NEW FLUX PROGRAM: BGFLUX2.

INDIRECT LEGENDRE TRANSFORMS

SUMMARY.

The ED=1 flux programs have a very large memory requirement at high resolution, larger than the ECMWF model itself. Thus a T106 global, 19 level diagnostic job requires ≥6 Mwords memory. Since the flux program is single tasked this entails a very inefficient use of the Cray X-MP/48, plus there is little room for expansion to more levels or chemical tracers.

The following pages describe a modification to the inverse Legendre Transform code (spectral to Fourier) which reduces memory requirements substantially while not incurring a computational overhead.

The original update was devised and written by Paul Valdes. Efficiency and clarification changes were added by me.

Contents.

Mike Blackburn.

2.2.89.

THEORY.

Transform fields from spectral to Fourier space, before using the FFT to transform to grid point space. Several required Fourier fields are derivatives or functions of the available spectral fields.

a) Spectral → Even/Odd contributions to Fourier at $+\mu$.

b) Sum/Difference → Fourier coeffs. at $\pm\mu$.

$$A_m(+\mu) = A_m^E(\mu) + A_m^O(\mu)$$
$$A_m(-\mu) = A_m^E(\mu) - A_m^O(\mu).$$

since $A_m^E(-\mu) = A_m^E(\mu)$, $A_m^O(-\mu) = -A_m^O(\mu)$.

i) Basic transform:

$$a(\lambda,\mu) = \sum_{m=0}^{M}{}' A_m(\mu) e^{im\lambda}$$
$$A_m(\mu) = \sum_{n=m}^{N}{}' A_n^m P_n^m(\mu)$$

ii) $$b(\lambda,\mu) = \frac{\partial a}{\partial \lambda} = \sum_m{}' B_m(\mu) e^{im\lambda}$$

$$B_m(\mu) = im\, A_m(\mu)$$

iii) $$c(\lambda,\mu) = (1-\mu^2)\frac{\partial a}{\partial \mu} = \sum_m{}' C_m(\mu) e^{im\lambda}$$

$$C_m(\mu) = \sum_n{}' A_n^m (1-\mu^2)\frac{\partial P_n^m}{\partial \mu}$$

iv) $$d(\lambda,\mu) = \nabla^{-2} a = \sum_m{}' D_m(\mu) e^{im\lambda}$$

$$D_m(\mu) = \sum_n{}' -A_n^m \frac{P_n^m}{n(n+1)}$$

since $\nabla^2 Y_n^m = -n(n+1) Y_n^m$ where $Y_n^m = P_n^m e^{im\lambda}$

v) $$e(\lambda,\mu) = \frac{\partial}{\partial \lambda}(\nabla^{-2} a) = \sum_m{}' E_m(\mu) e^{im\lambda}$$

$$E_m(\mu) = im \sum_n{}' -A_n^m \frac{P_n^m}{n(n+1)} = im\, D_m(\mu)$$

vi) $$f(\lambda,\mu) = (1-\mu^2)\frac{\partial}{\partial \mu}(\nabla^{-2} a) = \sum_m{}' F_m(\mu) e^{im\lambda}$$

$$F_m(\mu) = \sum_n{}' -A_n^m \frac{(1-\mu^2)}{n(n+1)}\frac{\partial P_n^m}{\partial \mu}$$

REQUIRED TRANSFORMS.

Grid point fields required in the (moist) flux program are as follows:

ζ, D, T, q, H, $\ln p_*$ $\quad \frac{\partial T}{\partial \lambda}$, $\frac{\partial \ln p_*}{\partial \lambda}$, $(1-\mu^2)\frac{\partial T}{\partial \mu}$, $(1-\mu^2)\frac{\partial \ln p_*}{\partial \mu}$ } for PV calc.

$\psi = \nabla^{-2}\zeta \quad (\chi = \nabla^{-2} D)$

$$U = U_\psi + U_\chi = -(1-\mu^2)\frac{\partial \psi}{\partial \mu} + \frac{\partial \chi}{\partial \lambda}$$

$$V = V_\psi + V_\chi = \frac{\partial \psi}{\partial \lambda} + (1-\mu^2)\frac{\partial \chi}{\partial \mu}.$$

A zonally averaged restoration-temperature field is also required.

