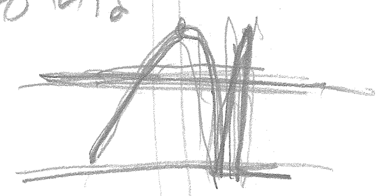
実験計画（９２年４月１８日）

＜ Ａ ＞完全非線形モデル（Ｔ８５）

（１）初期値
　第１セット：　Ｙ＆Ｙと同じ初期値で帯状成分（ｍ＝０）がゼロ

　第２セット：　エネルギースペクトルがホワイト

　第３セット：　ピーター ラインズ型（ある波数と次の波数だけに与える）



（２）実験パラメータ、実験内容はＹ＆Ｙと同じ

（３）データ解析：
　ア）全エネルギー、全エンストロフィーの時間変化
　イ）エネルギースペクトル
　ウ）帯状平均各運動量と波の活動度に関する収支解析
　　　帯状平均した、帯状角運動量、なみの活動度（Ａ）、
　　　絶対渦度、ｄ（絶対渦度）／ｄφ。
　　　帯状角運動量となみの活動度（Ａ）時間変化、
　　　ＥＰフラックスダイバージェンス（ＰＶフラックス）、粘性項
　エ）ピーター ラインズのパラメータの緯度時間依存性

　＃　４８ケースのアンサンブル平均、ウ）は１リアライゼーションの図。

（４）特別ラン
　ＥＰフラックスダイバージェンス（ＰＶフラックス）の時間積分値

＜ Ｂ ＞弱非線形モデル（Ｔ８５）
（１）初期値
　第１セット：　Ｙ＆Ｙと同じ初期値で違いを見る

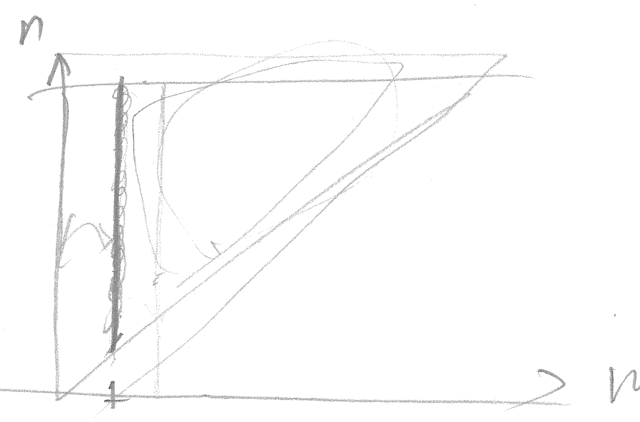
　第２セット：　きれいな場（ある波数の波・ガウス渦）

（２）実験パラメータ、実験内容はＹ＆Ｙと同じ

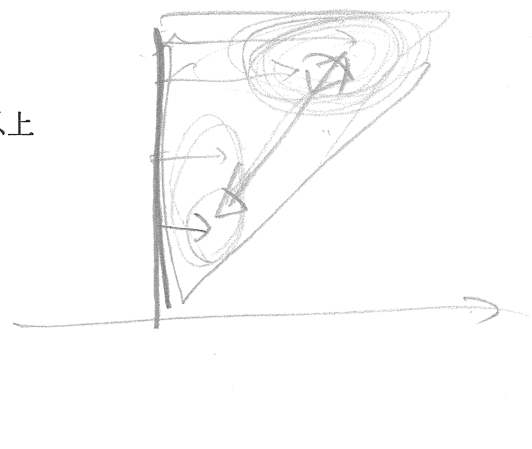
　＃ 第２セットは、大きな粘性係数（完全非線形の結果より決める）

（３）データ解析：
　ア）　帯状平均各運動量と波の活動度に関する収支解析
　　　　帯状平均した、帯状角運動量、なみの活動度（Ａ）、
　　　　絶対渦度、ｄ（絶対渦度）／ｄφ。
　　　　帯状角運動量となみの活動度（Ａ）時間変化、
　　　　ＥＰフラックスダイバージェンス（ＰＶフラックス）、粘性項

＜ Ｃ ＞ １次元超高分解能モデル
　どのメトリックでもいけるように作る



以上



Full NL

Ω : 1/10

パターン

case 毎に — Ω

① 毎 plot

1 realization

$\bar{u} = 0$

miter

1 white

3 peak

2 $\bar{u} = 0$

total energy

Budget 解析

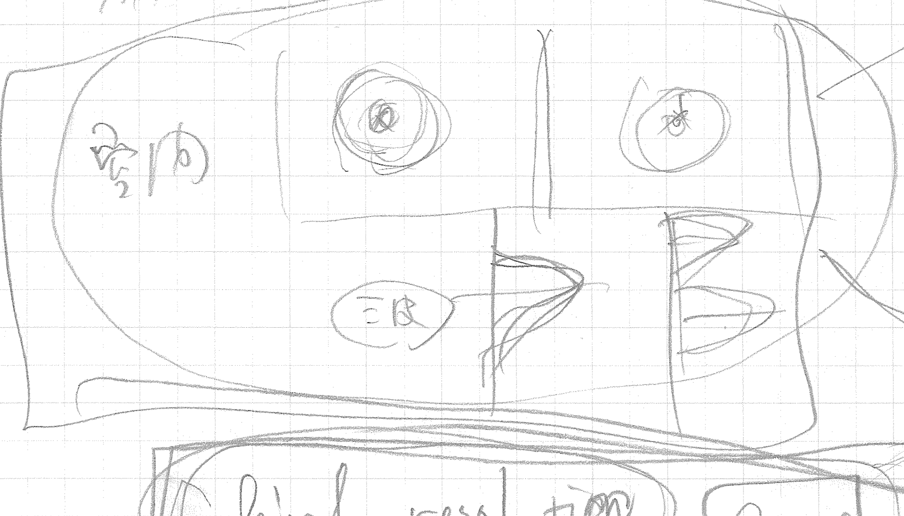
同じ初期値 で 違いそれよ.

