WHOI-84-24

**Models and Simulations of Tracer Dispersal
by Mid-Ocean Mesoscale Eddies**

by

Dale B. Haidvogel and Thomas Keffer

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543

May 1984

Technical Report

*Funding was provided by the National Science Foundation under
Grant No. OCE81-09486, and the Center for Analysis
of Marine Systems (CAMS).*

*Reproduction in whole or in part is permitted for any purpose
of the United States Government.*

*This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept.
WHOI-84-24.*

Approved for public release; distribution unlimited.

Approved for Distribution:

Nick P. Fofonoff, Chairman
Department of Physical Oceanography

TABLE OF CONTENTS

	Page No.
ABSTRACT	2
1. Introduction	3
2. Model equations, parameters and solution techniques	4
3. Tracer dispersal	11
4. Acknowledgments	13
TABLE 1	14
REFERENCES	15
APPENDIX A	16
APPENDIX B	29
APPENDIX C	42
APPENDIX D	49

ABSTRACT

Haidvogel and Keffer (1984) have examined the stirring and mixing of a passive tracer in a turbulent mesoscale eddy field for environmental parameters characteristic of a homogeneous mid-ocean region. This study utilized two numerical models, based respectively on the barotropic vorticity and advective-diffusive equations, to simulate the evolving mid-ocean eddy field and the associated dispersion of an embedded tracer spot. Here, these models are described in detail, and the full suite of tracer release simulations are shown in the form of contour plots.

1. Introduction

Haidvogel and Keffer (1984; hereafter HK) have examined the stirring and mixing of a passive scalar in two-dimensional β -plane turbulence as a prototype for tracer dispersal within a single isopycnal layer in the mid-ocean. Their modeling strategy is as follows. Specific realizations of a dynamically evolving mid-ocean streamfunction (velocity) field are generated by direct integration of the forced/damped barotropic vorticity equation over a spatially periodic domain, and for representative mid-ocean values of environmental parameters such as β and RMS velocity. An initial distribution of a dynamically passive tracer is then introduced within the mid-ocean domain. The source is a localized Gaussian "spot", instantaneously released. The subsequent evolution of the tracer from its initial state is determined by integration of the advective-diffusive equation, using the simulated velocity time series and an assumed parameterization for the subgrid-scale diffusive processes. In this report, we describe the models used, and show the complete sequence of tracer dispersal simulations used in their statistical analysis. Readers are referred to the published manuscript (Haidvogel and Keffer, 1984) for a complete scientific discussion of these results.

2. Model equations, parameters and solution techniques

Consider a homogeneous, two-dimensional fluid on a β -plane contained within a domain of dimensional size $(2\pi L)$ and characterized by an RMS fluid speed of U . Adopting L and U as dimensional scales, set

$$[x, y, t, \psi] \sim [L, L, (L/U), LU].$$

In the resulting non-dimensional system, the domain is of length (2π) units, the RMS fluid speed is $O(1)$, and the forced, dissipative vorticity equation may be written

$$\frac{D}{Dt} q = \left[\frac{\partial}{\partial t} + J(\psi, \cdot) \right] q = D + F \quad (1)$$

where the potential vorticity

$$q = \nabla^2 \psi + \hat{\beta} y = \zeta + \hat{\beta} y, \quad (2)$$

and the streamfunction ψ is related to the horizontally non-divergent velocity by the diagnostic relation

$$\hat{\nabla} = (u, v) = \hat{k} \times \nabla \psi = (-\partial \psi / \partial y, \partial \psi / \partial x).$$

Here, $\zeta = \nabla^2 \psi$ is the relative vorticity, and all parameters with a "hat" (e.g., $\hat{\beta}$) are non-dimensional. In Eq. (1), the term D denotes dissipative terms of the form:

$$D = -\hat{\lambda}\zeta + \hat{v}\nabla^2\zeta. \quad (3)$$

As has become common in ocean modeling, dissipative effects are parameterized in terms of linear ("bottom friction") and harmonic ("lateral mixing") terms.