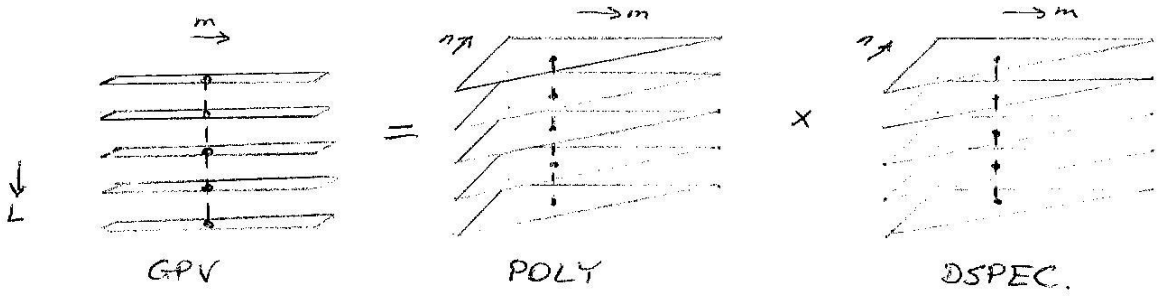
Spectral Field	Fourier/G.P. Field.	Method.	Spectral Symmetry.	G.P. Sym.
D	$U_\chi = \frac{\partial \chi}{\partial \lambda}$	$-im\sum'_n \frac{P_n^m}{n(n+1)} D_n^m$	E	E
ζ_{REL}	$V_\psi = \frac{\partial \psi}{\partial \lambda}$	$-im\sum'_n \frac{P_n^m}{n(n+1)} \zeta_n^m$	O	O
ζ_{REL}	$U_\psi = -(1-\mu^2)\frac{\partial \psi}{\partial \mu}$	$+\sum'_n \frac{(1-\mu^2)}{n(n+1)}\frac{\partial P_n^m}{\partial \mu}\zeta_n^m$	O	E
D	$V_\chi = (1-\mu^2)\frac{\partial \chi}{\partial \mu}$	$-\sum'_n \frac{(1-\mu^2)}{n(n+1)}\frac{\partial P_n^m}{\partial \mu} D_n^m$	E	O
ζ_{ABS}	ζ_{ABS}	$\sum'_n P_n^m \zeta_n^m$	O	O
D	D	$\sum'_n P_n^m D_n^m$	E	E
T	T	$\sum'_n P_n^m T_n^m$	E	E
q	q	$\sum'_n P_n^m q_n^m$	E	E
H	H	$\sum'_n P_n^m H_n^m$	E	E
T	$\frac{\partial T}{\partial \lambda}$	$im\sum'_n P_n^m T_n^m$	E	E
T	$(1-\mu^2)\frac{\partial T}{\partial \mu}$	$\sum'_n (1-\mu^2)\frac{\partial P_n^m}{\partial \mu} T_n^m$	E	O
$\ln p_*$	$\frac{\partial \ln p_*}{\partial \lambda}$	$im\sum'_n P_n^m (\ln p_*)_n^m$	E	E
$\ln p_*$	$(1-\mu^2)\frac{\partial \ln p_*}{\partial \mu}$	$\sum'_n (1-\mu^2)\frac{\partial P_n^m}{\partial \mu}(\ln p_*)_n^m$	E	O
$\ln p_*$	$\ln p_*$	$\sum'_n P_n^m (\ln p_*)_n^m$	E	E
ζ_{REL}	$\psi = \nabla^{-2}\zeta_{REL}$	$-\sum'_n \frac{P_n^m}{n(n+1)}\zeta_n^m$	O	O

METHOD IN BGCMFLUX, ED = 1.

Attain speed by performing all transforms in a single pass, with inner (vector) loop over levels and even/odd contributions : vector length $(11 \times NL + 2) \times NHEM$.

This requires copies of spectral fields for all transforms involving derivatives etc.

Also requires a large POLY array, same (real) size as the (complex) spectral data, to contain Legendre functions and their derivatives.