weak NL

きれいなり

$J(x, x') + J(x', x'') + J(,)$

初



FL

拡張

NL

weak

High resolution

Case

$m = m_0$ 担当

Williams

2

Brotherton

R.

ΦANL num

1 case

$\int \overline{F}\, dt$

Ω — u'2?

$\int \overline{g' u'}\, dt$

Mail

87 JFM

$\sqrt{U/\beta 3}$

$\overline{q} = 0$

|k|

April 92

initia

渦度 s initia

ψ : zf → zp

ζ zf → zp

ψ × $n(n+1)$

- frt
- mslfrt など
- cfft.o

$\zeta = \nabla^2 \psi$

$= -n(n+1)\psi$

$\psi = -\frac{\zeta}{n(n+1)}$

- ~~initia3.f @ rossby/run~~
- rossby/run/initia3.f

⇓

run/run.f

& stupid08 で 作る

Ω istep ⇔ tint

omega2=

initial data

~/turbsp/rossby/grf/ginimp3.f.

ごうつ

rossby/run/initia3.f

計算 → file

→ /a6/yoden/rossby/initia/ r10 r11 r12

r60 r61 r62

0° 30° 60°

.frt

.ms|frt

.cftt.0

run

~/turbsp/run/run.f

計算 → file

/a6/yoden/rossby/run/r1[illegible]

grf 球面map

~/turbsp/grf/gmapt.f. ← fileをえらぶ

zmanom

~/ /grf/cwzint.f

gwzint.f

計算 ← file をえらぶ

FPT 1 real. →

φ (11) ?

- 実 —O(m, n)
- = ## (lon, lat) ←
- = (lat, m)

