A SIMPLIFIED GLOBAL CIRCULATION MODEL

USERS' MANUAL

by

I.N. James, J.P. Dodd

Dept of Meteorology,
University of Reading,
READING RG6 2AU.

April 1993.

Any comments or corrections to this manual are welcomed, and will be considered for inclusion in future editions.

CONTENTS

1. Introduction

This is a development of the Reading Spectral Model, designed to be run as a Simplified Global Circulation Model or SGCM. The integration technique is traditional and is based on that described by Hoskins & Simmons (1975). The parent version of the code was described by Blackburn (1985). It was essentially conceived as a numerical weather prediction model, which reads initial data, balances it if required, and carries a short integration with fairly extensive diagnostic print outs. The model was extended, as described, for example, in James & Gray (1986) and James & James (1992) in order to carry out long integrations with a steady climatology with a variety of parameter settings. In these runs, the initial state was an extremely simple analytically defined flow, and the model was run for some hundreds of days until the flow had "spun up" to a state more or less typical of the climatological mean.

Blackburn (1985) remains a most useful reference to the model, and the reader is referred to that for fuller details of the computational procedures, the data organization and so on. This document repeats some of Blackburn (1985) but is generally less detailed and concentrates on the differences which characterize this particular version of the spectral model. The code as described exists as an UPDATE library on the CRAY YMP at the Rutherford/Appleton Laboratory. Apart from one or two library subroutines, principally for Fast Fourier Transforms and matrix manipulations, we have tried to stick to non-machine specific codes. So it ought not be too difficult to move the code to another machine. Be warned though: it is likely to be pretty slow on other than a supercomputer.

The main differences from the parent version are:

1 - The thermodynamic equation includes a "Newtonian cooling" term towards some "radiative equilibrium" temperature structure T_E:

$$\frac{\partial T}{\partial t} + \mathcal{L}_T + N_T = \frac{T_E - T}{\tau_E}.$$

2 - The vorticity and divergence equations contain "Rayleigh friction" terms:

$$\frac{\partial \zeta}{\partial t} + \mathcal{L}_z + N_z = -\frac{\xi}{\tau_F}$$

$$\frac{\partial D}{\partial t} + \mathcal{L}_D + N_D = -\frac{D}{\tau_F}.$$

3 - The initialization has been split into shorter, logically coherent subroutines.

4 - The code has been simplified by removing a number of redundant options. These include the provision for balancing of initial fields which are not needed for "GCM" type runs.

5 - The output has been simplified and redesigned. We note that most diagnostics will be calculated from history files in a subsequent run of a diagnostic programme. Provision is made for "quick look" diagnostics only appropriate to long runs to verify that the model is running smoothly, or for initial debugging. The format of output has been redesigned, recognizing that output is very rarely printed on 132column line printers nowadays, but is mainly viewed on an 80 column terminal display. Most format statements are not longer than 72 characters, enabling the results to be viewed on the CMS front end at the Rutherford Laboratory using XEDIT without splitting or truncating lines.

6 - The code has been commented, at least in those subroutines which the user is likely to want to modify frequently, and its layout redesigned to make its logical structure easier to follow. The standards suggested in Press et al (1989), Section 1.1, have been followed where practical.

7 - A policy decision has been made to reduce the number of options accessible from the code as it stands, particularly in setting up initial data and in controlling diagnostic output. In our view, the logical complexity which results from the provision of many options

makes it difficult to modify the code and encourages students to treat it as a black box. Because it is so difficult to predict just what will be wanted, we have taken the view that the code should be as straightforward as possible, providing basic facilities, but that it should be organized to make the addition of user defined facilities straightforward.

To this end, we have provided a number of simple utility subroutines within the update library, which will be of particular use in defining the initial and radiative equilibrium states. It is intended that these should be called from the users' code, and where necessary edited to provide the facilities needed by the individual. Simplicity and robustness is their keynote. We have been influenced by the approach of Press et al (1989) in this respect.

We expect to add to the number of these utility subroutines over a period of time.

2. Equation set

The model solves the primitive equations on a sphere. They are conveniently formulated in terms of the vertical component of absolute vorticity, ζ and the horizontal divergence D. Terrain following σ coordinates, where $\sigma = p/p_s$ are used in the vertical. This means that continuity is expressed as a prognostic equation for surface pressure p_s.

2a - Scaling

Variables are rendered dimensionless using the following characteristic scales:

Time Ω^{-1}, distance "a" (leading to a velocity scale $a\Omega$ and an energy scale $a^2\Omega^2$), temperature scale $(a^2\Omega^2)/R$ and pressure scale p_R which is taken to be 100kPa for the Earth.

Certain derived scalings are useful to know. Vorticity and divergence scale by Ω. Orographic height scales by $(a^2\Omega^2)/g$.

Coordinates: λ longitude and $\mu = \sin\phi$, ϕ being latitude. In the vertical, $\sigma = p/p_s$.

Temperature is divided into a reference part $T_R(\sigma)$ and an anomaly T_A. For reasons of computational stability, $T_R(\sigma)$ is usually set to be constant (250K for the Earth).

It is convenient to replace the zonal and meridional velocity components u and v by

$$U = u\cos\phi = u\sqrt{1-\mu^2}\ ,\ V = v\cos\phi = v\sqrt{1-\mu^2}\ .$$

2b - Vorticity equation:

$$\frac{\partial\zeta}{\partial t} = \frac{1}{1-\mu^2}\frac{\partial}{\partial\lambda}\mathcal{F}_v - \frac{\partial}{\partial\mu}\mathcal{F}_u - \frac{\zeta-\mu}{\tau_F} + K\,(-1)^{p/2}\,\nabla^p(\zeta-\mu)$$

where

$$\mathcal{F}_u = V\zeta - \dot{\sigma}\frac{\partial U}{\partial\sigma} - T_A\frac{\partial \ln p_s}{\partial\lambda},\quad \mathcal{F}_v = -U\zeta - \dot{\sigma}\frac{\partial V}{\partial\sigma} - T_A\,(1-\mu^2)\frac{\partial \ln p_s}{\partial\mu}$$

2c - Divergence equation

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

$$-\frac{D}{\tau_F} + K\,(-1)^{p/2}\,\nabla^p D.$$

2d - Thermodynamic equation

$$\frac{\partial T_A}{\partial t} = - \frac{1}{(1-\mu^2)} \frac{\partial}{\partial \lambda}(UT_A) - \frac{\partial}{\partial \mu}(VT_A) + D\,T_A - \dot{\sigma}\frac{\partial T}{\partial \sigma} + \kappa \frac{T\omega}{p} + \frac{T_E - T}{\tau_E}$$

$$+ K\,(-1)^{p/2}\,\nabla^p T_A.$$

2e - Continuity equation

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

2f - Hydrostatic equation

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

See Hoskins & Simmons (1975) for more details. The "hyperdiffusion" term $K\,(-1)^{p/2}\nabla^p Q$ (where p is even) included in the vorticity, divergence and temperature equations is there for computational reasons. It may be thought of as a parametrization of the cascade of energy onto subgrid scales and its subsequent dissipation.

3. Method of solution:

These equations are solved using the "spectral transform method" (Orszag 1970, Eliasen et al 1970). Any variable Q is represented by a truncated series of spherical harmonics:

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

where m is the zonal wavenumber and $n \geq m$ is called the "total wavenumber". $P_n^m(\mu)$ are the associated Legendre functions. The heart of the method relies on a transformation between Q and its 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 spectral space and nonlinear products (such as $U\zeta$) are evaluated in gridpoint space. The gridpoint phase of the step provides the opportunity to introduce local parametrizations of radiation, convective adjustments, friction and so on. In the SGCM, all such processes have been represented by linear terms, so they can be evaluated in spectral space. This may be a bit crude but it is very fast, and is perfectly adequate for some kinds of general investigations.