In a global run adjacent "levels" (L) contain even and odd contributions or half-triangles for a given level.

In a hemispheric run each field contains only even or odd contributions at each level.

Main Prog.

i) Make spectral copies: -

DDA	ZDA	ZDB	DDB	TDA	TDB	SPDU
$= (-i)D$	$(-i)Z$	Z	$-D$	$(i)T$	T	SP
for U_χ	V_ψ	U_ψ	V_χ	$\frac{\partial T}{\partial \lambda}$	$(1-\mu^2)\frac{\partial T}{\partial \mu}$	$(1-\mu^2)\frac{\partial \ln p_*}{\partial \mu}$.

NHEXP

ii) Prepare POLY array, size $(NWJ2 \times NHEM \times (11 \times NL + 2))$
i.e. contains one element for every spectral coefficient.

iii) Main transform : $A_m(\mu) = \sum_n A_n^m f(P_n^m(\mu))$, for all m, levels, fields.

iv) Copy $(\ln p_*)_m$ into PLG array.
Then multiply PMG by (im) to form $\frac{\partial \ln p_*}{\partial \lambda}$.
(other λ-derivative, for T, was performed by modifying the Legendre function $(\times m)$.

Sum wind components into UG, VG.

EONS

v) Sum and difference half transforms for $A_m(\pm\mu)$.

FLUX PROGRAM : LEGENDRE TRANSFORMS

Spectral → Half-Transform. BFLUX/BGCMFLUX, ED=1.

	SPECTR	GRIDP	Symm.	Spectral fn.	H.T. fn.	(POLT name)	Function.	METHOD
	DDA	UGT	EVEN	$\frac{-im}{n(n+1)} P_n^m D_n^m$	$[U_\chi]_m = [\frac{\partial \chi}{\partial \lambda}]_m$	DP	$\frac{-im}{n(n+1)} P_n^m$	CMPLX(0,-1)* D * DP
(*)	ZDA	VGT	ODD	$\frac{-im}{(n+1)n} P_n^m \zeta_n^m$	$[V_\psi]_m = [\frac{\partial \psi}{\partial \lambda}]_m$	DP	$\frac{-im}{n(n+1)} P_n^m$	CMPLX(0,-1)* Z * DP
*	ZDB	UG	ODD	$\frac{(1-\mu^2)\partial P_n^m}{n(n+1)\partial \mu} \zeta_n^m$	$[U_\psi]_m = [-(1-\mu^2)\frac{\partial \psi}{\partial \mu}]_m$	DQ	$\frac{(1-\mu^2)\partial P_n^m}{n(n+1)\partial \mu}$	Z * DQ
	DDB	VG	EVEN	$-\frac{(1-\mu^2)\partial P_n^m}{n(n+1)\partial \mu} D_n^m$	$[V_\chi]_m = [(1-\mu^2)\frac{\partial \chi}{\partial \mu}]_m$	DQ	$\frac{-(1-\mu^2)\partial P_n^m}{n(n+1)\partial \mu}$	- D * DQ
	Z	ZG	ODD	$\zeta_n^m P_n^m$	$[\zeta]_m$	ALP	P_n^m	Z * ALP
	D	DG	EVEN	$D_n^m P_n^m$	$[D]_m$	ALP	P_n^m	D * ALP
	T	TG	EVEN	$T_n^m P_n^m$	$[T]_m$	ALP	P_n^m	T * ALP
	Q	QG	EVEN	$q_n^m P_n^m$	$[Q]_m$	ALP	P_n^m	Q * ALP
	H	HG	EVEN	$H_n^m P_n^m$	$[H]_m$	ALP	P_n^m	H * ALP
	TDA	TXG	EVEN	$im P_n^m T_n^m$	$[\frac{\partial T}{\partial \lambda}]_m$	ALP	$im P_n^m$	T * ALP * CMPLX(0,m)
	TDB	TYG	ODD	$(1-\mu^2)\frac{\partial P_n^m}{\partial \mu} T_n^m$	$[(1-\mu^2)\frac{\partial T}{\partial \mu}]_m$	DALP	$(1-\mu^2)\frac{\partial P_n^m}{\partial \mu}$	T * DALP
	SP	PMG	EVEN	$im(\ln p_*)_n^m P_n^m$	$[\frac{\partial \ln p_*}{\partial \lambda}]_m$	ALP	$im P_n^m$	CMPLX(0,m) * SP * ALP
	SPDL	PJG	ODD	$(1-\mu^2)\frac{\partial P_n^m}{\partial \mu}(\ln p_*)_n^m$	$[(1-\mu^2)\frac{\partial \ln p_*}{\partial \mu}]_m$	DALP	$(1-\mu^2)\frac{\partial P_n^m}{\partial \mu}$	SP * DALP
	(SP)	PLG	EVEN	$(\ln p_*)_n^m P_n^m$	$[\ln p_*]_m$	ALP	P_n^m	SP * ALP