全部

FORMAT

自由

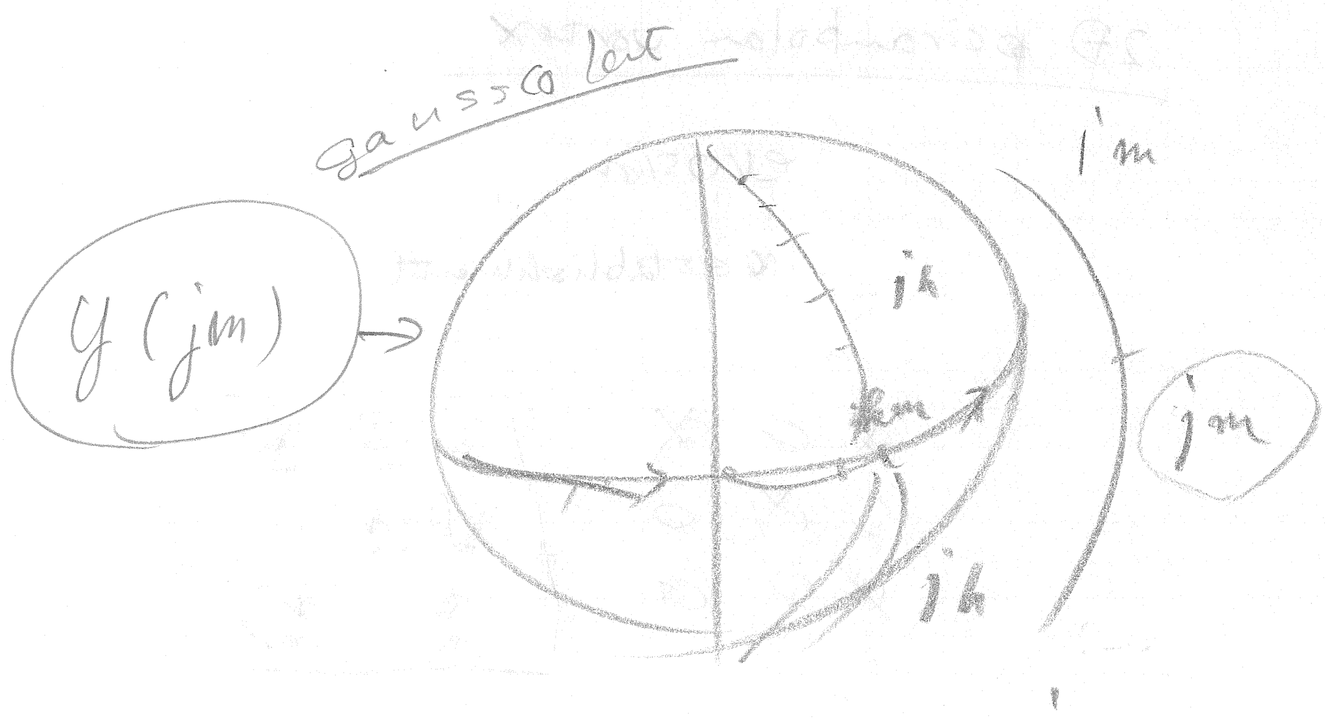
g' 2

A lat time

$\bar{u}\cos\phi$

weak nonlinear

FL



$x(j^{h}, 2)$

$1 \rightarrow \sin \phi_j$

$2 \rightarrow \cos \phi_j$

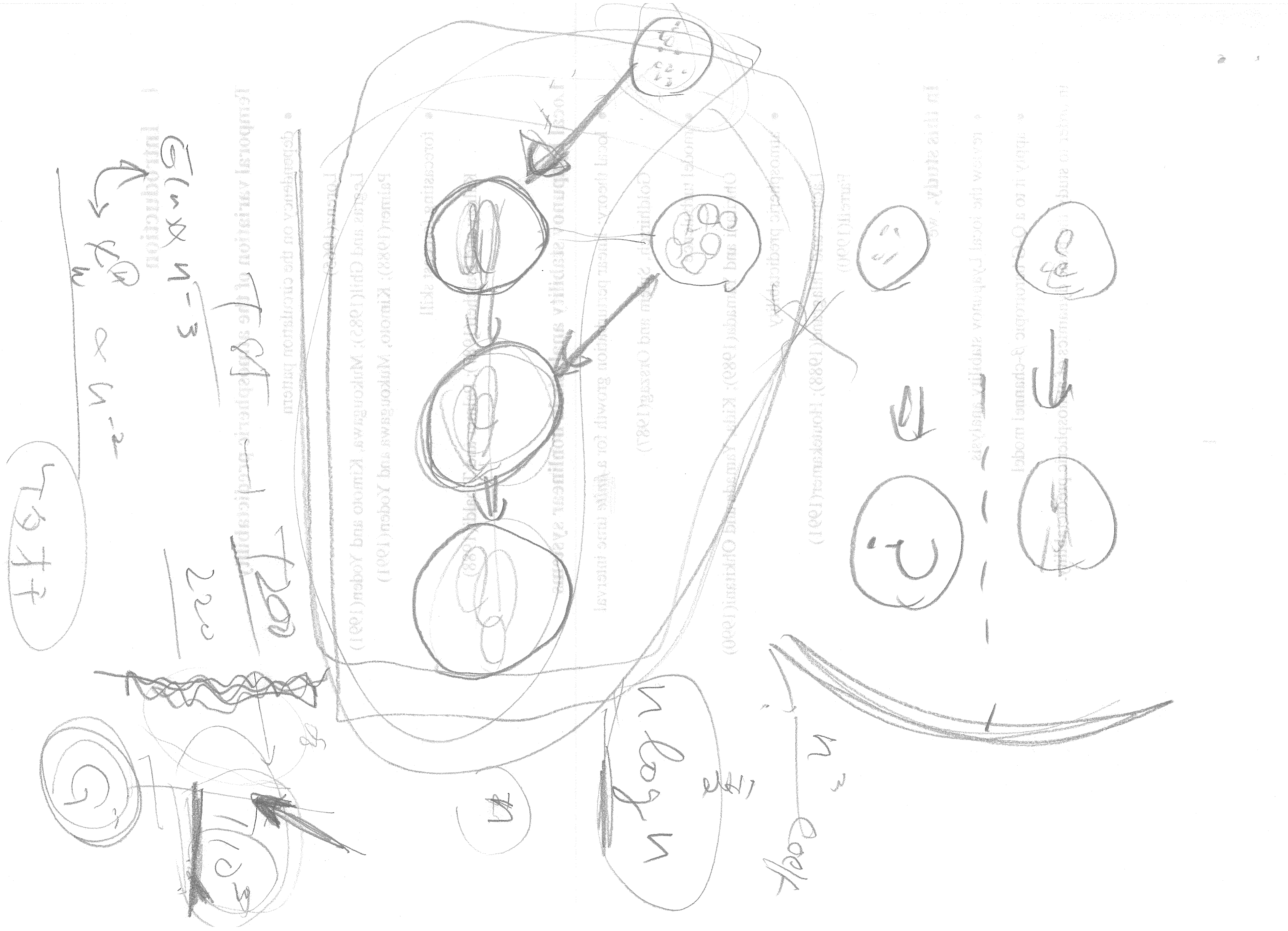
ϕ_j Glat.

