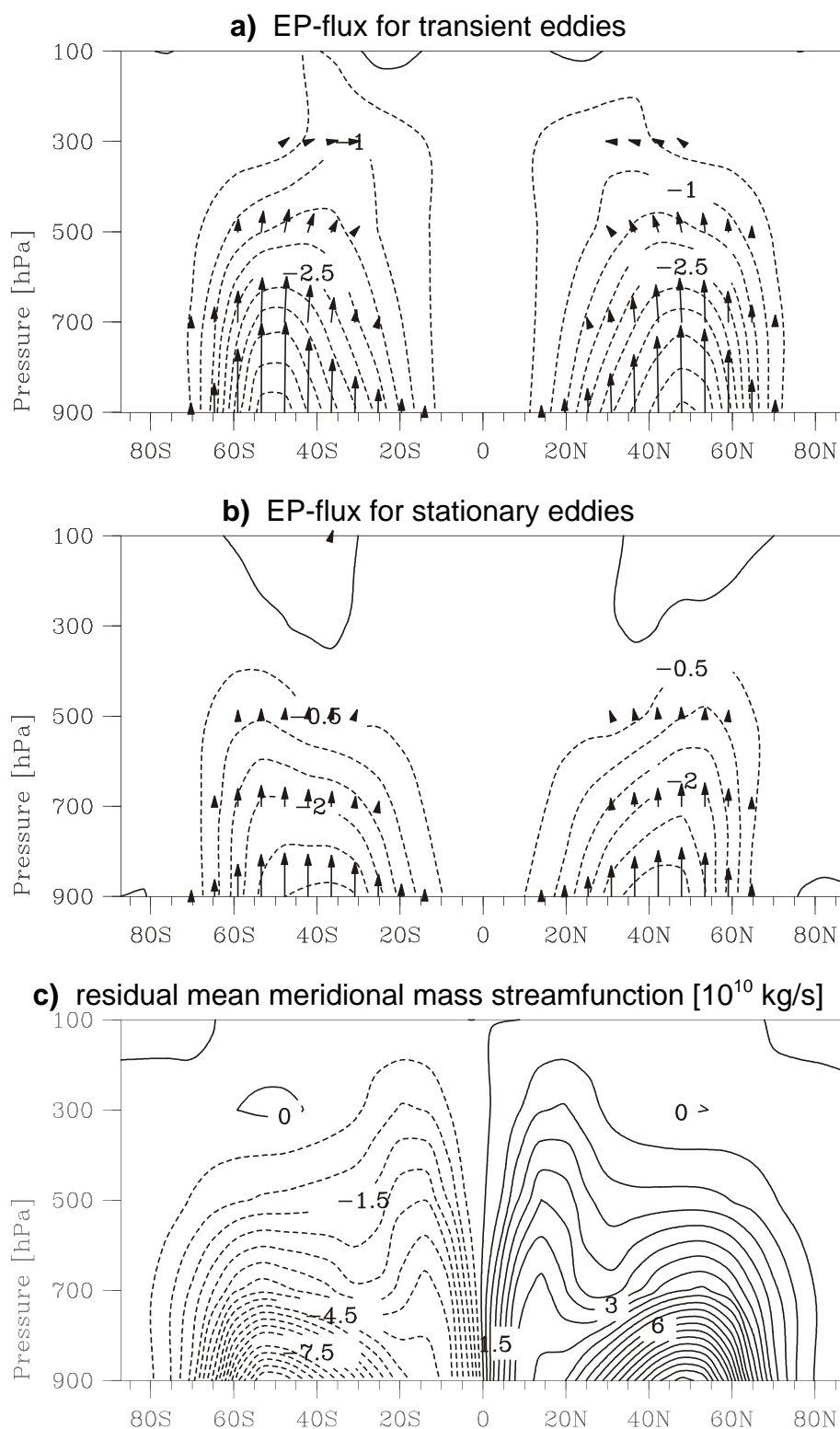
**f)**  $\overline{u'v'}$  300 hPa in [m<sup>2</sup>/s<sup>2</sup>]



**Figure C-6:** Northern hemisphere distributions of band-pass (left column) and low-pass (right column) filtered transient eddies.



**Figure C-7:** Simulated Lorenz energy cycle with energy reservoirs (boxes in  $[10^5 \text{ J/m}^2]$ ) and energy conversions (arrows in  $[\text{W/m}^2]$ ): ZPE and ZKE are the globally averaged mean available potential and kinetic energy, EPE and EKE define the eddy contributions. The subscripts 'stat' and 'trans' indicate the contribution by stationary and transient eddies. Ultralong eddies (1) and synoptic scale eddies (2) comprise the energy contributions by zonal wavenumber 1-3 and 4-10, respectively. The terms related to wavenumbers greater than 10 are negligibly small.



**Figure C-8:** Eliassen-Palm flux (arrows) and its divergence (contours in [ $10^{15}$  m<sup>3</sup>]) for (a) transient and (b) stationary eddies; (c) residual mean meridional mass streamfunction, which is obtained by a transformation utilizing the Eulerian mean circulation induced by the poleward temperature fluxes.