The Fast Fourier Transform (see, eg, Press et al, 1989, Section 12.2) provides an extremely fast transform in the zonal direction. The "half transforms" $Q^m(\mu)$, ie, the Fourier coefficients at each latitude, are calculated in this way. From these, the spectral coefficients are obtained by integration with respect to μ, using the orthogonality of the $P_n^m(\mu)$:

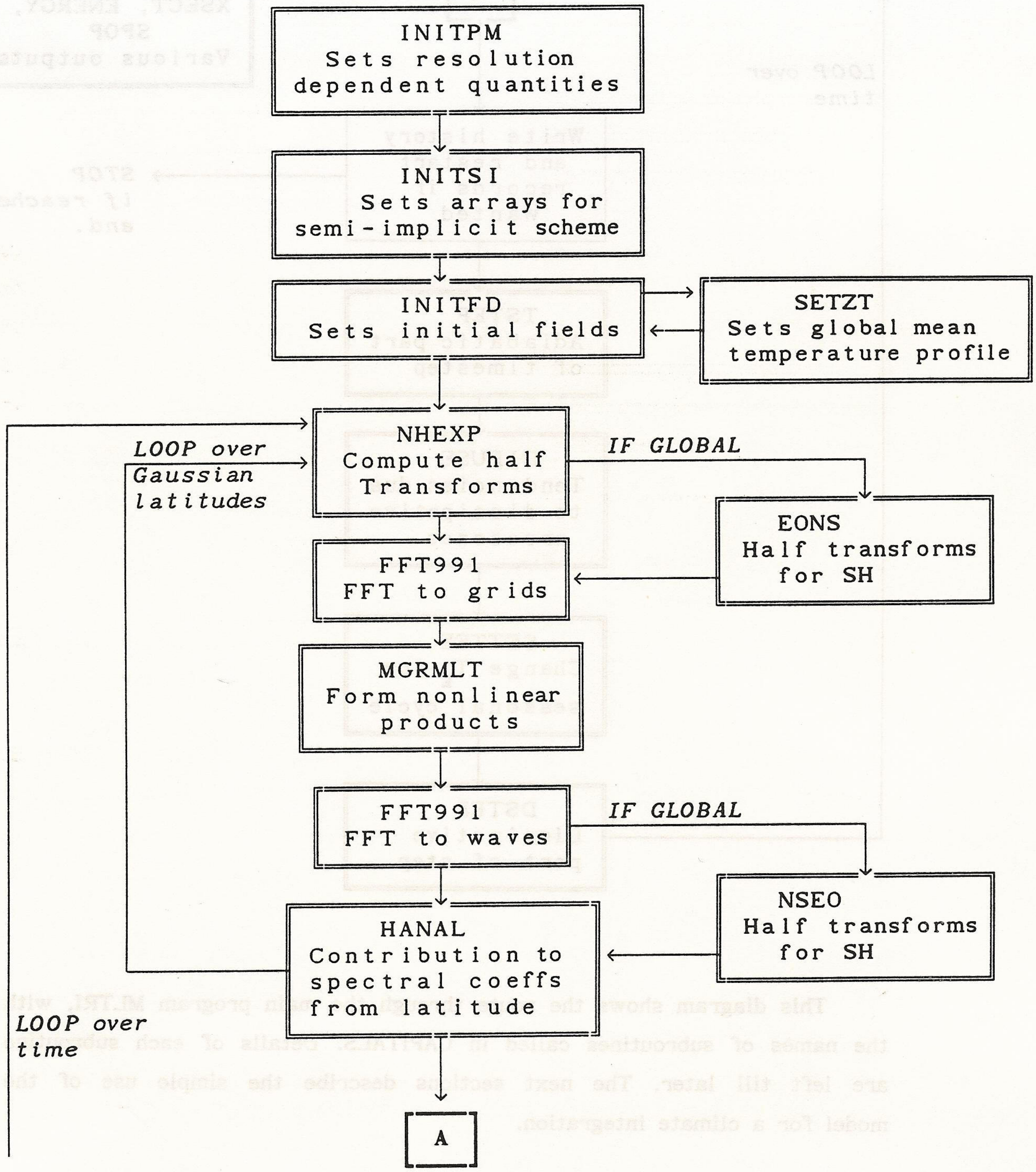
$$Q_n^m = \int_{-1}^{+1} Q^m(\mu)\ P_n^m(\mu)\ d\mu.$$

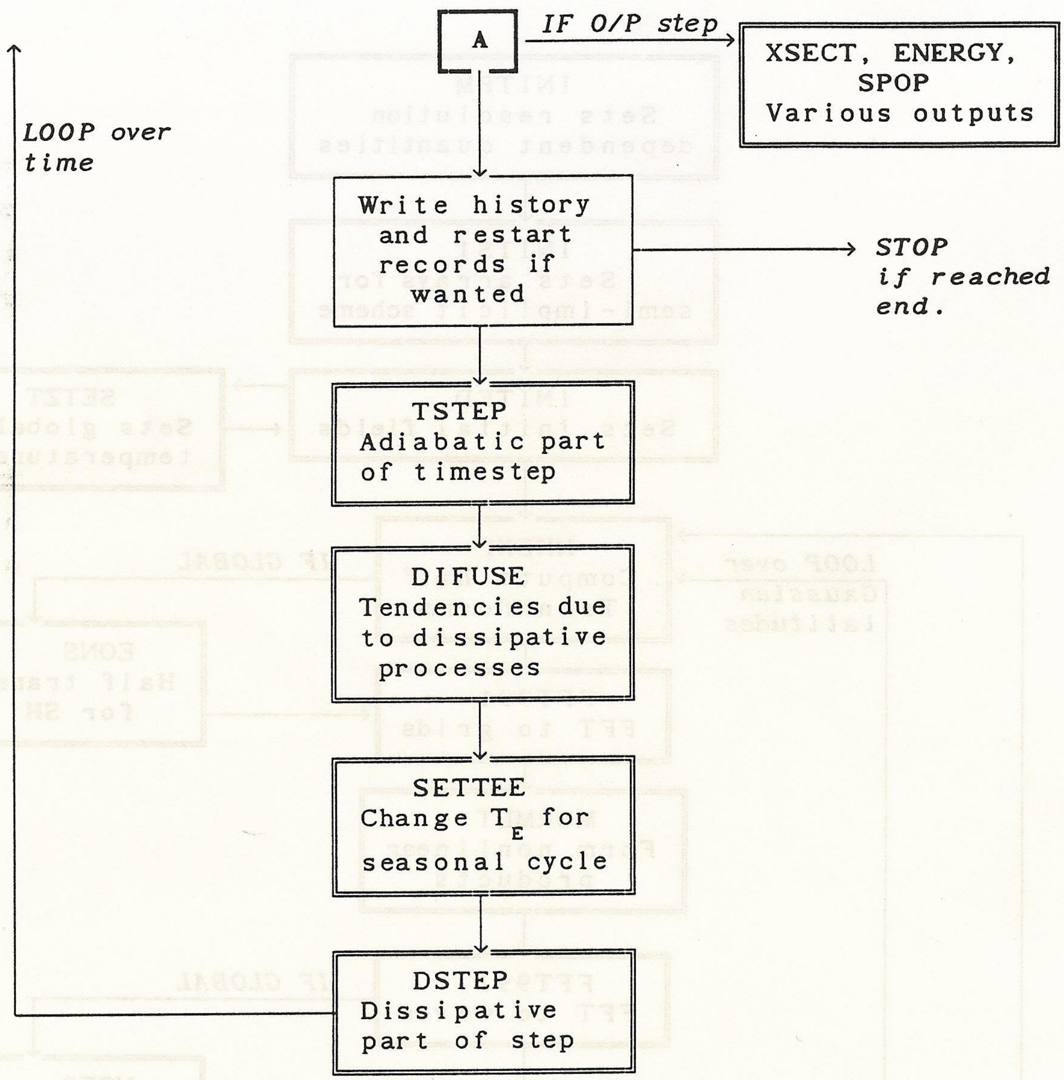
The transformation in μ is carried out by Gaussian quadrature (see Press et al 1989, Section 4.5).