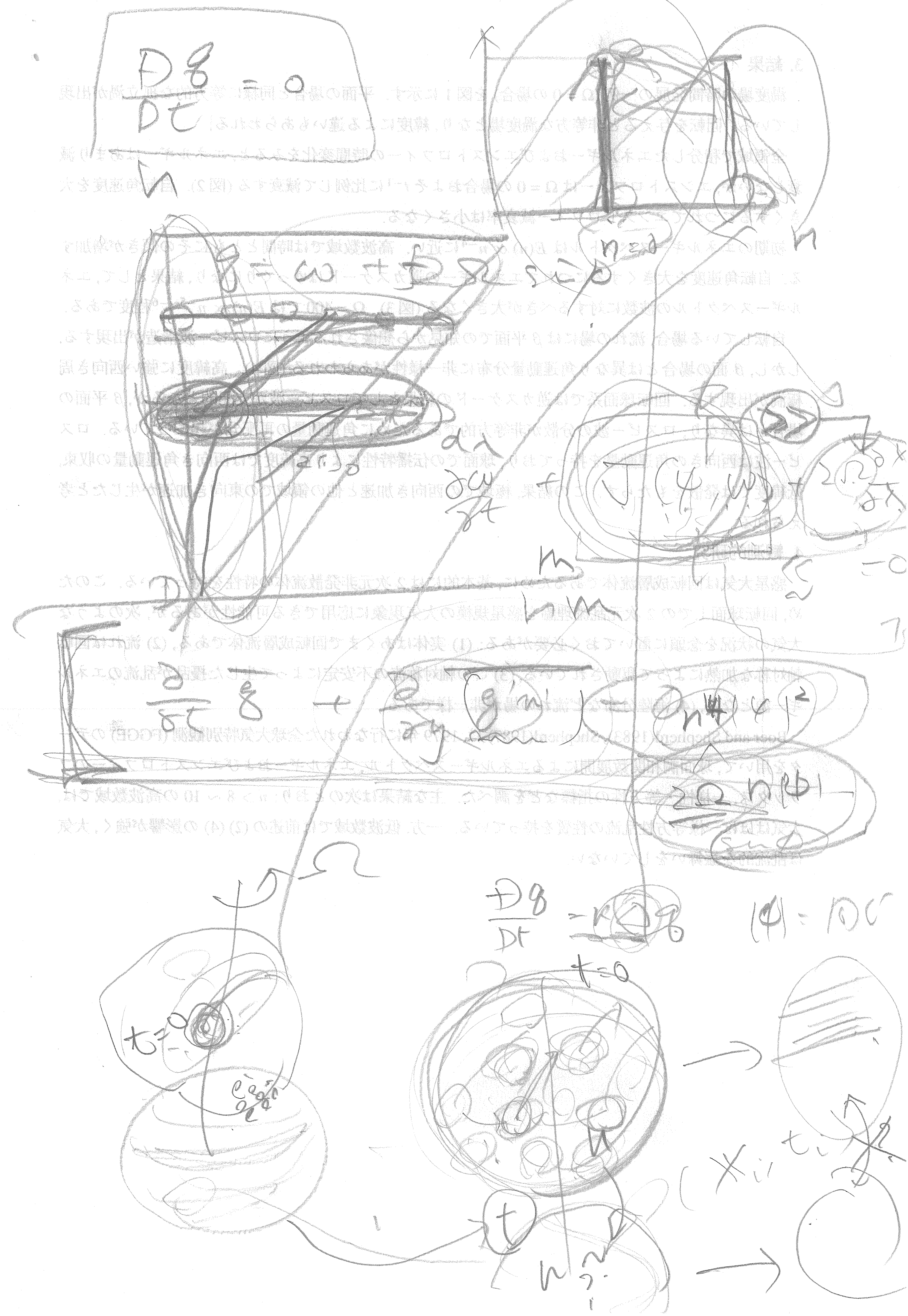
緯度

経 緯

$a(0{:}km, j^{m}_{j})$

0 — 360




Dq/Dt = 0
t=0
Dq/Dt
J(ψ,ω)
∂ω/∂t

中大気

NO

$\overline{u}(t, z)$

1970 − 90

$\frac{\partial \overline{u}}{\partial t} = \frac{\partial}{\partial z}(\overline{w_1}, \overline{w_2})$

$\overline{w}$ KB

$f\overline{w}_1(\overline{u})$

$f\overline{w}_2(\overline{u})$

RG

$\overline{u}(t, y, z)$

91〜

Bo

obs

88

宇宙

$$\overline{q} = \overline{\zeta} + f$$

$$\frac{2V(n\ 0)}{\zeta} + \frac{2\Omega \sin\phi}{2\Omega\ \mu}$$

$$\frac{\partial \overline{q}}{\partial \phi} = \sqrt{1-\mu^2}\left(\frac{\partial \overline{q}}{\partial \mu} + 2\Omega\right)$$: d$\overline{q}$bar

$$\overline{\zeta'^2} = \frac{1}{2\pi}\int_0^{2\pi} \zeta'^2 \, d\lambda$$

~~$= \frac{1}{2\pi}\int_0^{2\pi} \sum_{m=0}^{M} \zeta'(m) e^{im\lambda} \sum_{m=0}^{M} \zeta' e^{im\lambda} d\lambda$~~

$$= \frac{1}{2\pi}\int_0^{2\pi} \left[\sum_{m=0}^{n} \left(\zeta^m(t) e^{im\lambda} + \zeta^{-m}(t) e^{-im\lambda}\right)\right]^2 d\lambda$$

$$= \frac{1}{2\pi}\int_0^{2\pi}$$

$$\overline{A} = \left(\frac{\partial \overline{q}}{\partial \phi}\right)^{-1} \frac{\overline{\zeta'^2}}{2} \cos\phi \quad \boxed{9.10}$$