In order to ensure a statistical equilibrium in which area-mean energy $[E=(8\pi^2)^{-1}\int_0^{2\pi}\int_0^{2\pi}|\nabla\psi|^2 dx dy]$ and relative enstrophy $[Z=(8\pi^2)^{-1}\int_0^{2\pi}\int_0^{2\pi}\zeta^2 dx dy]$ are approximately constant in time, the vorticity equation must include forcing as well as dissipative terms. In the simulations described here, $F(x,y,t)$ takes on a simple (but highly idealized) form. To understand it, consider representing $\psi(x,y,t)$ in terms of a double Fourier series with Fourier coefficients $\psi_{kl}(t)$, where k and l are the x and y wavenumbers respectively. Similarly, equation (1) may be transformed to Fourier space, where it then expresses the time rate of change of specific Fourier components of the vorticity field. Letting $\kappa = (k^2 + l^2)^{1/2}$ be the total (circular) wavenumber, the forcing $F_{kl}(t)$ as a function of horizontal wavenumber is

$$F_{kl}(t) = \begin{cases} -\hat{\tau}\psi_{kl}(t)/|\psi_{kl}(t)|^2 & 6 \leq \kappa \leq 8 \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

The forcing is therefore being applied only in a narrow wavenumber band centered on $\kappa = 7$.

An important property of F is noted by multiplying equation (1) by $-\psi_{kl}^*$ (i.e., $-\psi_{kl}$ conjugate) to form the associated kinetic energy equation. The result is an energy tendency due to the forcing given by:

$$\frac{\partial}{\partial t} E_{kl}(t) = \hat{\tau}, \quad 6 \leq k \leq 8 \quad (5)$$

That is, the forcing function has been chosen to inject kinetic energy at a uniform rate into the narrow wavenumber band $6 \leq k \leq 8$. (Because of the narrowness of the forced spectral band, it is easily seen that enstrophy is nearly uniformly injected.) The result is a smooth excitation of the fluid, and a smooth variation of E and Z within the flow field.

The physical non-dimensional parameters describing the vorticity equation are then:

$$\begin{aligned} \hat{\beta} &= \beta L^2 / U \\ \hat{\lambda} &= \lambda L / U \\ \hat{\nu} &= \nu / UL \\ \text{and} \quad \hat{\tau} &= \tau L / U^3, \end{aligned} \quad (6)$$

and are associated respectively with the effects of β , bottom and lateral dissipative processes, and the applied forcing. In (6), β , λ , ν and τ are the dimensional values associated with these processes. (Note that τ has the units of a rate of energy injection.)

Given initial conditions and a set of values for the parameters (6), the vorticity equation (1) can be advanced in time to yield a time-dependent β -plane streamfunction field. Having computed $\psi(x,y,t)$, an evolving passive tracer field $\theta(x,y,t)$ can be then computed from a suitable initial state using the advective-diffusive equation:

$$\frac{D\theta}{Dt} = \left[\frac{\partial}{\partial t} + J(\psi, \theta) \right] \theta = \hat{\eta} \nabla^2 \theta. \quad (7)$$

Non-dimensionalized as above, equation (7) has only one physical parameter, the subgrid-scale or explicit diffusivity

$$\hat{\eta} = \eta/UL. \quad (8)$$

In the absence of turbulent advection, the explicit diffusivity $\hat{\eta}$ controls the rate at which tracer dispersal occurs. However, the presence of a turbulent flow field greatly accelerates the mixing process.

By assumption, $\psi(x,y,t)$ is periodic in each horizontal direction within a square domain $[-\pi, \pi] \times [-\pi, \pi]$. Consistent with the assumed spatial homogeneity, it is both convenient and appropriate to expand the streamfunction in a double Fourier series, and to apply the pseudospectral solution technique (Orszag and Israeli, 1974). A more detailed description of the technique, and some of the prior results obtained with it, have been reviewed by Haidvogel (1983).

To begin, let

$$\psi(x,y,t) = \sum_{k=-N/2}^{(N/2)-1} \sum_{l=-N/2}^{(N/2)-1} \psi_{kl}(t) e^{i(kx + ly)} \quad (9)$$

where the complex Fourier series for ψ is truncated at the cutoff wave-number $(N/2)$ in each direction. An identical Fourier series representation is used for the spectral coefficients $\theta_{kl}(t)$ of the tracer field. In the experiments which follow $N = 128$, corresponding to a

physical space grid of 128×128 points, and a truncation wavenumber $(N/2) = 64$.