The code has been written in FORTRAN 77. On the CRAY the line editor UPDATE is used to maintain the code. The basic code is kept in an UPDATE library to which members of the Department may have read access. Modifications are introduced by deleting or replacing specified lines. This is a bit clumsy, but has the huge advantage that we all work from a common code, which can be corrected or updated centrally should the need arise. Most importantly from your point of view, *all modifications you make are reversible.*

In the following description, we outline the broad structure of the code, and give enough details for you to run the model. Detailed description of common blocks and subroutines follow, and finally some sample job decks are included.

4. Model Structure





This diagram shows the route through the main program MLTRI, with the names of subroutines called in CAPITALS. Details of each subroutine are left till later. The next sections describe the simple use of the model for a climate integration.

The simplified set of switches used in this version of the code permit just two types of integration:

i - "Start" runs. The model starts from initial fields set in INITFD. These are usually zonally symmetric together with a small amplitude "white noise" perturbation needed to seed baroclinic waves. The defaults make the initial state a motionless, stably stratified atmosphere. It is set into motion as the temperature field relaxes towards T_E. Eventually it will become baroclinically unstable and the initial perturbations will begin to grow. With terrestrial parameters, the initial transient phase takes about 50-100 days; thereafter, the model wobbles about close to its climatology.

It is possible to begin the integration with a short forward timestep, followed by centred timesteps, each double the previous one, until the required timestep is acheived. This is used if you want to start smoothly from an unsteady initial state defined by data or a previous history record.

ii - "Restart" runs. These are intended to extend a previous integration. A full set of spectral coefficients is picked up from a "restart" file and the model carries on integrating. No perturbation is added. The only point to watch is that the various parameter settings and T_E have been set up correctly and consistently with the previous run.

5. Parameter statements

5a - PARAM1

The dimensions of all the arrays are set by parameter statements in the common decks PARAM1 and PARAM2. You should only need to modify PARAM1. The values in PARAM2 are derived from PARAM1 or are invariants that need no modification. Variables set in PARAM1 are as follows:

MM Highest zonal wavenumber retained in the spectral series.
NN Highest total wavenumber retained in the spectral series.
(for the ordinary triangular truncation, MM = NN)
NL Number of levels
NHEM Number of hemispheres: 1 means a "hemispheric" run, ie, flow

	symmetric about the equator, 2 means ordinary global run.
MOCT	Symmetry in longitude, so that only zonal wavenumbers 0, MOCT, 2*MOCT, etc, are retained. For ordinary global runs, MOCT=1.
MG	Number of longitudes in transform grid. Must be of the form 2^p3^q where p≥1, and must satisfy MG≥(3*MM+1)/MOCT to prevent aliasing of quadratic terms.
JG	Number of Gaussian latitudes in transform grid between pole and equator. Must satisfy JG≥(3*MM+1)/4 to prevent aliasing in quadratic terms.
NWJ2	Number of either odd or even spectral coefficients (equal for jagged triangular truncation). It may be computed from the formula (assuming FORTRAN integer division): NWJ2=(MM+1)/2 + (MM+1-MOCT)/2 + (MM+1-2*MOCT)/2....etc
NCRAY	Number of spectral transforms to be performed in parallel. Optimum for CRAY is 64.
JGL	Number of latitudes at which Legendre functions and their derivatives are stored in core. This may be JG for lower resolution runs but has to be 1 for high resolution runs (ie, T63 and higher).

5b - PARAM2

A large number of derived constants are set in this parameter statement, including most of the array dimensions. A list is given here:

MH	2 (for counting odd/even spectral coefficients).
PI	3.141592......etc.
PI2	2.*PI
NNP	NN+1
MGPP	MG+2, no of gridpoints stored per latitude. Only MG independent numbers, though!
JGP	JG+1
MJP	NWJ2*2 - used for equivalencing spectral arrays to REAL arrays.
NLM	NL-1
NLP	NL+1

NLPP	NL+2
NLA	NL+3
NLB	NL+4
NL2	NL*NL - array dimension for semi-implicit scheme.
IDA	(3*MG)/2+1 - no of trig functions needed by FFT991.
IDB	NWJ2*NL
IDC	2*IDB
IDD	MGPP*NL - dimension of gridpoint arrays for Gaussian latitude.
IDE	NL2*NN - dimension of BM1
IDF	NCRAY*(MG+1) - size of work area for FFT991.
IDG	JG*NL
IDM	JGP*(2*MG+1)
IDI	NNP/2
IDJ	IDI*IDI - redundant, used for balancing.
IDK	NL*IDI - redundant, used for balancing.
IDL	MGPP/2 - dimension of complex half transform array for single level.
IDM	NNP/2 - No of coefficients for m=0 at a single level.
IDN	NL*IDM
NWW	1 + (MM-1)/MOCT - no of m wavenumbers or half transforms.
IGA	NHEM*NWJ2 - No of spectral coefficients at single level.
IGB	NL*IGA
IGC	MGPP*NHEM - Dimension of gridpoint array at single level.
IGD	MGPP*NHEM*NL
IGG	IDG*NHEM
IGL	IDL*NHEM
IGM	IDM*NHEM
IGN	IDN*NHEM
IGO	2*IGA
IGP	2*IGB
NVRI	7*NL+2 - No of single level fields to be transformed in NHEXP or HANAL.
IDPOL	NVRI*IGA
JGT	NHEM*JG

6. Namelists

Once the resolution has been fixed by setting PARAM1 variables, other characteristics of the desired run are set by means of data in NAMELISTs which are read in by subroutine INITPM. Usable defaults have been set for these variables, but you will almost certainly need to change some of these. Take care to check that you have put the correct variables in your namelist. CRAY Fortran gives especially enigmatic and unhelpful messages when an error in a namelist is encountered.

(Nb, SI units are assumed unless overwise stated. The usual convention for variable types, ie, first letter I to N means INTEGER, else REAL, is assumed).

6a - Namelist INPPL: planetary parameters.

Variable	Default	Meaning
GA	9.81	Gravitational acceleration g.
WW	7.292E-5	Rotation rate of planet Ω.
RADEA	6.371E6	Radius of planet a.
GASCON	287	Gas constant R.
AKAP	0.286	R/c_p.

6b - Namelist INPRN: run specific parameters.

Variable	Default	Meaning
PNU	0.02	Robert time filter parameter.
TDISS	0.25	Dissipation time in sidereal days for wavenumber NN by artificial hyperdiffusion.
NDEL	6	Order of hyperdiffusion (p in Section 2).
LSTRETCH	.F.	Switches between equispaced σ levels when .F. and a stretch defined by a formula in INITPM when .T..
TFRC	(NL-1)*0.,	1. : Array length NL containing the friction timescale τ_F in days for each level.
RESTIM	NL*15.	Radiative equilibrium timescale τ_E for each level.
TO	NL*250.	Reference temperature profile $T_R(\sigma)$.
DTEP	60.	Equator-pole temperature difference at surface.

DTNS	0.	Pole-pole temperature difference at surface; controls seasonality.
ALR	6.5E-3	Temperature lapse rate in troposphere for T_E.
TGR	288.	Global mean temperature of ground used to set T_E.
ZTROP	12.E3	Height of tropopause in T_E.
DTTRP	2.	A temperature increment which controls the sharpness of the tropopause in T_E.
YRLEN	0.	Length of year in days. 0. means no seasonal cycle.
LRSTRT	.F.	LOGICAL variable; .TRUE. if restart run.
BEGDAY	0	First day of a restart run, irrelevant to a start run.
KRUN	0	Total number of timesteps to be carried out in this run.
KITS	3	Number of short initial timesteps, irrelevant in a restart run.
TSPD	24.	Number of timesteps per day.