① data 解析

$A \propto \overline{q'^2}$

Lat

② NL ⊙T $\overline{u}\cos\phi$

A

$(u'v')\,dz$

```
C*****SUB*****CALCULATION OF INITIAL FIELDS
      SUBROUTINE INITF(ZT0,ZTM,P0,PM)
      PARAMETER(JMAX=32,MMAX=16)
      PARAMETER(JMPL1=JMAX+1,JMPL2=JMAX+2,MMPL1=MMAX+1)
      PARAMETER(JM2PL1=JMAX*2+1,MM2PL1=MMAX*2+1)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  ZT0(JMPL2),      P0(JMPL2)
      COMPLEX*16 ZTM(MMPL1,JMPL1),PM(MMPL1,JMPL1)
      DATA PI/3.1415926536D0/
      COMMON/CONST/RAD,UTIME,SCLP,SCLQ,SCLV
      COMMON/TRIG/CN(JM2PL1),SN(JM2PL1)
      COMMON/WIND/UBA0(JMPL1),U0(JMPL1)
C
C
      WRITE(6,'(//''      ***** INITIAL FIELD *****''/)')
C
      CLAT0 = 0.0/180.0*PI
      SLAT  = 15.0/180.0*PI
      ZINIT = 100. / SCLQ / 10000.
C             (ZINIT : nondimensional amplitude of vorticity)
      M0    = 1
      WRITE(6,'(1H ,10X,15HCENTER (LAT)   ,  F12.0)') CLAT0/PI*180.0
      WRITE(6,'(1H ,10X,15HHABA   (LAT)   ,  F12.0)') SLAT/PI*180.0
      WRITE(6,'(1H ,10X,15HINITIAL WAVE NO,  F12.0)') M0
      WRITE(6,'(1H ,10X,15HVORTICITY INT. ,1PE12.3)') ZINIT
C
      CALL ZETSET(ZT0,ZTM,ZINIT,CLAT0,SLAT,M0)
C
C    **CORRESPONDING STREAM FUNCTION
C
      CALL P0ZT0(P0,ZT0)
      CALL PMZTM(PM,ZTM)
C
      RETURN
      END
C
C
C
C*****SUB MOUNTAIN DEFINITION
      SUBROUTINE ZETSET(ZT0,ZTM,ZINIT,CLAT0,SLAT,M0)
      PARAMETER(JMAX=32,MMAX=16)
      PARAMETER(JMPL1=JMAX+1,JMPL2=JMAX+2,MMPL1=MMAX+1)
      PARAMETER(JM2PL1=JMAX*2+1,MM2PL1=MMAX*2+1)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMPLEX*16 ZTM(MMPL1,JMPL1)
      DIMENSION  ZT0(JMPL2)
      DIMENSION  AREAL(MMAX,JMPL1)
C
      COMPLEX*16 CTEM(MMAX)
C
      COMMON/TRIG/CN(JM2PL1),SN(JM2PL1)
      COMMON/GRID/DT,DLAT,FLAT0,NMAX
      COMMON/CONST/RAD,UTIME,SCLP,SCLQ,SCLV
      COMMON/PARM/NLNR,NWW,NUZI,NFORCE,NOUTLP,NOUTDI
      DATA PI/3.1415926536D0/
```