The spectral coefficients $\psi_{kl}(t)$ and $\theta_{kl}(t)$ are advanced in time according to equations (1) and (7), written in their equivalent Fourier space form. Leapfrog time-differencing is used, with semi-implicit treatment of the dissipative terms to avoid unconditional computational instability. The time step chosen, $\Delta t = 0.005$ in non-dimensional units, was stable in all cases. The Jacobian terms $J(\psi, q)$ and $J(\psi, \theta)$, evaluated using the pseudospectral procedure, conserve energy (E), square vorticity (enstrophy, Z), and tracer variance (S) in the absence of dissipation, forcing and time-differencing error. The Fourier integration scheme also identically preserves the total tracer inventory ($\iint \theta dx dy$). In addition, the computed non-linear $[J(\psi, q)]$ and quadratic $[J(\psi, \theta)]$ product terms are alias-free. A variety of analytically tractable test problems—including pure diffusion of a Gaussian spot, and dispersal within a large-scale shear field—were run, showing the numerical codes to be accurate to within accountable machine and time truncation error. Listings of the vorticity and advective-diffusive models, and the Fast Fourier transform, are given in Appendices A-C.

HK examine two specific velocity space/time series. These velocity fields, labeled V01 and V02 respectively, result from equation (1) with the choice of parameter values listed in Table 1. Note that the two velocity fields differ only in the value of $\hat{\beta}$. While V01 represents pure two-dimensional turbulence ($\hat{\beta} = 0$), V02 corresponds to a value of

$\hat{\beta}$ appropriate to a quiet mid-ocean region at mid-latitudes ($\hat{\beta}=10$). The values of the frictional parameters, particularly $\hat{\nu}$ and $\hat{\eta}$, were chosen to be as small as computationally appropriate; further reduction in these parameters led to a rapid evolution to the thermodynamic equipartition state (Bennett and Haidvogel, 1983).

To obtain velocity fields V01 and V02, the vorticity equation was integrated in each case from an initially isotropic energy spectrum of nearly constant amplitude for $1 \leq \kappa \leq 8$, and decreasing as κ^{-5} for larger κ . The initial phases of the complex streamfunction amplitudes $\psi_{k1}(0)$ were supplied by a random number generator. The particular value of the forcing parameter of $\hat{\tau}$ was chosen, after some experimentation, to achieve an approximate statistical equilibrium with the frictional effects in which energy is nearly constant at a value of about 0.7. From the initial conditions, the vorticity equation was integrated for approximately 20 time units. After this initial period, the V01 and V02 streamfunction fields were stored at each time step for later access by the advective-diffusive model. HK discuss in detail the character of the resulting turbulent velocity field.

The non-dimensional values of the parameters ($\hat{\beta}, \hat{\lambda}, \hat{\nu}, \hat{\tau}$) which have been used to generate the V01 and V02 velocity space/time series could have arisen from any of a number of choices for the length and velocity scales L and U. This is particularly true since, of all the dimensional parameters in the problem, only β has a well-defined value (here, taken to be $2 \times 10^{-13} \text{ cm}^{-1} \text{ sec}^{-1}$). Therefore, the interpretation of these numerical results in terms of dimensional units is not unambiguous.

HK chose a consistent estimate for the dimensional length scale L by requiring the simulated eddy length scale, as evidenced by the Lagrangian transverse velocity correlation function to equal that observed in the Western North Atlantic (50 km; Freeland, Rhines and Rossby, 1975). To establish this connection, particle trajectories were derived from the V01 and V02 fields by appropriate interpolation techniques. [The details of these techniques, and further discussion of particle motions on a β -plane, is discussed elsewhere (Haidvogel, 1984).] As noted by HK, the desired compatibility of observed and simulated Lagrangian length scales is achieved by choosing L to be 100km, yielding a characteristic eddy length scale comparable to that observed. The dimensional domain size is thus $(2\pi L)$ or 628km on a side, and the grid spacing $(2\pi L/N)$ is 4.9km.

Having set L , and knowing β , the velocity scale U may be consistently determined from the relation $U = \beta L^2 / \hat{\beta}$. For V02 ($\hat{\beta}=10$), the velocity scale is 2 cm/sec, a value crudely representative of quiet mid-ocean regions. With $U = 2$ cm/sec, the dimensional (advective) time scale can be computed as $(L/U) = 5 \times 10^6$ sec $\approx$ 58 days. In what follows, one non-dimensional time unit therefore corresponds to approximately two months real time for $\hat{\beta} = 10$. With fixed values for L and U , the equivalent dimensional value for the dissipative and forcing parameters may also be established. These are given in Table 1 for the V02 simulation.