* Remove planetary vorticity before using ζ_n^m (for $\frac{\partial}{\partial \mu}$ fns.)

$ALP \equiv P_n^m$

$DALP \equiv (1-\mu^2)\frac{\partial P_n^m}{\partial \mu}$

$DP \equiv \frac{m}{n(n+1)} P_n^m \Rightarrow -m[\nabla^{-2}\,\zeta\,]$

$DQ \equiv \frac{(1-\mu^2)}{n(n+1)}\frac{\partial P_n^m}{\partial \mu} \Rightarrow -(1-\mu^2)\frac{\partial}{\partial \mu}[\nabla^{-2}(\,\zeta\,)]$

HXPOOT	*Z	UZG	ODD	$-\frac{P_n^m}{n(n+1)} \zeta_n^m$	$[\psi]_m = [\nabla^{-2}\zeta]_m$	(RSQ)	$-\frac{P_n^m}{n(n+1)}$	-ALP * RSQ * Z
HEXRES	$T_{(m=0)}$	TORES	EVEN	$\Rightarrow P_n^0 T_n^0$	$[T]_0$ *	ALP	P_n^0	TRES * ALP

BGFLUX2 : LEGENDRE FUNCTIONS

MB 29.6.94

LGNDRE returns 2 arrays of Associated Legendre Functions and their derivatives:

ALP	$\equiv$	$P_n^m(\mu)$.
DALP	$\equiv$	$(1-\mu^2)\frac{dP_n^m}{d\mu}$.

INILAT derives 2 further arrays for transforms involving ∇^{-2}:

RLP	$=$	$-\frac{1}{n(n+1)} P_n^m(\mu)$
RDLP	$=$	$-\frac{1}{n(n+1)}(1-\mu^2)\frac{dP_n^m}{d\mu}$.

'Ordering' All arrays are stored with increasing total wavenumber 'n' varying fastest, within increasing zonal wavenumber 'm'. Even and odd coefficients alternate.
A jagged triangular truncation is used, having equal numbers of even and odd coefficients for each zonal wavenumber.

Transforms: Note that $\nabla^2\left(\sum_m' \sum_n' A_n^m Y_n^m\right) = -\frac{n(n+1)}{a^2}\left(\sum_m' \sum_n' A_n^m Y_n^m\right)$

i.e. $\frac{-a^2}{n(n+1)}$ is the eigenvalue of the ∇^{-2} operator.

If $A_n^m = \int_{-1}^{1} A_m(\mu) P_n^m(\mu)\, d\mu$, $A_m = \frac{1}{2\pi}\int_0^{2\pi} f(\lambda,\mu) e^{-im\lambda} d\lambda$,

then:

ALP	transforms	$A_n^m \longleftrightarrow [f]_m$
DALP	transforms	$A_n^m \longleftrightarrow \left[(1-\mu^2)\frac{\partial f}{\partial \mu}\right]_m$
RLP	transforms	$A_n^m \longleftrightarrow [\nabla^{-2} f]_m$
RDLP	transforms	$A_n^m \longleftrightarrow \left[(1-\mu^2)\frac{\partial}{\partial \mu}\nabla^{-2} f\right]_m$.

In practice, Gaussian weights $w_j' = \frac{2}{1-\mu^2} w_j$ are used when creating the A_n^m, introducing/cancelling the $(1-\mu^2)$ here.