0
90
60

```
C
      WRITE(6,'(//''           ***** VORTICITY SETTING *****''/)')
C
C
C*****ZONAL MEAN COMPONENT
C
      DO 20 J = 1,JMAX+2
            ZT0(J)= 0.0
   20 CONTINUE
C
C*****WAVY FORCING
C                (PLACED AT J)
      DO 50 J = 1,JMAX+1
            FLAT  = DLAT*FLOAT(J-1)+FLAT0
            DO 30 I = 1,MMAX
                  FLONG = 2.0*PI*FLOAT(I-1)/FLOAT(MMAX)
                  CTEM(I)  = ZINIT*EXP(-((FLAT-CLAT0)/SLAT)**2)
     &                         * SIN(FLOAT(M0)  * FLONG)
C
   30       CONTINUE
C
            CALL FFT(CTEM,MMAX,-1.0)
            CTEM(MMAX/2+1)  = CTEM(MMAX/2+1)/2.0
C
            DO 40 M = 1,MMAX/2
                  ZTM(MMAX/2-M+1,J)
     $                  = CONJG( CTEM(M+1)/SQRT(FLOAT(MMAX))  )
                  ZTM(MMAX/2+M+1,J)
     $                  = CTEM(M+1)/SQRT(FLOAT(MMAX))
   40       CONTINUE
            ZTM(MMAX/2+1,J)  = 0.0
   50 CONTINUE
      DO 60 M = 1,MMAX+1
            ZTM(M,1)  = 0.0
            ZTM(M,JMAX+1)  = 0.0
   60 CONTINUE
C
C
      WRITE(6,'(/'' INITIAL ZONAL VORTICITY'')')
      CALL WRTZ(ZT0,JMAX+2,SCLQ,-0.5)
      WRITE(6,'(/'' INITIAL WAVE  VORITCITY'')')
      CALL REWA(AREAL,ZTM)
      CALL WRTLL(AREAL,SCLQ)
C
C
      RETURN
      END
```

```
************************************************************************
*       TIME-INTEGRATION OF                                            *
*       A SPECTRAL BAROTROPIC MODEL IN A SPHERICAL DOMAIN              *
*                                                                      *
*       < version 0 >                                                  *
*         for VP-400 / VP-2600                                         *
*             + Temperton's FFT                                        *
*                                                                      *
*       Shigeo Yoden (Dept. of Geophysics, Kyoto Univ.)                *
*       May XX, 1990 / JULY 1, 1991                                    *
************************************************************************

      PROGRAM FRUNVP

      IMPLICIT    REAL* 8(A-H,O-Y)
      IMPLICIT COMPLEX*16(Z)

* transform resolution parameters
      INCLUDE (FPARA)
      PARAMETER ( DUMMY = 999.9 )

* parameters
* grid resolution
*                  NLON = number of longitudes (must be an even number)
*                  NLAT = number of latitudes
*                  NLOH = NLON/2
*                  NLAH = number of latitudes between pole and equator
* spectral resolution
*                  MT   = maximum wavenumber for triangle truncation
*                  MCOE = total number of spectral components
* time-integrations
*                  IMAX  = number of time integrations
*                  IMAXE = number of sampling points for TENG
*                  IMAXP = number of sampling points for ZP and SENG
*                  DT    = a time increment
*+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

      DIMENSION ZP(0:MT,0:MT)
      REAL      TENG(0:IMAXE), SENG(MT,0:IMAXP)
      REAL      TENS(0:IMAXE), SENS(MT,0:IMAXP)

*            ZP     : spectral coefficients of stream functions
*            TENG   : total energy to be saved
*            TENS   : total enstrophy to be saved
*            SENG   : energy spectra to be saved
*            SENS   : enstrophy spectra to be saved

      DIMENSION JMIN(0:MT), JMAX(0:MT)

*            JMIN, JMAX : index arrays for one-dimensionalization

      DIMENSION COLAT(NLAT), GW(NLAT)
      DIMENSION PMN(NLAT,0:MT,0:MT), DPMN(NLAT,0:MT,0:MT)

*            COLAT  : colatitude(deg) of the Gaussian grids
*            GW     : Gaussian weight
*            PMN    : normalized Legendre functions
*            DPMN   : d(PMN)/d(COSCLT)

      DIMENSION IFCTR(2:5)
      DIMENSION TC(NLOH), TS(NLOH), TC2(NLOH), TS2(NLOH)

*            IFCTR    : factor of 2, 3, 4, and 5 for Temperton's FFT
*            TC , TS  : cosine and sine values for FFT
*            TC2, TS2 : cosine and sine values for FFT
```

```
      COMMON /GLINDX/ JMIN, JMAX
      COMMON /GLLATS/ COLAT, GW
      COMMON /GLFUNC/ PMN, DPMN
      COMMON /GLTRIG/ IFCTR, TC, TS, TC2, TS2
      COMMON /CONSTS/ OME, RNU4

      CHARACTER CRDAT1*50, CRDAT2*50, CWDAT1*50, CWDAT2*50

      DATA CRDAT1/'X57071.ZCOEF.DATA(T85F)'/
      DATA CRDAT2/'X57071.AINIT.DATA(A??)'/
      DATA CWDAT1/'X57071.AZ###.DATA(A??)'/
      DATA CWDAT2/'X57071.AE###.DATA(A??)'/
*-----------------------------------------------------------------------

* initialization of index arrays for 1-dim. data
      CALL ININDX

* initialization of Legendre functions
      CALL INILGD( CRDAT1 )

* initialization of FFT
      CALL INIFFT
*         also check IFCTRs in INIFFT

      DO 10 III=1,3
* give some experimental constants
        IJK  = 50 * 2**III
        OME  = 1.0D0 * IJK