3. Tracer dispersal

To examine tracer dispersal in these mesoscale eddy fields, a narrow Gaussian spot -- $\theta(x,y,t_0) = \exp\{[(x-x_0)^2 + (y-y_0)^2]/2\sigma^2\}$ -- of an (assumed passive) tracer is released at a given space-time location (x_0, y_0, t_0) within the evolving flow field. The subsequent distribution of tracer is obtained by integration of the advective/diffusive equation (7) using an explicit (subgridscale) diffusivity $\hat{\eta} = 0.005$ (dimensionally $\eta = 10^5 \text{ cm}^2 \text{ sec}^{-1}$). Initially, the tracer spot size (2σ) equals about $3 \Delta x$ (17 km). A narrow spot was chosen to maximize the scale separation between the tracer field and the energy-containing eddies.

Each tracer dispersal realization was integrated for 800 time steps (four non-dimensional time units). In many cases, as shown below, the elapsed time was sufficient for the tracer to have dispersed over a considerable fraction of the doubly periodic domain. Not counting the computational time initially necessary to construct the velocity field, each tracer realization required approximately six hours of CPU time on the Woods Hole DEC VAX 11/780 system.

The character and detail of these tracer dispersal simulations are functionally dependent to some degree on all the parameter values chosen and model assumptions made in generating the velocity fields, and in integrating the advective-diffusive equation. Of the parameters involved, the one most likely to be important, and fortunately the one with the most well-defined value, is $\hat{\beta}$. For evaluation of the

effects of $\hat{\beta}$, ensembles of 10 realizations each were carried out in both the V01 and V02 velocity fields. In comparison to V01, V02 has essentially identical total energy, enstrophy and RMS strain, as well as energy spectral shape for $\kappa \geq 3$. Characteristic flow quantities such as RMS velocity and eddy scale are the same in both cases.

Appendix D contains figures which show the time evolution of these tracer spots in each of the ten V01 and V02 simulations. The simulations are numbered V01H01-V01H10 ($\hat{\beta} = 0$) and V02H01-V02H10 ($\hat{\beta} = 10$).

4. Acknowledgments

13

This work has been funded by the National Science Foundation through Grant No. OCE81-09486 to the Woods Hole Oceanographic Institution, and by the Center for Analysis of Marine Systems (CAMS).

Parameter	Units	Value	Reference
Salinity	psu	35.0	1
Temperature	°C	15.0	2
Density	σ _t	1.025	3
Wind speed	m/s	5.0	4
Wave height	m	1.0	5
Current speed	cm/s	10.0	6
Current direction	°	90	7
Sea level pressure	hPa	1013.25	8
Relative humidity	%	70	9
Cloud cover	%	50	10
Surface albedo		0.06	11
Ice albedo		0.50	12
Soil albedo		0.20	13
Vegetation albedo		0.15	14

TABLE 1

Parameter	Symbol	Non-dimensional value	Dimensional value (V02)
planetary vorticity gradient	β	$0(V01)/10 (V02)$	$2 \times 10^{-13} \text{ cm s}^{-1}$
bottom friction	λ	0.1	$2 \times 10^{-8} \text{ s}^{-1}$
lateral dissipation	ν	1×10^{-3}	$2 \times 10^4 \text{ cm}^2 \text{ s}^{-1}$
forcing	τ	2×10^{-3}	$1.6 \times 10^{-9} \text{ cm}^2 \text{ s}^{-3}$
lateral diffusivity	η	5×10^{-3}	$1 \times 10^5 \text{ cm}^2 \text{ s}^{-1}$
area-mean kinetic energy	K	0.67	$2.7 \text{ cm}^2 \text{ s}^{-2}$
area-mean enstrophy	Z	36	$1.4 \times 10^{-12} \text{ s}^{-2}$
RMS rate of strain	γ	2.9	$5.8 \times 10^{-7} \text{ s}^{-1}$
domain size		2π	628 km
eddy scale		0.5	50 km
tracer spot size	2σ	0.17	17 km

Parameter values used to generate the V01 and V02 velocity fields.

Also shown are some of the resulting mean flow properties.

REFERENCES

- Bennett, A. F. and Haidvogel, D. B., 1983. Low-resolution numerical simulation of decaying two-dimensional turbulence. J. Atmos. Sciences, 40(3): 738-748.
- Freeland, H. J., Rhines, P. B. and Rossby, T., 1975. Statistical observations of the trajectories of neutrally buoyant floats in the North Atlantic. J. Mar. Res., 33: 383-404.
- Haidvogel, D. B., 1983. Chapter XVIII, Periodic and regional models. In: A. R. Robinson, editor, Eddies in Marine Science, Springer-Verlag.
- Haidvogel, D. B. Particle dispersion and Lagrangian vorticity conservation in models of β -plane turbulence, in preparation.
- Haidvogel, D. B. and T. Keffer, 1984. Tracer dispersal by mid-ocean mesoscale eddies. I. Ensemble statistics. Dyn. Atm. Oceans, 8: 1-40.
- Keffer, T., and Haidvogel, D. B. Tracer dispersal by mid-ocean mesoscale eddies. II: streakiness. In preparation.
- Orszag, S. A. and Israeli, M., 1974. Numerical simulation of viscous incompressible flows. Am. Rev. Fluid Mech., 6: 281-318.