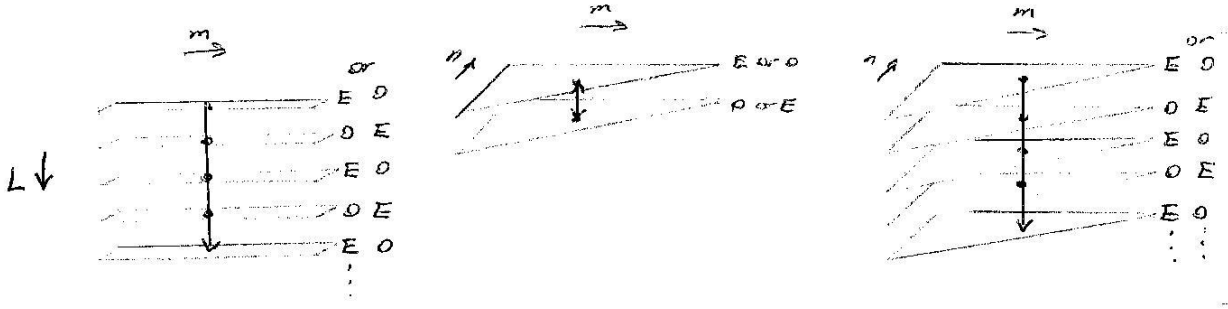
METHOD IN BGCMFLUX, ED=2 etc.

Save memory by omitting copies of spectral arrays and by shortening the POLY array to a single level, length (NWJ2 x NHEM).

Now only transforms involving a single polynomial function and spectral symmetry can be performed in a single pass. Thus the vector efficiency is reduced, but the streamfunction transform can be performed by the new routine and the old inefficient MXPOUT removed.

By using separate faster code for hemispheric runs (see below), in fact both hemispheric and global runs (for 15 level tests) are faster than originally, as well as giving the large memory saving.

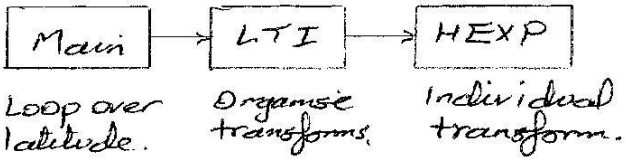
The inner (vector) loop is still over levels, but now generally for a single field unless several having the same symmetry and requiring the same polynomial function can be placed sequentially in common blocks SPECTR and GRIDP.



For a global run, adjacent values of "L" contain even and odd contributions. The POLY array has size (NWJ2, NHEM), so must be accessed (..., 1, 2, 1, 2, 1 ....)

For a hemispheric run, a field has only even or odd contributions, so there is a single (NWJ2) length of POLY. Separate code for NHEM=1 for the main loop avoids calculation of the (..., 1, 2, 1, 2, 1....) counter, so improving efficiency.

Program Flow :



Routine LTI : Organises transforms for single ±μ latitude.

Spectral Field.	Fourier Field.	Trans. Type.	Method.	Arrays.	Spec. Sym.	Fourier Sym.
ζ	ζ	1	$\zeta_m = \Sigma_n' P_n^m \zeta_n^m$	ZG = ALP × Z	O	O

Remove planetary vorticity from ζ_0'.

ζ	$-U_\psi = (1-\mu^2)\frac{\partial \psi}{\partial \mu}$	7	$A_m = \Sigma_n' \frac{-(1-\mu^2)}{n(n+1)} \frac{\partial P_n^m}{\partial \mu} \zeta_n^m$	UG = RDLP × Z	O	E
ζ	$\psi = \nabla^{-2}\zeta$	5	$A_m = \Sigma_n' \frac{-P_n^m}{n(n+1)} \zeta_n^m$	PSIG = RLP × Z	O	O
D	$\chi = \nabla^{-2} D$	6	$A_m = \Sigma_n' \frac{-P_n^m}{n(n+1)} D_n^m$	TG (temp) = RLP × D	E	E
D	$V_\chi = (1-\mu^2)\frac{\partial \chi}{\partial \mu}$	8	$A_m = \Sigma_n' \frac{-(1-\mu^2)}{n(n+1)} \frac{\partial P_n^m}{\partial \mu} D_n^m$	VG = RDLP × D	E	O