6c - Namelist INPOP: variables relating to output type and frequency.

Variable	Default	Meaning
RNTAPE	0.	A unique run name, used to identify history records.
KOUNTH	0	Number of timesteps between history dumps; 0 means no history.
KOUNTR	0	Number of timesteps between restart dumps; 0 means no restarts.
KOUNTP	0	Number of timesteps between output of cross sections and spectral coefficients; 0 means no such output.
KOUNTE	0	Number of timesteps between global mean diagnostics; 0 means no such diagnostics.
NCOEFF	0	Total wavenumber to which spectral coefficients are to be output. 0 means no spectral coefficients.
LSPO	NL*.F.	LOGICAL array. Selects levels for spectral coefficients. If all are .FALSE., only surface pressure coefficients written.

7. Associated files

A number of files are used by the model. Some hold data, some are used to record the progress of an integration for future analysis and some are used as temporary backing while the run proceeds.

File	Purpose
fort.2	Lineprinter output from the model. Monitors initial setup and the progress of the run, according to the variables set in namelist INPOP.
fort.7	Used to read the namelists controlling the run.
fort.9	History file built up during integrations, containing arrays of spectral coefficients at predetermined times. These files are used by BGFLUX and its derivatives to generate diagnostics, and form the primary output of a run. History records have the form: RKOUNT,RNTAPE,DAY,Z,D,T,SP,RNTAPE Writing RNTAPE twice affords diagnostic programs a check that the correct resolution has been selected.
fort.10	Restart file used to initiate a restart run. Restart records are similar to history records, except they contain spectral data for the previous timestep as well as the present step, enabling an integration to be restarted exactly. Restart data is read from fort.10 when the run is started. Restart records have the following form: RKOUNT,RNTAPE,DAY,Z,D,T,SP,TRES,ZMI,DMI,TMI,SPMI,RNTAPE
fort.11	This file is used for restart records written during a restart run. The structure of its records are the same as for fort.10. You may wish to merge fort.10 and fort.11 at the end of the run to build up a complete file of restart records for a long

run.

fort.12 Used at the end of the run to take a copy of the final restart record. The main use of this is to simplify running a very long run; the latest restart can be used without having to change the job deck.

fort.24 Temporary file used to hold gridpoint data for every Gaussian latitude, starting at the pole, for output purposes.

fort.25 Temporary file used to hold the Legendre functions and their derivatives with respect to μ at each Gaussian latitude for high resolution runs with JGL = 1.

8. Common blocks

The common blocks are described in alphabetical order. Note that in general arrays of vertical profiles run from *top* to *surface*, and arrays over Gaussian latitudes run from the *pole* to the *equator*.

```
COMMON   SQ(NNP),RSQ(NNP),RCS(JG),SIGMAH(NLM),SIGMA(NL)
         ,T01S2(NLM),T0(NL),ALPHA(NL), DSIGMA(NL),RDSIG(NL)
         ,TKP(NL),C(NL2),SQH(NNP),LSTRETCH
         ,MF,MFP,JZF,NF,NFP
         ,AKAP,RCSJ,GA,GASCON,RADEA,WW,PFAC
         ,EZ,AIOCT,LRSTRT
COMPLEX EZ,AIOCT
LOGICAL LRSTRT,LSTRETCH
```

A heterogeneous collection of constants and switches.

Variable	Description
SQ	n(n+1) - needed for hyperdiffusion.
RSQ	$[n(n+1)]^{-1}$
RCS	$1/(1 - \mu^2)$ for each Gaussian latitude, starting at the pole.
SIGMAH	σ at half levels.
SIGMA	σ at full levels.
T01S2	$(T_R(\sigma_{l+1}) - T_R(\sigma_l))$ for l = 1 to NL-1.
T0	$T_R(\sigma)$
ALPHA	$\ln(\sigma_{l+1/2}/\sigma_{l-1/2})$, needed for vertical differencing.
DSIGMA	$\Delta\sigma$ between adjacent half levels.
RDSIG	$1/(2\Delta\sigma)$
TKP	κT_R at each level
C	Matrix set in INITSI for semi-implicit scheme.
SQH	n(n+1)/2
LSTRETCH	Switches between equispaced σ levels when .F. and a stretch defined by a formula in INITPM when .T.. The actual stretch coded was used by Simmons & Hoskins (1978); the user may wish to implement alternative

	stretches.
MF	MM-1
MFP	MM
JZF	MGPP-2*NWW. Number of blank real values between adjacent half transforms.
NF	NN-1
NFP	NN
AKAP	$\kappa = R/c_p$.
RCSJ	$1/(1-\mu^2)$ at current Gaussian latitude in loop in MLTRI.
GA	g
RADEA	a
WW	Ω
PFAC	$a^2\Omega^2 p_R/(2g)$ - scaling for global energy integrals.
EZ	$((8/3)^{1/2},0)$ - Spectral coefficient of the planetary vorticity scaled by Ω.
AIOCT	$i\times$(MOCT)
LRSTRT	Switch for START/RESTART runs.

==

```
COMMON/BATS/BM1(IDE),ALAT(JG),AK(NNP)
            ,AQ(NL2),G(NL2),TAU(NL2),BEGDAY,KRUN
            ,KOUNT,KITS,KTOTAL,ITSPD,PNU21
            ,SRGTO,DELT,DETL2,CV,CG,CT,PNU,PNU2,DAMP(NL)
```

Constants and counters to do with the time stepping scheme and the semi-implicit scheme, together with constants used to de-dimensionalize variables. Vectors and matrices dealing with the semi-implicit scheme are marked SI; see Hoskins & Simmons (1975) for details.

Variable	Description
BM1	Matrix A^{-1} for n=1,NN (SI).
ALAT	Contains Gaussian latitudes, in *degrees*, starting at the pole. For output purposes.
AK	$K\times[n(n+1)]^{p/2}$, non dimensional hyper-diffusion coefficient for wave n.
AQ	Matrix used in SI scheme (SI).

G	Matrix which defines vertical scheme (SI).
TAU	Matrix for temperature tendency (SI).
BEGDAY	Initial day of a RESTART run, used to find the correct restart record. Ignored for START runs.
KRUN	Total number of full timesteps to be taken during the run.
KOUNT	Number of timesteps completed.
KITS	Number of short initial timesteps.
KTOTAL	Number of timesteps in integration. For a START run, this is simply KRUN+KITS-1. For a RESTART run, it is set to be KOUNT+KRUN where KOUNT is taken from the initial restart record.
SRGTO	Needed for balancing, redundant.
DELT	Timestep Δt (dimensionless).
DELT2	$2\Delta t$
CV	$a\Omega$, velocity scale.
CG	$a^2\Omega^2$, energy scale.
CT	CG/R, temperature scale.
PNU	Constant for Robert time filter.
PNU2	2×PNU
DAMP	$1/\tau_E$ for each level, dimensionless.
ITSPD	Integer number of timesteps per day.
PNU21	(1 - 2×PNU).

==

```
COMMON/CFFT/NTGW,NRSTGW,NTWG,NRSTWG
           ,TRIG(IDA),WORK(IDF),IFAX(10)
```

All variables have type REAL. Contains data required for the Fast Fourier Transforms, particularly with regard to vectorization on the CRAY. The vector processor works optimally when 64 simultaneous transforms (one transform being a Fourier transform of one variable around one latitude circle) are carried out. The code is arranged to carry out the transforms for each latitude circle in batches of 64, with any remainder performed with a final call to FFT991.