        RNU4 = 1.0D-6

        WRITE(CWDAT1(10:12),'(I3.3)') IJK
        WRITE(CWDAT2(10:12),'(I3.3)') IJK

        DO 20 I=1,8
        DO 20 J=1,6

          WRITE(CRDAT2(20:20),'(I1)') I
          WRITE(CWDAT1(20:20),'(I1)') I
          WRITE(CWDAT2(20:20),'(I1)') I
          WRITE(CRDAT2(21:21),'(I1)') J
          WRITE(CWDAT1(21:21),'(I1)') J
          WRITE(CWDAT2(21:21),'(I1)') J

* read initial conditions
          CALL INITIA( CRDAT2, ZP )

          OPEN (11, FILE=CWDAT1, ACTION='WRITE', STATUS='SHR')
          WRITE(11) IMAX, IMAXP, DT, OME, RNU4

* time integrations
          CALL INTEGR( ZP, TENG, TENS, SENG, SENS )
          CLOSE(11)

* output
          CALL OUTEE( CWDAT2, TENG, TENS, SENG, SENS )

 20   CONTINUE
 10 CONTINUE

      STOP
      END
************************************************************************
      SUBROUTINE ININDX

*  ININDX initializes index arrays for 1-dim. data
*-----------------------------------------------------------------------
```

```
      IMPLICIT    REAL* 8(A-H,O-Y)
      IMPLICIT COMPLEX*16(Z)

* transform resolution parameters
      INCLUDE (FPARA)
      PARAMETER ( DUMMY = 999.9 )

      DIMENSION JMIN(0:MT), JMAX(0:MT)

      COMMON /GLINDX/ JMIN, JMAX
*-----------------------------------------------------------------------
* set the index arrays
      JMIN(0) = 1
      JMAX(0) = 1 + MT

      DO 10 N=1,MT
        JMIN(N) = JMAX(N-1) + 1
        JMAX(N) = JMIN(N)   + MT - N
   10 CONTINUE

      RETURN
      END
************************************************************************
      SUBROUTINE INILGD( CRDAT1 )

*  INILGD initializes Legendre functions at Gaussian latitudes
*
*  On entry:
*     CRDAT1 - data set name to be read
*-----------------------------------------------------------------------

      IMPLICIT    REAL* 8(A-H,O-Y)
      IMPLICIT COMPLEX*16(Z)

* transform resolution parameters
      INCLUDE (FPARA)
      PARAMETER ( DUMMY = 999.9 )

      DIMENSION COLAT(NLAT), GW(NLAT)
      DIMENSION PMN(NLAT,0:MT,0:MT), DPMN(NLAT,0:MT,0:MT)

      CHARACTER CRDAT1*50

      COMMON /GLLATS/ COLAT, GW
      COMMON /GLFUNC/ PMN, DPMN
*-----------------------------------------------------------------------

* read Gaussian weights and Legendre functions
      OPEN (1, FILE=CRDAT1, ACTION='READ', STATUS='SHR')
      READ (1) NMT, NNLAT, COLAT, GW, PMN, DPMN
      CLOSE(1)

      IF(NMT.NE.MT .OR. NNLAT.NE.NLAT) STOP 'ZCOEF'

      RETURN
      END
************************************************************************
      SUBROUTINE INITIA( CRDAT2, ZP )

*  INITIA gives an initial condition.
*
*  On entry:
*     CRDAT2 - data set name of the initial condition
*  On exit:
*     ZP     - spectral coefficients with initial values
```

```
*-----------------------------------------------------------------------

      IMPLICIT    REAL* 8(A-H,O-Y)
      IMPLICIT COMPLEX*16(Z)

* transform resolution parameters
      INCLUDE (FPARA)
      PARAMETER ( DUMMY = 999.9 )

      DIMENSION ZP(0:MT,0:MT)
      CHARACTER CRDAT2*50, CT*5
*-----------------------------------------------------------------------
* set initial values
        OPEN (2, FILE=CRDAT2, ACTION='READ', STATUS='SHR')
        READ (2) CT, NMT, NNLAT, ZP
        CLOSE(2)

      IF(NMT.NE.MT .OR. NNLAT.NE.NLAT) STOP 'AINIT'

      RETURN
      END
************************************************************************
      SUBROUTINE INTEGR( ZP0, TENG, TENS, SENG, SENS )

*  Time integration by Runge-Kutta scheme
*  On entry:
*      ZP0  - spectral coefficients with initial values
*  On exit:
*      TENG - total energy to be saved
*      TENS - total enstrophy to be saved
*      SENG - energy spectra
*      SENS - enstrophy spectra
*-----------------------------------------------------------------------

      IMPLICIT    REAL* 8(A-H,O-Y)
      IMPLICIT COMPLEX*16(Z)

* transform resolution parameters
      INCLUDE (FPARA)
      PARAMETER ( DUMMY = 999.9 )

      REAL      TENG(0:IMAXE), SENG(MT,0:IMAXP)
      REAL      TENS(0:IMAXE), SENS(MT,0:IMAXP)
      DIMENSION ZP0(0:MT,0:MT), ZK1(0:MT,0:MT), ZK2(0:MT,0:MT),
     +          ZK3(0:MT,0:MT), ZK4(0:MT,0:MT), ZWK(0:MT,0:MT)
*-----------------------------------------------------------------------

      ISTPE = IMAX / IMAXE
      ISTPP = IMAX / IMAXP

      IE = 0
      IP = 0

* time-integration loop-------------------------------------------------
      DO 10 I=0,IMAX
* storage of energy and enstrophy for each ISTPE steps
        IF(MOD(I,ISTPE) .EQ. 0) THEN
          CALL ENGENS( IE, ZP0, TENG, TENS )
          IE = IE + 1
        END IF

* storage of ZP, energy and enstrophy spectra for each ISTPP steps
        IF(MOD(I,ISTPP) .EQ. 0) THEN
          CALL SEEZP( I, IP, ZP0, SENG, SENS )
          IP = IP + 1
        END IF
```

```