Sum wind components:

$$U = U_\psi + U_\chi = -(-U_\psi) + \frac{\partial \chi}{\partial \lambda}$$

$$UG = -UG + \underset{(lim)}{CMPA} \times TG.$$

$$V = V_\psi + V_\chi = \frac{\partial \psi}{\partial \lambda} + V_\chi$$

$$VG = \underset{(lim)}{CMPA} \times PSIG + VG$$

D, q, H, T, lnP*	D, q, H, T, lnP*	2	$A_m = \Sigma_n' P_n^m A_n^m$	DG, QG, HG, TG, PLG = ALP × D, Q, H, T, SP	E	E
T, lnP*	$(1-\mu^2)\frac{\partial T}{\partial \mu}$, $(1-\mu^2)\frac{\partial \ln P_*}{\partial \mu}$	4	$A_m = \Sigma_n' (1-\mu^2)\frac{\partial P_n^m}{\partial \mu} A_n^m$	TXG, PMG = DALP × T, SP	E	O

Take λ-derivatives :

$$\left(\frac{\partial T}{\partial \lambda}, \frac{\partial \ln P_*}{\partial \lambda}\right)_m = (im) \times (T, \ln P_*)_m$$

$$(TXG, PMG) = CMPA \times (TG, PLG)$$

Add planetary vorticity back to ζ_0'.

Polynomial Functions.

$$ALP \equiv P_n^m(\mu) \quad : \quad A_m^m \Rightarrow a$$

$$DALP \equiv (1-\mu^2)\frac{\partial P_n^m}{\partial \mu} \quad : \quad A_n^m \Rightarrow (1-\mu^2)\frac{\partial a}{\partial \mu}.$$

$$RLP = \frac{-P_n^m}{n(n+1)} \quad : \quad A_n^m \Rightarrow \nabla^{-2} a.$$

$$RDLP = \frac{-(1-\mu^2)}{n(n+1)}\frac{\partial P_n^m}{\partial \mu} \quad : \quad A_n^m \Rightarrow (1-\mu^2)\frac{\partial}{\partial \mu}(\nabla^{-2} a).$$

Routine SFGCPR. : Modified version, 30.6.94.

Spectrally analyse gridded winds to create gridded ψ, χ, ζ, D.

- Reads gridded winds from scratch files.
- Writes gridded ψ, χ, ζ, D to scratch files.

$$\zeta = \underset{\sim}{k}\cdot\nabla\wedge\underset{\sim}{V} = \frac{1}{1-\mu^2}\frac{\partial V}{\partial\lambda} - \frac{\partial U}{\partial\mu} = \nabla^2\psi$$

$$D = \nabla\cdot\underset{\sim}{V} = \frac{1}{1-\mu^2}\frac{\partial U}{\partial\lambda} + \frac{\partial V}{\partial\mu} = \nabla^2\chi.$$

COMMON GRIDP :	UG,	VG,	ZG,	DG
Equivalences :	PSIG,	CHIG,	UIMG,	VIMG
COMMON SPECTR :	Z,	D,	T,	Q
Equivalences :	—	—	"U"	"V"

1 Preset spectral arrays to zero: Z, D, U, V.

2 Direct transform in latitude loop.

- FFT : UG, VG only. $\longrightarrow U_m, V_m$.
- Copies : UIMG, VIMG $= \frac{im}{1-\mu^2}U_m, \frac{im}{1-\mu^2}V_m$.
- Legendre transform :

Fourier Field	Spectral Fld.	Trans Type	Method.	Arrays	Four. Sym.	Spec Sym.
U_m	$\partial U/\partial\mu$	3	$U_n^m = \sum_j' U_m(1-\mu^2)\frac{\partial P_n^m}{\partial\mu}w_j'$	U = UG*DALP *(-AW)	E	O
V_m	$\frac{\partial V}{\partial\mu}$	4	$V_n^m = -\sum_j' V_m(1-\mu^2)\frac{\partial P_n^m}{\partial\mu}w_j'$	V = VG*DALP *(-AW)	O	E
$\frac{im}{1-\mu^2}U_m$	$\frac{1}{1-\mu^2}\frac{\partial U}{\partial\lambda}$	2	$D_n^m = \sum_j' im U_m P_n^m w_j'$	D = UIMG*ALP *CSSQ*AW	E	E
$\frac{im}{1-\mu^2}V_m$	$\frac{1}{1-\mu^2}\frac{\partial V}{\partial\lambda}$	1	$Z_n^m = \sum_j' im V_m P_n^m w_j'$	Z = VIMG*ALP *CSSQ*AW	O	O