Variable	Description
NTWG	No of complete batches of 64 wave to grid transforms, ie, ((5*NL+1)*NHEM-1)/NCRAY.
NRSTWG	No of transforms left for final batch.
NTGW	As NTWG, but for grid to wave transforms, ie, ((7*NL+2)*NHEM-1)/NCRAY.
NRSTGW	As NRSTWG, but for grid to wave transforms.
TRIG	Trignometric functions for FFT991.
WORK	Workspace for FFT991.
IFAX	Factors of MG for FFT991.

==

```
COMMON/GRIDP/PJT(IGC),PLT(IGC),UGT(IGD),VGT(IGD)
              ,UG(IGD),VG(IGD),ZG(IGD),DG(IGD),TG(IGD)
              ,PMG(IGC),PJG(IGC),PLG(IGC),VPG(IGC),EG(IGD)
              ,UTG(IGD),VTG(IGD),UZG(IGD),VZG(IGD),TNLG(IGD)
              ,VZGT(IGD),UZGT(IGD)
```

All variables have type REAL.

IGC - number of gridpoints at a single level for a given Gaussian latitude, ie, MGPP*NHEM.

IGD - number of gridpoints at NL levels, ie, NL*IGC.

These arrays are best thought of as holding gridpoint values of quantities. You may want to access them for diagnostic purposes. Note that at some parts of the step, they are over written with "half transforms", ie, a Fourier transform of the gridpoint data. Only data for a single pair of latitude circles (NH and SH) is available at any one time. If gridpoint data at all longitudes is wanted for diagnostic purposes, it is written to fort.24 during the loop over Gaussian latitudes. In principle, these gridpoint data may be used to compute tendencies due to locally parametrized processes. However, a more elaborate "split" integration scheme, which avoids these tendencies passing through the semi-implicit scheme is required. Dissipative-type processes need a forward timestep or some kind of implicit timestepping (eg, Du Fort-Frankel) to ensure numerical stability. The "moist" code BGCM5, due to M. Blackburn, incorporates these points.

On the assumption that most users will not need to mess with the half transform data, you are referred to Blackburn (1985) for details of the way in which the half transform data is stored. The gridpoint data is stored by level, starting at the top and working down. Within each level, there are MGPP values for the northern hemisphere ($\mu > 0$), followed by MGPP values for the southern hemisphere ($\mu < 0$). All are stored in order of increasing longitude. Note that the values at MGPP-1 and MGPP are the same as the values at longitude 1 and 2 respectively.

For hemispheric runs (NHEM = 1), data for μ>0 only is stored.

Variable	Description

The following contain gridpoint data, created by a transform from spectral space.

PJT	$(1 - \mu^2)\ \partial(\ln p_s)/\partial\mu$
PLT	$\ln p_s$
UGT	U_χ (zonal component of divergent wind)
VGT	V_χ (meridional component of divergent wind)
UG	U - first rotational, then total.
VG	V - first rotational, then total.
ZG	ζ
DG	D
TG	T_A
PMG	$\partial(\ln p_s)/\partial\lambda$
PJG	$(1 - \mu^2)\ \partial(\ln p_s)/\partial\mu$
PLG	$\ln p_s$ or p_s (changed to p_s at beginning of MGRMLT)
VPG	Grid point version of VP (see SPECTR).

The following hold nonlinear products to be transformed to spectral *space.*

EG	$(U^2 + V^2)$
UTG	UT_A
VTG	VT_A
UZG	$\mathcal{F}_v$
VZG	$\mathcal{F}_u$
TNLG	Nonlinear temperature tendencies.
VZGT	$\partial(\mathcal{F}_u)/\partial\lambda$.
UZGT	$\partial(\mathcal{F}_v)/\partial\lambda$.

==

```
COMMON/LEGAU/ALP(MJP,JGL),DALP(MJP,JGL),DP(MJP,JGL),DQ(MJP,JGL)
            ,AW(JG),CS(JG),SI(JG)
            ,WEIGHT,CSJ,MJPP,SIT,JL,
            ,ALPW(MJP),DALP(MJP),CALPW(MJP),SQALP(MJP)
```

All these variables have type REAL.

Contains various constants for the Legendre transforms as well as values of the Legendre functions on Gaussian latitudes. Arrays of Legendre functions and associated fields stored in ascending order of n within an ascending order of m, even and odd functions alternating, for each Gaussian latitude starting at the pole.

Variable	Description
ALP	$P_n^m(\mu)$ - either for current latitude (JGL=1) or all latitudes. (JGL = JG)
DALP	$(1 - \mu^2)dP_n^m/d\mu$
DP	$mP_n^m(\mu)/[n(n+1)]$
DQ	$(1 - \mu^2)dP_n^m/d\mu/[n(n+1)]$
AW	Gaussian weight $w(\mu)$ for all Gaussian latitudes.
CS	$(1 - \mu^2) = \cos^2(\phi)$ for all Gaussian latitudes.
SI	$\mu = \sin(\phi)$ for all Gaussian latitudes.
WEIGHT	AW for current latitude.
CSJ	CS for current latitude.
MJPP	MJP + NWW.
SIT	SI for current latitude.
JL	Subscript for ALAT, AW, CS, SI etc for current latitude. (No! only for ALP, DALP, DP, DQ.)
ALPW	$w(\mu)P_n^m(\mu)$
DALPW	$w(\mu)(1 - \mu^2)dP_n^m/d\mu$.
CALPW	$w(\mu)P_n^m(\mu)/(1 - \mu^2)$.
SQALP	$\frac{1}{2}n(n+1)w(\mu)P_n^m(\mu)$

==

```
COMMON/OUTCON/RNTAPE,NCOEFF,LSPO(NL),INSPC,
        ,KOUNTE,KOUNTP,KOUNTH,KOUNTR,KOUTE,KOUTP,KOUTH,KOUTR,DAY,
        ,SQR2,RSQR2,EAM1,EAM2,TOUT1,TOUT2,RMG
LOGICAL LSPO
```

Contains constants, switches and counters for printed output used to monitor the run.

RNTAPE	User defined run identifier.
NCOEFF	Controls maximum total wavenumber for printed spectral coefficients. If 0, no coefficients printed.
LSPO	If .TRUE., coefficients of Z, D and T are printed for that level. If all elements .FALSE., just SP printed.
INSPC	Defined in WRSPCA, number of coefficients per field per level to be printed.

(In the following counters, selecting 0 switches off the output).

KOUNTE	Number of timesteps between printout of global integral quantities.
KOUNTP	Number of timesteps between printed cross sections from XSECT and spectral coefficients.
KOUNTH	Number of timesteps between dumping history records.
KOUNTR	Number of timesteps between dumping restart records.

(The following counters are tested to see whether output is due).

KOUTE	Test for global integrals.
KOUTP	Test for printed output.
KOUTH	Test for history dump.
KOUTR	Test for restart dump.

(The following are simply useful numbers).

DAY	Day of integration (ie, FLOAT(KOUNT-KITS+1)/TSPD)

SQR2	$2^{1/2}$.
RSQR	$2^{-1/2}$.
EAM1	$2^{1/2}/3$
EAM2	$(2/45)^{1/2}$.
TOUT1	$\sum T_R \Delta\sigma$
TOUT2	$\sum T_R^2 \Delta\sigma$
RMG	1./FLOAT(MG)

==

```
COMMON/POLYNO/POLY(IDPOL),CMPA(IGL)
REAL POLY
COMPLEX CMPA
```

These arrays are set in NHEXP and HANAL and aid the vectorization of the Legendre transforms.

==

```
COMMON/RESTOR/TRES(IGB),DTNS,DTEP,DTTRP,FAC(NL),TFRC(NL),
              ,YRLEN,TRS(NL),ALR,ZTROP,TGR
COMPLEX TRES
```

Contains data needed for the Rayleigh friction and Newtonian cooling terms.