APPENDIX A.

This appendix lists the source code for program process which generated the turbulent velocity field. The case $\hat{\beta} = 10$ (VO2 velocity field) is shown.

PROGRAM PROCESS

BAROTROPIC TURBULENCE ON A BETA-PLANE

INPUT PARAMETERS:

N 2K (TWICE THE CUTOFF WAVENUMBER;
 MUST BE A POWER OF 2)
 LP (N/2) + 1
 NP2 N+2
 DT TIME STEP
 BETA BETA
 RBF SURFACE FRICTION COEFFICIENT
 QNU LATERAL MIXING COEFFICIENT
 NONU = 1 => LAPLACIAN FRICTION
 = 2 => BIHARMONIC FRICTION
 = 3,4,... => TRIHARMONIC AND BEYOND!
 GAMMA ENERGY INPUT RATE (SEE FUNCTION FORCE)
 NTIMES NUMBER OF TIME STEPS TO BE TAKEN IN THIS RUN
 NRST NUMBER OF STEPS BETWEEN STORING RESTART DATA
 NDIAG NUMBER OF STEPS BETWEEN STORING DIAGNOSTICS
 NTRP HOW OFTEN TO TAKE A LEAPFROG-TRAPEZOIDAL STEP
 NPLOT HOW OFTEN TO MAKE CONTOUR PLOTS
 NREC NUMBER OF RESTART FILES ALREADY WRITTEN
 FOR THIS EXPERIMENT
 NRG NUMBER OF DIAGNOSTIC FILES ALREADY WRITTEN
 FOR THIS EXPERIMENT
 NDEAL = 0 => DEALIASING OPTION OFF
 = 1 => DEALIASING OPTION ON
 NCUT CUTOFF WAVENUMBER; I.E., /K+L/ < NCUT

INPUT INITIAL CONDITIONS: SEE SUBROUTINE INIT

TO RUN PROGRAM, YOU MUST ISSUE THESE THREE ASSIGN COMMANDS:

\$ASSIGN FILE1.XXX FOR006
 \$ASSIGN FILE2.XXX DIAGNOSTICS
 \$ASSIGN FILE3.XXX RESTART

WHERE FILE1.XXX, FILE2.XXX AND FILE3.XXX ARE TO CONTAIN
 THE PRINT-OUT, DIAGNOSTIC DATA AND RESTART FIELDS,
 RESPECTIVELY.

PARAMETER LP=65,N=128,NP2=130
 COMMON T,U1(2,LP,N),U1M(2,LP,N),S2(2,LP,N),
 1 S3(2,LP,N),S4(2,LP,N),S5(2,LP,N)
 COMMON /STR/ E0,E1(LP,N),E2(LP,N)
 DIMENSION CF1(LP),CI1(LP),OPRM(6),ID0(N)
 EQUIVALENCE (DT,OPRM)
 REAL JJ
 COMPLEX CI,CJ,CARG
 COMMON /PRM/ DT,QNU,BETA,RBF,NQNU
 COMMON /ND0/ IDP
 COMMON /NAR/ ND,NM,NP,NP2D,L,LM,LPD,LP2,NL,NLP,FAC
 COMMON /FT/ NF(2)
 COMMON /TRP/ CN,CNM1,MTRP
 COMMON /WVI/ FI2(LP)
 COMMON /WVJ/ FJ2(N)

```

3100      COMMON /FRC/  GAMMA
3150      COMMON /ADV/  NDEAL,NCUT
3200      COMMON /EPI/  CI(LP)
3250      COMMON /EPJ/  CJ(N)
3300      DATA      PI      /3.1415926536/
3350      C
3400      C      INITIALIZE PARAMETERS
3450      C
3500      ND=N
3550      NM=N-1
3600      NP=N+1
3650      NP2D=NP2
3700      L=LP-1
3750      LM=L-1
3800      LPD=LP
3850      LP2=L+2
3900      NL=N+L
3950      NLP=NL+1
4000      NW1=1 + 2*LP*N
4050      NW2=1 + 2*LP*N
4100      IOP=0
4150      C
4200      DT=0.005
4250      BETA=10.0
4300      RBF=0.1
4350      ONU=1.0E-03
4400      NONU=1
4450      GAMMA=2.0E-03
4500      C
4550      NTIMES=600
4600      NRST=100
4650      NDIAG=200
4700      NPLOT=200
4750      NTRP=17
4800      NREC=15
4850      NRG=0
4900      NDEAL=1
4950      NCUT=4*L/3
5000      C
5050      C      INITIALIZE FFT FOR NM X NM ARRAYS
5100      C
5150      FAC=1.0/FLOAT(N*N)
5200      NF(1)=N
5250      NF(2)=N
5300      C
5350      C      SET UP VECTORIZATION ARRAYS
5400      C
5450      DO 14  I=1,LP
5500      CARG=CMPLX( 0.0 , PI*FLOAT(I-1)/FLOAT(N) )
5550      CI(I)=CEXP(CARG)
5600      14  FI2(I)=FLOAT(I-1)**2
5650      DO 15  J=1,N
5700      IDO(J)=1
5750      IF(J.EQ.LP) IDO(J)=2
5800      CARG=CMPLX( 0.0 , PI*FLOAT(J-LP)/FLOAT(N) )
5850      CJ(J)=CEXP(CARG)
5900      15  FJ2(J)=FLOAT(J-LP)**2
5950      C
6000      C      READ CURRENT INITIAL CONDITIONS
6050      C
6100      IF(NREC.NE.0) GO TO 30