3 Combine spectral terms for ζ, D. Add planetary vorticity. Set ζ_0^0, D_0^0 to zero.

4 Indirect Legendre transform in latitude loop:

Spectral Fld	Fourier Field	Trans Type	Method.	Arrays	Spec. Sym.	Four. Sym.
ζ_A	ζ_A	1	$\zeta_m = \sum_n' \zeta_n^m P_n^m$	ZG = Z*ALP	O	O
D	D	2	$D_m = \sum_n' D_n^m P_n^m$	DG = D*ALP	E	E
ζ	$\psi = \nabla^{-2}\zeta_R$	5	$\psi_m = \sum_n' \frac{1}{n(n+1)}\zeta_n^m P_n^m$	PSIG = Z*RLP	O	O
D	$\chi = \nabla^{-2}D$	6	$\chi_m = \sum_n' \frac{1}{n(n+1)}D_n^m P_n^m$	CHIG = D*RLP	E	E

Routine HEXP

: Perform transform for a set of fields having same symmetry and polynomial type. One call per latitude (pair).

CALL HEXP(SV, GV, NLS, ITYPE)

SV (NWJ2 × NHEM × NLS) Complex array of spectral data.

GV ($\frac{NGPP}{2}$ × NHEM × NLS) Complex array to receive Fourier data.

NLS Number of levels of data.

ITYPE Type and symmetry of transform. Even/odd for even/odd spectral field.

ITYPE	Polynomial type	Transform type.	Spectral Sym.	Four. Sym
1	$ALP \equiv P_n^m$	$A_n^m \rightarrow a$	O	O
2			E	E
3	$DALP \equiv (1-\mu^2)\frac{\partial P_n^m}{\partial \mu}$	$A_n^m \rightarrow (1-\mu^2)\frac{\partial a}{\partial \mu}$	O	E
4			E	O
5	$RLP \equiv \frac{-P_n^m}{n(n+1)}$	$A_n^m \rightarrow \nabla^{-2} a$	O	O
6			E	E
7	$RDLP \equiv \frac{-(1-\mu^2)\partial P_n^m}{n(n+1)\partial \mu}$	$A_n^m \rightarrow (1-\mu^2)\frac{\partial}{\partial \mu}(\nabla^{-2} a)$	O	E
8			E	O

Method.

i) Preset Fourier array to zero.

ii) Use ITYPE to define transform type and symmetries.

IGPAR = (0,1) for (even, odd) spectral

IGPAR = (0,1) for (even, odd) Fourier.

IGPAR = ISPAR unless the transform involves a $\partial/\partial\mu$.

iii) Calculate POLY(NWJ2, NHEM).

ISPAR = (0;1) → (even before odd; odd before even)

iv) Main transform loop: inner loop over levels and even/odd for each (m,n)

$$A_m^{E,O}(\mu) = \sum_n' A_n^{m\,E,O}\left(P_n^{m\,E,O}(\mu)\right)^*$$

v) Sum (difference in a global run for the Fourier coefficients $A_m(\pm\mu)$.

Routine HANAL: Perform Fourier to spectral transform for a set of fields having the same symmetry and Legendre Function type. One call per latitude (pair).

CALL HANAL (GV, SV, NLS, ITYPE)

SV(NWJ2 * NHEM * NLS)	Complex array of spectral data.
GV($\frac{MGPP}{2}$ * NHEM * NLS)	Complex array of Fourier data.
NLS	Number of levels.
ITYPE	Type & symmetry of transform. Even/odd for even/odd spectral field.