Variable	Description
TRES	Spectral coefficients of the radiative equilibrium temperature field.
YRLEN	Length of the year in days (used by SETTEE).
TRS	Global mean of T_E at each σ level. Set by SETZT.
FAC	Determines vertical variation of horizontal temperature gradients. Currently set to be large near surface and zero in stratosphere, falling off in the upper

	troposphere.
TFRC	Contains the Rayleigh friction timescales for each level. After INITPM, it contains the dimensionless damping rates.

(Remaining variables are determined by NAMELIST INPRN).

DTEP	Equator-pole temperature difference in T_E.
DTNS	Pole-pole temperature difference in T_E.
ALR	Lapse rate in T_E near surface.
ZTROP	Height of tropopause.
TGR	Global mean ground temperature in T_E.
DTTRP	Temperature increment which controls sharpness of tropopause (see description of SETZT).

==

```
COMMON/SPECTR/SP(IGA),SPDU(IGA),DDA(IGB),ZDA(IGB),ZDB(IGB),DDB(IGB)
             ,Z(IGB),D(IGB),T(IGB),SPA(IGA),VP(IGA)
             ,TT(IGB),DT(IGB),ZT(IGB),GS(IGA)
             ,ZMI(IGB),DMI(IGB),TMI(IGB),SPMI(IGA)
```

All arrays COMPLEX. IGA = NHEM*NWJ2 is the number of coefficients at a single level, IGB = IGA*NL is the number of coefficients at NL levels. Coefficients are stored by level, beginning at the top and working downwards. Within the level, first find the NWJ2 odd coefficients (for Z, ZT, ZMI, ZDA, ZDB) or NWJ2 even coefficients (for SP, D, T, VP, GS, SPDU, DDA, DDB, T, SPA, TT, DT, DMI, TMI & SPMI), followed by the NWJ2 even (odd) coefficients. For a hemispheric integration, only the first NWJ2 coefficients are retained. Within each block of coefficients, storage is by increasing n within increasing m, ie,

for a variable with "odd" symmetry (eg, vorticity):

$$Z_1^0,\ Z_3^0,\ Z_5^0,\ldots\ldots,Z_2^1,\ Z_4^1,\ Z_6^1,\ldots\ldots\ldots,Z_{NN}^{NN-1}.$$

For a variable with "even" symmetry (eg, divergence, temperature or surface pressure):

$$D_0^0,\ D_2^0,\ D_4^0,\ldots\ldots,D_1^1,\ D_3^1,\ D_5^1,\ldots\ldots\ldots,D_{NN-1}^{NN-1}$$

Variable	Description
SP	ln p_s coefficients

The next five fields are essentially copies of spectral arrays which improve vectorization.

SPDU	Copy of SP
DDA	$(-i)$*D
ZDA	$(-i)$*Z
ZDB	Copy of Z
DDB	-D

More basic variables.

Z	ζ coefficients - *Absolute* vorticity!
D	D coefficients
T	T_A coefficients (ie, $T - T_R(\sigma)$).
SPA	p_s coefficients
VP	Contribution to ln p_s tendency: $\sum_{l=1}^{NL} p_l \Delta\sigma_l$

The next three arrays hold tendencies

ZT	$\partial Z_n^m/\partial t$
DT	$\partial D_n^m/\partial t$
TT	$\partial T_n^m/\partial t$ *(remember this is T_A not T!)*

Orography

GS	Coefficients of surface geopotential, ie, $h(\phi,\lambda)g$, scaled by $a^2\Omega^2$.

The next arrays hold coefficients for the previous timestep.

ZMI ζ coefficients at previous timestep.

DMI D coefficients at previous timestep.

TMI T_A coefficients at previous timestep.

SPMI $\ln p_s$ coefficients at previous timestep.

9. Brief description of subroutines

For ease of reference, these are listed in alphabetical order. See section 4 for their sequence within the model run. In some cases, fuller details are given in Blackburn (1984); the code itself is the primary source of information!

DIFUSE - Calculates the spectral tendencies due to hyper-diffusion, Newtonian cooling and Rayleigh friction. The interim fields generated by TSTEP are used.

Called from MLTRI.

DSTEP - Performs the "diabatic" part of the timestep, ie, includes tendencies calculated by DIFUSE and completes the Robert filter.

Called from MLTRI.

ENERGY - Calculates a number of global integral quantities from spectral coefficients, namely, rms relative vorticity, rms divergence, rms temperature, PE +IE, mean p_s. All are nondimensional except PE+IE.

Called from MLTRI.

EONS - Sums and differences the even and odd contributions to the half transforms to form the complete half transforms for each latitude.

eg,

$$\zeta^m(\mu) = \zeta^m_{odd}(\mu) + \zeta^m_{even}(\mu)$$

$$\zeta^m(-\mu) = -\zeta^m_{odd}(\mu) + \zeta^m_{even}(\mu)$$

Not called when NHEM=1. Called by MLTRI.

FFT991 - Fast Fourier Transform subroutine, for taking similar forward or reverse transforms simultaneously. 64 is the optimum number for the CRAY YMP. The number of data points MG must be of the form $2^p \times 3^q$, $p \geq 1$.

Called by MLTRI.

HANAL - Evaluates the contributions of the nonlinear tendencies calculated by MGRMLT and NSEO using Gaussian quadrature, and adds them to the spectral tendencies SPA, VP, TT, DT and ZT.

Called from MLTRI.

INITPM - Initializes the various parameters for the run. Reads data from the namelists and sets up the various arrays for the run. The default is to select equispaced σ-levels for NL<15, otherwise the stretch developed by Simmons and Hoskins (1978) for their lifecycle experiments is used. Details of the set up selected are written to fort.2.

Calls WRSPS to initialize the printing of spectral coefficients.
Called by MLTRI.

INITSI - Initializes the various arrays for the semi-implicit scheme and vertical scheme for the model.

Called by MLTRI.
Calls QREIG, MINV.

INITFD - Initializes the spectral arrays for the run. For a "restart" run, fields of initial spectral coefficients are read from fort.10 and the run proceeds from these. For a "start" run, fields are set to a motionless, stably stratified atmosphere with a "white noise" perturbation.

Finally sets arrays needed by FFT991.

Called by MLTRI.

Calls SETZT, FAX, FFTRIG.

LGNDRE - Calculates the Legendre functions $P_n^m(\mu)$ and derivatives $(1-\mu^2)dP_n^m/d\mu$ for each Gaussian latitude.

Called from INITPM.

MGRMLT - Forms products and other nonlinear functions of gridpoint data in order to generate nonlinear tendencies. Takes the arrays UG, VG, ZG, DG, TG, PMG, PJG and PLG and uses them to set the arrays PLG, VPG, EG, UTG, VTG, UZG, VZG, TNLG.

Called from MLTRI.

MLTRI - Main program. Organizes calls to subroutines and handles i/o to history and restart files. Its organization is illustrated in Section 4.

Calls DIFUSE, DSTEP, EONS, FFT991, HANAL, INITFD, INITPM, INITSI, MGRMLT, NHEXP, NSEO, SETTEE, TSTEP.

NHEXP - Performs the Legendre transform from spectral space to form the even and and odd contributions to the half transforms at the current Northern Hemisphere latitude (μ>0). In global mode, its call must be followed by a call to EONS. In hemispheric mode, the half transforms are ready for the Fast Fourier Transform.

Called by MLTRI.

NOISE - Adds a white noise perturbation to the spectral coefficients of ln p_s for which m≥1. The amplitude of the perturbation is normalized to 10Pa.

Called from SETZT.

NSEO - The inverse to EONS. Sums and differences the complete half transforms to produce even and odd contributions to the half transforms in the northern hemisphere, eg,

$$\zeta^{m}_{odd}(\mu) = \frac{1}{2}\left(\zeta^{m}(\mu) - \zeta^{m}(-\mu)\right)$$

$$\zeta^{m}_{even}(\mu) = \frac{1}{2}\left(\zeta^{m}(\mu) + \zeta^{m}(-\mu)\right)$$

Operates on arrays PLG, VPG, EG, UTG, VTG, UZG, VZG, TNLG. Called from MLTRI.