```



```

6150      T=0.0
6200      CALL INIT(U1,S2,LP)
6250      OPEN(UNIT=1,NAME='RESTART',TYPE='NEW',DISP='SAVE',
6300      1      ACCESS='SEQUENTIAL',FORM='UNFORMATTED')
6350      WRITE(1,ERR=101) (DPRM(I),I=1,6)
6400      CALL OPUT(T,1,NW1,NREC)
6450      GO TO 31
6500      101 WRITE(6,402)
6550      402 FORMAT('1','ERROR 101 ON OUTPUT IN PROCESS')
6600      STOP
6650      30 OPEN(UNIT=1,NAME='RESTART',TYPE='OLD',DISP='SAVE',
6700      1      ACCESS='SEQUENTIAL',FORM='UNFORMATTED')
6750      READ(1,ERR=104) (CF1(I),I=1,6)
6800      GO TO 105
6850      104 WRITE(6,403)
6900      403 FORMAT('1','ERROR 104 ON INPUT TO PROCESS')
6950      STOP
7000      105 WRITE(6,434) NREC
7050      434 FORMAT(' ','READ FOLLOWING ',I5,' RECORDS')
7100      DO 32 I=1,NREC
7150      CALL IPUT(T,1,NW1)
7200      32 CONTINUE
7250      31 WRITE(6,401) T,DT,ONU,RBF,BETA,XH,GAMMA
7300      401 FORMAT(' ','      T = ',1PE10.3, '//, '      DT = ',1PE10.3, '//,
7350      1      '      ONU = ',1PE10.3, '//, '      RBF = ',1PE10.3, '//,
7400      2      '      BETA = ',1PE10.3, '//, '      XH = ',1PE10.3, '//,
7450      3      '      GAMMA = ',1PE10.3)
7500      C
7550      C      SKIP NECESSARY NUMBER OF DIAGNOSTIC FILES
7600      C
7650      IF(NRG.NE.0) GO TO 40
7700      OPEN(UNIT=2,NAME='DIAGNOSTICS',TYPE='NEW',DISP='SAVE',
7750      1      ACCESS='SEQUENTIAL',FORM='UNFORMATTED')
7800      GO TO 41
7850      40 OPEN(UNIT=2,NAME='DIAGNOSTICS',TYPE='OLD',DISP='SAVE',
7900      1      ACCESS='SEQUENTIAL',FORM='UNFORMATTED')
7950      DO 42 I=1,NRG
8000      42 CALL IPUT(E0,2,NW2)
8050      41 CONTINUE
8053      C
8056      C      SCRAMBLE PHASES OF STREAMFUNCTION COMPONENTS
8059      C
8062      C      CALL SCRAMBL(U1,LP)
8100      C
8150      C      PLOT INITIAL FIELDS
8200      C
8250      IF(T.EQ.0.0) CALL PLT(U1,S2,N,LP)
8300      C
8350      C      ZERO OUT WORK AND STORAGE ARRAYS
8400      C
8450      DO 11 J=1,N
8500      DO 11 I=1,LP
8550      U1M(1,I,J)=U1(1,I,J)
8600      U1M(2,I,J)=U1(2,I,J)
8650      E1(I,J)=0.0
8700      E2(I,J)=0.0
8750      S2(1,I,J)=0.0
8800      S2(2,I,J)=0.0
8850      S3(1,I,J)=0.0
8900      11 S3(2,I,J)=0.0
8950      C