GV is input : Fourier coefficients at latitudes $\pm\mu$.
SV is output : contribution to spectral coefficients.

	ITYPE	Polynomial type	Transform type	Spectral Sym.	Fourier Sym.
	1 2	ALP $\equiv P_n^m$	$a_m \rightarrow [a]_n^m$	O E	O E
	3 4	DALP $\equiv (1-\mu^2)\frac{dP_n^m}{d\mu}$	$a_m \rightarrow \left[\frac{\partial a}{\partial \mu}\right]_n^m$	O E	E O
*	5 6	RLP $\equiv \frac{-P_n^m}{n(n+1)}$	$a_m \rightarrow [\nabla^{-2} a]_n^m$	O E	O E
*	7 8	RDLP $\equiv \frac{-(1-\mu^2)}{n(n+1)}\frac{dP_n^m}{d\mu}$	$a_m \rightarrow \left[\frac{\partial}{\partial \mu}\nabla^{-2} a\right]_n^m$	O E	E O

* 5–8 not coded in original version.

Method

o) Spectral array SV must be preset outside latitude loop from which HANAL is called (zero preset).

i) Use ITYPE to define transform type and symmetries:
ISPAR = (0,1) for (even, odd) spectral
IGPAR = (0,1) " " " Fourier.
These differ when the transform involves a $\partial/\partial\mu$.

ii) For a global run, average the Fourier coefficients at $\pm\mu$ to form even and odd contributions at $+\mu$.

iii) Set up POLY(NWJ2, NHEM) for the desired transform.
ISPAR = (0,1) → (even before odd, odd before even).

iv) Main transform loop: contribution to Gaussian integral

Gaussian Integrals in HANAL:

1,2 $$A_n^m = \sum_{j=1}^{JG} \bar{A}_m (1-\mu^2) P_n^m w_j' = \sum_{j=1}^{2xJG} A_m P_n^m w_j \equiv \int_{-1}^{1} A_m P_n^m d\mu.$$

(CSSQ)(ALP)(AW)

3,4 $$A_n^m = \sum_{j=1}^{JG} -\bar{A}_m (1-\mu^2) \frac{dP_n^m}{d\mu} w_j' = \sum_{j=1}^{2xJG} -A_m \frac{dP_n^m}{d\mu} w_j \equiv +\int_{-1}^{1} \frac{\partial A_m}{\partial \mu} P_n^m d\mu.$$

(DALP) (-AW)

5,6 $$A_n^m = \sum_{j=1}^{JG} -\frac{\bar{A}_m (1-\mu^2)}{n(n+1)} P_n^m w_j' = \sum_{j=1}^{2xJG} \frac{-A_m}{n(n+1)} P_n^m w_j \equiv +\int_{-1}^{1} (\nabla^{-2} A_m) P_n^m d\mu.$$

(CSSQ)(RLP)(AW)

7,8 $$A_n^m = \sum_{j=1}^{JG} +\frac{\bar{A}_m (1-\mu^2)}{n(n+1)} \frac{dP_n^m}{d\mu} w_j' = \sum_{j=1}^{2xJG} +\frac{A_m}{n(n+1)} \frac{dP_n^m}{d\mu} w_j = +\int_{-1}^{1} \frac{\partial}{\partial \mu} (\nabla^{-2} A_m) P_n^m d\mu.$$

(DRLP) (-AW)

Note The integration by parts in the μ-derivative transforms (3,4,7,8) assumes that the zonal mean of the field vanishes at the poles, ie. $A_m(\mu = \pm 1) = 0$.
This is true of the wind components (u,v) and all fluxes.

$\bar{A}_m(\pm\mu)$ are the even/odd contributions to the Fourier transform:

$$\bar{A}_m = A_m^{E,O}(\pm\mu) = \begin{cases} \frac{1}{2}[A_m(+\mu) + A_m(-\mu)], & \text{even Fourier} \\ \frac{1}{2}[A_m(+\mu) - A_m(-\mu)], & \text{odd Fourier.} \end{cases}$$

Then the Gaussian integral is over the northern hemisphere only; w_j' contains a factor of 2 to normalise the $\int_{-1}^{1}() d\mu$ integral correctly.