SETTEE - Modifies the radiative equilibrium temperature T_E as the run proceeds to introduce a seasonal cycle if wanted. The formula for the equilibrium temperature (see SETZT) is modified to read:

$$T_E(\phi,\sigma) = T_{ER}(\sigma) + f(\sigma)\left(\Delta T_{NS} P^0_1(\mu)\cos(2\pi t/\tau_y) + \Delta T_{EP} P^0_2(\mu)\right)$$

where τ_y is the length of the year, set with the variable YRLEN. If YRLEN = 0., the seasonal cycle is switched off.

Called from MLTRI.

SETZT - Sets up a radiative equilibrium temperature field for the run. This breaks into two steps:

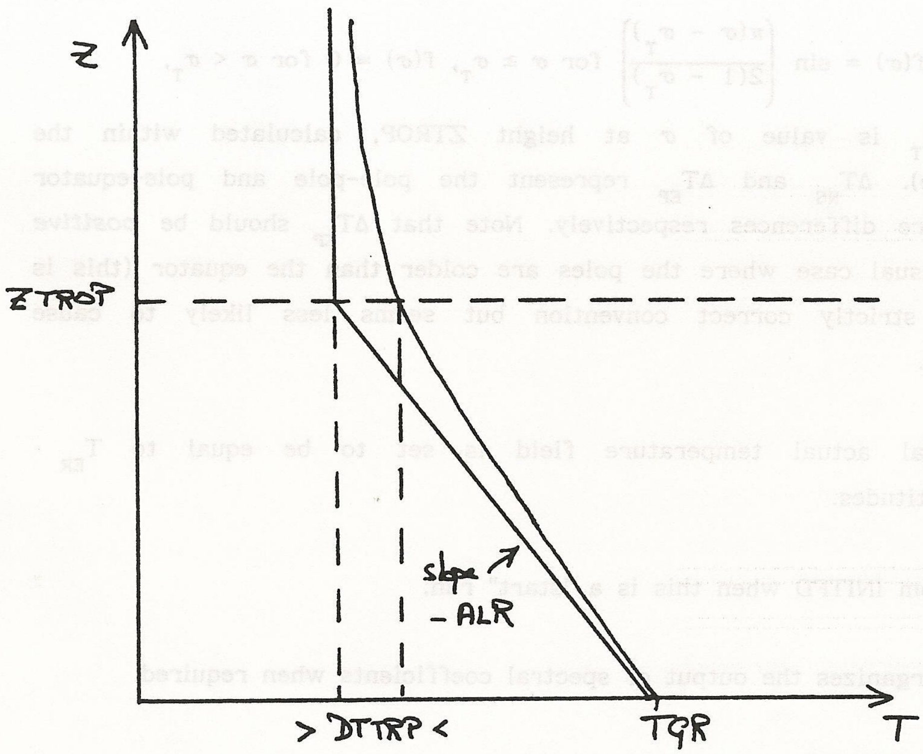


Fig (9.1) - Specifying T_{ER}.

i - First define a global mean radiative equilibrium temperature profile $T_{ER}(\sigma)$. A hyperbolic function of height is used to define T_{ER}, illustrated in Fig (9.1). As $z \rightarrow -\infty$, the profile tends to a uniform lapse rate, ALR, passing though the temperature TGR at $z = 0$. As $z \rightarrow +\infty$, the profile becomes isothermal. The transition takes place at a "tropopause" at a 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 is spread. The hydrostatic relation is used to determine the heights and hence the temperatures of the model levels.

ii - The radiative equilibrium temperature is then set to be:

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

The function $f(\sigma)$ is set to become small near the upper boudary. In the code supplied, it has the form:

$$f(\sigma) = \sin\left(\frac{\pi(\sigma - \sigma_T)}{2(1 - \sigma_T)}\right) \text{ for } \sigma \geq \sigma_T,\ f(\sigma) = 0 \text{ for } \sigma < \sigma_T,$$

(where σ_T is value of σ at height ZTROP, calculated within the subroutine). ΔT_{NS} and ΔT_{EP} represent the pole-pole and pole-equator temperature differences respectively. Note that ΔT_{EP} should be *positive* for the usual case where the poles are colder than the equator (this is not the strictly correct convention but seems less likely to cause accidents).

The initial actual temperature field is set to be equal to T_{ER} at all latitudes.

Called from INITFD when this is a "start" run.

SPOP - Organizes the output of spectral coefficients when required.

Called from MLTRI.

TSTEP - Performs a timestep using the adiabatic tendencies (advection, etc). The normal timestep is centred in time, and includes a Robert time filter to control time splitting. For KOUNT≤KITS, short initial timesteps, ie, an initial forward timestep followed centred steps, each twice its predecessor, are taken in order to initiate a run from data at only one time level. No Robert filter is included in the short steps. Details of the timestepping scheme are in Hoskins and Simmons (1978).

WRSPS - Writes spectral coefficients upto total wavenumber NCOEFF to file fort.2 every KOUNTP timesteps. NCOEFF=0 disables this type of output. The levels to be plotted are selected with the logical array LSPO, which is set to NL*.F. by default.

Called from SPOP.

XSECT - Rudimentary output subroutine. Prints latitude-sigma cross sections of [u], [T] and $[v^* T^*]$ to fort.2 every KOUNTP timesteps. Reads gridpoint data off fort.24. Note that each plot is for Gaussian latitudes and model levels. If the vertical grid is stretched, plots may need care in their interpretation. The subroutine is easy to edit in order to generate other diagnostics if needed.

Called from MLTRI.

10. Sample job deck

This sample job deck performs a start run of the SGCM for 100 days for a T21, 5 level global integration for terrestrial conditions, with a typical setting of the various parameters. Defaults have been assumed for a number of the namelist variables. The basic update library is held on my filestore in a file called /home/ymp8/ij2/uplib/ijsgcm.pl which has universal read access. Various utility subroutines are needed from the library kd/sgcm/sublib6.a.

```
# QSUB -r sgcm  -eo -lM 1200kw -lT 300
# QSUB
set +vSe
#---------------------sgcm----------------------------#
#Sample job - run SGCM with Newtonian cooling & friction#
#-----------------------------------------------------#
set +v
cd $TMPDIR   ; pwd
# This directory needs to have been created before the job runs
EXPDIR=$HOME/sgcm/1000.
cat << /EOF > updates
*ID JMOD
*D PARAM1.5,PARAM1.6
      PARAMETER(MM=21,NN=21,NHEM=2,NL=5,MOCT=1,MG=64,JG=16,NWJ2=121,
     1 NCRAY=64,JGL=JG)
*/=================================================================
*/ Any further modications you wish to make to the program should be
*/ included here.
*/=================================================================
/EOF
# data for run with typical terrestrial parameters
cat << /EOF > data
 &INPPL &END
 &INPRN
 DTEP=80.,DTNS=40.,RESTIM=5*30.,TFRC=4*0.,1.,
 TDISS=0.25,NDEL=8,
 &END
 &INPOP
 RNTAPE=1000.,
 KRUN=2400,TSPD=24.,
 KOUNTP=240,KOUNTE=24,KOUNTH=240,KOUNTR=600,
 NCOEFF=10,LSPO=4*.F.,.T.,
 &END
/EOF
# jcl section
while [ $? = 0 ]
do
# update library of code is on ijsgcm.pl
update -p /home/ymp8/ij2/uplib/ijsgcm.pl \
```

```
      -i updates -c prog \
      -f -o in id sq
cft77 -V -eaiz -a stack -A fast prog.f
# various utility subroutines taken off library sublib6.a
segldr -V -f indef -M ,stat -o prog \
      prog.o /home/ymp8/kd/sgcm/sublib6.a
rm prog.* updates
assign -a data fort.7
./prog || break
set +e
cat fort.2
# history and restart saved for further diagnostics
cp fort.9 $EXPDIR/hist
cp fort.11 $EXPDIR/restart
ls -alF $EXPDIR
exit 0
done
# error processing
set +e
cat fort.2
debug -B -s prog
exit 1
# end of job
```

11. Timings and storage requirements

For reference, some timings are given for the model at three typical resolutions. All refer to a global, MOCT=1 run, running on the Rutherford Laboratory CRAY YMP. Also given are values of the parameters to be set in deck PARAM1.