        IF(IE.GT.IMAXE .OR. IP.GT.IMAXP) GO TO 999

* computation of tendency
        CALL DELTA( ZP0, ZK1 )
        DO 21 M=0,MT
        DO 21 N=M,MT
          ZK1(N,M) = DT*ZK1(N,M)
          ZWK(N,M) = ZP0(N,M) + 0.5D0*ZK1(N,M)
   21   CONTINUE

        CALL DELTA( ZWK, ZK2 )
        DO 22 M=0,MT
        DO 22 N=M,MT
          ZK2(N,M) = DT*ZK2(N,M)
          ZWK(N,M) = ZP0(N,M) + 0.5D0*ZK2(N,M)
   22   CONTINUE

        CALL DELTA( ZWK, ZK3 )
        DO 23 M=0,MT
        DO 23 N=M,MT
          ZK3(N,M) = DT*ZK3(N,M)
          ZWK(N,M) = ZP0(N,M) + ZK3(N,M)
   23   CONTINUE

        CALL DELTA( ZWK, ZK4 )
        DO 24 M=0,MT
        DO 24 N=M,MT
          ZK4(N,M) = DT*ZK4(N,M)
   24   CONTINUE

* one step foreward
        DO 30 M=0,MT
        DO 30 N=M,MT
          ZP0(N,M) = ZP0(N,M)
     +    + ( ZK1(N,M) + 2.D0*(ZK2(N,M) + ZK3(N,M)) + ZK4(N,M) )/6.D0
   30   CONTINUE

*check overflow
          SUM = 0.D0
          DO 40 M=0,MT
          DO 40 N=M,MT
            SUM = SUM + ABS( ZP0(N,M) )
   40     CONTINUE
          IF(SUM .GE. 1.D1) GO TO 999

   10 CONTINUE
*-----------------------------------------------------------------------
  999 CONTINUE

      RETURN
      END
***********************************************************************
***********************************************************************
      SUBROUTINE SEEZP( I, IP, ZP, SENG, SENS )
*
*  ENGENS computes energy and enstrophy.
*
*  On entry:
*     I    - step #
*     IP   - step # for energy spectra calculation
*     ZP   - spectral coefficients
*  On exit:
*     SENG - energy spectra
*     SENS - enstrophy spectra
*-----------------------------------------------------------------------
```

```
      IMPLICIT    REAL* 8(A-H,O-Y)
      IMPLICIT COMPLEX*16(Z)

* transform resolution parameters
      INCLUDE (FPARA)
      PARAMETER ( DUMMY = 999.9 )

      DIMENSION ZP(0:MT,0:MT)
      DIMENSION JMIN(0:MT), JMAX(0:MT)
      DIMENSION  ENG(MT),         ENS(MT)
      REAL      SENG(MT,0:IMAXP), SENS(MT,0:IMAXP)
      COMPLEX   ZOUT(MCOE)

      COMMON /GLINDX/ JMIN, JMAX
      COMMON /CONSTS/ OME, RNU4
*-----------------------------------------------------------------------

* compute energy and enstrophy spectrum
      DO 10 N=1,MT
        ENG(N)  = 0.5D0*N*(N+1)*ABS(ZP(N,0))**2
        ENS(N)  = 0.5D0*N*(N+1)*ABS(ZP(N,0))**2   *N*(N+1)
      DO 10 M=1,N
        ENG(N)  = ENG(N)  + N*(N+1)*ABS(ZP(N,M))**2
        ENS(N)  = ENS(N)  + N*(N+1)*ABS(ZP(N,M))**2 *N*(N+1)
   10 CONTINUE

      DO 20 N=1,MT
        SENG(N,IP)  = ENG(N)
        SENS(N,IP)  = ENS(N)
   20 CONTINUE

* store in one-dimensional array
      DO 30 L=0,MT
        M = -1
        DO 40 J=JMIN(L),JMAX(L)
          M = M + 1
          N = M + L
          N = M + L
          ZOUT(J)  = ZP(N,M)
   40   CONTINUE
   30 CONTINUE

      WRITE(11) I, ZOUT

      RETURN
      END
***********************************************************************
      SUBROUTINE OUTEE( CWDAT2, TENG, TENS, SENG, SENS )

***********************************************************************
      SUBROUTINE DELTA( ZP, ZFUN )

*  DELTA computes the time derivative of each dependent variables.
*
*  On entry:
*     ZP   - spectral coefficients
*  On exit:
*     ZFUN - time derivatives for each components
*-----------------------------------------------------------------------

      IMPLICIT    REAL* 8(A-H,O-Y)
      IMPLICIT COMPLEX*16(Z)

* transform resolution parameters
      INCLUDE (FPARA)
```

```
      PARAMETER ( DUMMY = 999.9 )

      DIMENSION ZP(0:MT,0:MT), ZFUN(0:MT,0:MT),
     +          ZV(0:MT,0:MT), ZJAC(0:MT,0:MT)

      DIMENSION GIP(NLAT,NLON), GJP(NLAT,NLON),
     +          GIV(NLAT,NLON), GJV(NLAT,NLON),
     +          GJA(NLAT,NLON)

      COMMON /CONSTS/ OME, RNU4
*-----------------------------------------------------------------------

      ZIX2 = (0.D0,2.D0)

* compute the vorticity
      CALL LGPSTV( ZP, ZV )

* compute gradients in grids
      CALL LGGSTG( ZP, GIP, GJP )
      CALL LGGSTG( ZV, GIV, GJV )

* compute nonlinear terms in grids
      CALL LGGNON( GIP, GJP, GIV, GJV, GJA )

* convert the nonlinear terms into spectral coeff.
      CALL LGGTS( GJA, ZJAC )

* compute ZFUN for the derivatives
      DO 10 M=0,MT
      DO 10 N=M,MT
        ZFUN(N,M) = ZJAC(N,M)
     +            - ZIX2*OME*M*ZP(N,M)
--     +            - ZIX2*OME*M*ZP(N,M)
     +            - RNU4*N*N**(N+1)*(N+1)*ZV(N,M)
   10 CONTINUE

* special section for m=0
      ZFUN(0,0) = (0.D0,0.D0)
      DO 20 N=1,MT
        ZFUN(N,0) = - ZFUN(N,0) / ( N*(N+1) )
   20 CONTINUE

* all other m and n
      DO 30 M=1,MT
      DO 30 N=M,MT
        ZFUN(N,M) = - ZFUN(N,M) / ( N*(N+1) )
   30 CONTINUE

      RETURN
      END
***********************************************************************
      SUBROUTINE LGPSTV( ZP, ZV )
```

ZM