```

```

9000 C TAKE A MODIFIED FULER STEP FIRST, LEAPFROG THEREAFTER
9050 C
9100 MTRP=0
9150 DELT=DT
9200 CN=1.0
9250 CNM1=0.0
9300 IOP=0
9350 C
9400 C MAIN COMPUTATIONAL LOOP
9450 C
9500 DO 61 IIC=1,NTIMES
9550 T=T+DT
9600 IOP=IOP+1
9650 19 CONTINUE
9700 C
9750 C CALCULATE NONLINEAR TERMS
9800 C
9850 CALL JAC(U1,S3,S2,S4,S5,LP)
9900 C
9950 C TIME STEP ON VORTICITY EQUATION
10000 C
10050 DO 22 J=2,N
10100 JJ=FJ2(J)
10150 IST=IDO(J)
10200 IFN=MINO( NCUT-IABS(J-LP) , L )
10250 IFNP=IFN+1
10300 DO 23 I=IST,IFN
10350 CF1(I)=-((FI2(I)+JJ)/DELT) + 0.5*(QNU*((FI2(I)+JJ)**NQNU)
10400 1 +RBF)*(FI2(I)+JJ)
10450 23 CI1(I)=-((FI2(I)+JJ)/DELT) - 0.5*(QNU*((FI2(I)+JJ)**NQNU)
10500 1 +RBF)*(FI2(I)+JJ)
10550 DO 24 I=IST,IFN
10600 S2(1,I,J)=( CF1(I)*U1M(1,I,J)-S3(1,I,J) )/CI1(I)
10650 24 S2(2,I,J)=( CF1(I)*U1M(2,I,J)-S3(2,I,J) )/CI1(I)
10700 DO 25 I=IFNP,LP
10750 S2(1,I,J)=0.0
10800 25 S2(2,I,J)=0.0
10850 22 CONTINUE
10900 DO 26 I=1,LP
10950 S2(1,I,1)=0.0
11000 26 S2(2,I,1)=0.0
11050 S2(1,1,LP)=U1M(1,1,LP)
11100 S2(2,1,LP)=U1M(2,1,LP)
11150 S2(2,1,1)=0.0
11200 S2(2,LP,1)=0.0
11250 S2(2,LP,LP)=0.0
11300 C
11350 C SHUFFLE ARRAYS
11400 C
11450 IF(MTRP.EQ.1) GO TO 50
11500 DO 28 J=1,N
11550 DO 28 I=1,LP
11600 U1M(1,I,J)=U1(1,I,J)
11650 28 U1M(2,I,J)=U1(2,I,J)
11700 50 DO 20 J=1,N
11750 DO 20 I=1,LP
11800 U1(1,I,J)=S2(1,I,J)
11850 20 U1(2,I,J)=S2(2,I,J)
11900 C
11950 C TEST FOR LEAPFROG TRAPEZOIDAL STEP
12000 C SET TIME STEP FOR NEXT ITERATION