MM	NL	MG	JG	NWJ2	CPU time per step	CPU time per day
21	5	64	16	121	4.74×10^{-2}s	1.14s
31	10	96	24	256	0.305s	10.99s
42	15	128	32	462	0.608s	29.17s

At all three resolutions, the maximum field length (MFL parameter on the job statement) is 1.108 Mwords, a space required by the compiler/link editor. Compilation and initial set up take a significant fraction of the CPU time for short runs. It is quite difficult to estimate this accurately, since the CPU time of your job varies accordingly to the work load on the CRAY. The following estimates are intended as a guide; you should always leave a reasonable margin of uncertainty when submitting a long run to avoid it running out of time prematurely.

Resolution	CPU time for compile and setup.
T21L5	22s
T31L10	25s
T42L15	33s

The space needed to store history and restart files also needs to be estimated. Each history record has a length (3*NL+1)*NWJ2*2 + 4 Cray words (each of which is 8 bytes). Each restart record has a length (7*NL+2)*NWJ2*2 + 4 Cray words. When estimating how much data will fit onto a tape, allow for dead space as data is blocked onto the tape, and remember that the length of tapes is somewhat variable. Play safe with your precious data! For those unable to do arithmetic, these figures are summarized for the 3 standard resolutions in the following table.

Resolution	History records		Restart records	
	Kwords	Kbytes	Kwords	Kbytes
T21L5	3.9	31	9.0	72
T31L10	15.9	127	39.9	295
T42L15	42.5	340	98.9	791

These tables will give a rough indication of computer resources needed for each resolution. But to refine these indications, you are advised to experiment with small initial jobs before embarking on a long run at a new resolution.

12. Associated utilities

A number of subroutines have been written which may prove useful for additional diagnostics and for setting up the initial fields for a run. It is left to the user to edit in the appropriate calls, and it is assumed that the reader will wish to customize these subroutines for his own particular requirements. If you have other useful utilities subroutines, we will be pleased to consider including them in later versions of the library.

CALL GDPLOT(IFLD,LVL)

PURPOSE: Provides a quick gridpoint printout of a field on a model level. Each gridpoint is represented by a single character in the range I, H, G, F, E, D, C, A, 0, 1, 2, 3, 4, 5, 6, 7, 8, or 9 where I represents the minimum value and 9 represents the maximum. The field is printed with latitude *across* the page and longitude *down* the page, so that fields will fit across the width of an 80 column screen or printer for resolutions up to T42.

IFLD - Determines the field to be printed, as follows:

1 - Absolute vorticity,
2 - Divergence,
3 - Zonal wind,
4 - Meridional wind,
5 - Temperature,
6 - Surface pressure.

LVL - The model level to be printed, using the usual convention, ie, 1 is the top level and NL is the lowest level. The value is irrelevant for surface pressure, but should be in the range 1 to NL.

==

CALL LSTOSP(FLDX,FLDSP,ISYM)

PURPOSE: Takes data on a σ-latitude cross section (on a grid consisting of model levels and Gaussian latitudes) and creates the appropriate spectral coefficients.

FLDX(JGT,NL) - REAL array containing the input gridpoint data.

FLDSP(IGB) - COMPLEX array containing the output spectral coefficients.

ISYM - a switch which determines whether the odd coefficients are stored before the even (ISYM = 1; eg, vorticity fields) or the even fields before the odd (ISYM = 0; eg, temperature, divergence, orography or surface pressure fields).

==

CALL LLTOSP(FLD,FLDSP,ISYM)

PURPOSE: Takes data on a model level (ie, a Gaussian latitude- longitude grid) and creates the corresponding spectral coefficients.

FLD(MG,JGT) - REAL array containing the input gridpoint data.

FLDSP(IGA) - COMPLEX array containing the output spectral coefficients.

ISYM - switch determining order of odd and even coefficients (see description of ZONTOSP).

==

The following pair of subroutines are used to introduce a vertical diffusion to vorticity and divergence.

CALL VDIFF

PURPOSE: Adds tendencies due to vertical diffusion into the arrays ZT and DT. The vertical diffusion is calculated in spectral space, but can vary in the vertical; the mesh may have an arbitrary shape. The vertical diffusion has the form:

$$\frac{\partial \zeta}{\partial t} = \frac{\sigma}{H^2} \frac{\partial}{\partial \sigma} \left(\sigma K \frac{\partial \zeta}{\partial \sigma} \right), \quad \frac{\partial D}{\partial t} = \frac{\sigma}{H^2} \frac{\partial}{\partial \sigma} \left(\sigma K \frac{\partial D}{\partial \sigma} \right).$$

$H = RT_R/g$ is a pressure scale height; this is assumed constant, which is the major approximation in the formulation of the term. This is not regarded as very significant. Since the term is linear in ζ and D, it can be applied in spectral space, with considerable savings in time and numerical complexity. In finite difference terms, the scheme can be summarized by:

$$\frac{\partial}{\partial t} \zeta_n^m(\sigma_l,t) = a_l \zeta_n^m(\sigma_{l+1},t) + b_l \zeta_n^m(\sigma_l,t) + c_l \zeta_n^m(\sigma_{l-1},t).$$

and similarly for the divergence coefficients D_n^m.

The subroutine needs a new common block:

COMMON/CVDIFF/DFV(NL),DFA(NL),DFB(NL),DFC(NL)

(All variables REAL).

DFV	Array of vertical diffusion coefficients.
DFA	Array of a_l.
DFB	Array of b_l.
DFC	Array of c_l.

The call to this subroutine should be inserted between the calls to DIFUSE and DSTEP. During the set up, it is essential to call subroutine INITVD which sets up the common block CVDIFF.

CALL INITVD

Initializes the arrays in the common block CVDIFF before calls to VDIFF are made. Should be called *after* the call to INITPM. The subroutine reads an additional namelist, INPVD.

NAMELIST/INPVD/DFV

DFV A REAL array of dimension NL containing the vertical diffusion coefficients for each model level. These are de-dimensionalized (by $H^2\Omega$) and used to compute a_l, b_l and c_l. Default is NL*2.5m^2s^{-1}.

===

13. References

Blackburn, M. (1985): Program description for the multilevel global spectral model. Department of Meteorology informal note, 37pp.

Eliasen, E., B. Machenauer & E. Rasmussen (1970): On a numerical method for thei ntegration of the hydrodynamical equations with a spectral representation of the horizontal fields. Institute of Theoretical Meteorology, University of Copenhagen, Report No 2.

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.

James, I.N. & L.J. Gray (1986): Concerning the effect of surface drag on the circulation of a baroclinic planetary atmosphere. Q.J. Roy. Met. Soc., **112**, 1231-1250.

James, I.N. & P.M. James (1992): Spatial structure of ultra-low frequency variability of the flow in a simple global circulation model. Q.J. Roy. Met. Soc., **118**, 1211-1234.

Orzsag, S.A. (1970): Transform method for calculation of vector coupled sums: application to the spectral form of the vorticity equation. J. Atmos. Sci., **27**, 890-895.

Press, W.H., B.P. Flannery, S.A. Teukolsky & W.T. Vetterling (1989): Numerical recipes - the art of scientific computing. Cambridge University Press, 702pp.

Simmons, A.J. & B.J. Hoskins (1978): The life cycles of some nonlinear baroclinic waves. J. Atmos. Sci., **35**, 414-432.