```



```

12050 C
12100 IF(IOP.EQ.1.AND.MTRP.EQ.0) GO TO 52
12150 IF(MOD(IOP,NTRP).NE.0.OR.MTRP.EQ.1) GO TO 51
12200 52 MTRP=1
12250 DELT=DT
12300 CN=0.5
12350 CNM1=1.0
12400 GO TO 19
12450 51 MTRP=0
12500 DELT=2.0*DT
12550 CN=1.0
12600 CNM1=0.0
12650 C
12700 C ACCUMULATE STATISTICS
12750 C
12800 DO 12 J=1,N
12850 DO 12 I=1,LP
12900 EO=U1(1,I,J)*U1(1,I,J) + U1(2,I,J)*U1(2,I,J)
12950 E1(I,J)=E1(I,J) + EO
13000 12 E2(I,J)=E2(I,J) + EO*EO
13050 C
13100 C TEST FOR DIAGNOSTICS AND OUTPUT
13150 C
13200 IF(MOD(IOP,NRST).EQ.0) CALL OPUT(T,1,NW1,NREC)
13250 IF(IOP.EQ.1) CALL SPECTRA(E1,E2,S4,S5,L,LP,1)
13300 IF(MOD(IOP,NPLOT).EQ.0) CALL PLT(U1,S2,N,LP)
13350 C
13400 C CALCULATE TIME-MEAN SPECTRA AND STANDARD DEVIATION
13450 C
13500 IF(MOD(IOP,NDIAG).NE.0) GO TO 61
13550 EO=FLOAT(NDIAG)
13600 CALL OPUT(E0,2,NW2,NRG)
13650 CALL SPECTRA(E1,E2,S4,S5,L,LP,NDIAG)
13700 C
13750 61 CONTINUE
13800 CLOSE(UNIT=1,DISP='SAVE')
13850 WRITE(6,33) NREC
13900 33 FORMAT(1H0,I5,' RESTART RECORDS WRITTEN')
13950 CLOSE(UNIT=2,DISP='SAVE')
14000 WRITE(6,43) NRG
14050 43 FORMAT(1H0,I5,' DIAGNOSTIC RECORDS WRITTEN')
14100 STOP
14150 END
14152 SUBROUTINE SCRAMBL(P,LP)
14154 COMPLEX P(LP,1),ARG
14156 COMMON /NAR/ N
14158 INTEGER*4 IRAN
14160 IRAN=36723477
14162 PI=3.1415926536
14164 DO 200 J=1,N
14166 DO 200 I=1,LP
14168 ARG=CMPLX( 0.0 , 2.0*PI*IRAN(IRAN) )
14170 P(I,J)=P(I,J)*CEXP(ARG)
14172 200 CONTINUE
14174 RETURN
14176 END
14200 SUBROUTINE SPECTRA(E,S,EE,SS,L,LP,NAV)
14250 COMMON T
14300 DIMENSION E(LP,1),S(LP,1),EE(L,1),SS(L,1),
14350 1 CE(98),CS(98),CDE(98),CSE(98),
14400 2 CZ(98),CDZ(98),CSZ(98),TOT(7),

```

```

14450      3      XPT(98),NC(98),LTIT(5)
14500      REAL      JJ,KK
14550      COMMON /NAR/ N,NM,NP,NP2,LD,LM,LPD
14600      COMMON /PRM/ DT,QNU,BETA,RBF,NQNU
14650      COMMON /NDO/ IDP
14700      COMMON /WVI/ FI2(1)
14750      COMMON /WVJ/ FJ2(1)
14800      DATA IDASH /-990099/
14850  C
14900  C      AVERAGE ENERGY SPECTRUM
14950  C
15000      ECON=0.5/FLOAT(NAV)
15050      SCON=0.25/FLOAT(NAV)
15100      DO 200 I=1,L
15150      EE(I,1)=ECON*E(I,LP)*FI2(I)
15200      SS(I,1)=SCON*S(I,LP)*FI2(I)*FI2(I)
15250      DO 201 J=2,L
15300      EE(I,J)=ECON*( E(I,J+L)+E(I,L+2-J) )*(FI2(I)+FJ2(J+L))
15350 201 SS(I,J)=SCON*( S(I,J+L)+S(I,L+2-J) )*(FI2(I)+FJ2(J+L))**2
15400      ECON=1.0/FLOAT(NAV)
15450      SCON=0.5/FLOAT(NAV)
15500 200 CONTINUE
15550  C
15600  C      AVERAGE STANDARD DEVIATION
15650  C
15700      DO 202 J=1,L
15750      DO 202 I=1,L
15800 202 SS(I,J)=SQRT( ABS(EE(I,J)*EE(I,J)-SS(I,J)) )
15850  C
15900  C      PLOT E(K,L)
15950  C
16000      IF(T.GE.0.0) GO TO 101
16050      ENCODE(20,600,LTIT) T
16100 600 FORMAT(12HE(K,L) (T=,F5.2,1H),2X)
16150      FL=FLOAT(L)
16200      CALL SET(0.1,0.9,0.1,0.9,1.0,FL,1.0,FL,1)
16250      CALL PERIM(1,LM,1,LM)
16300      CALL PWRITY(512,950,LTIT,18,1,0,0)
16350      CALL CONREC(EE,L,L,L,0.0,0.0,-10.0,1,-1,IDASH)
16400      CALL FRAME
16450 101 CONTINUE
16500  C
16550  C      CALCULATE AVERAGE K WAVENUMBER SPECTRA
16600  C
16650      DO 210 I=1,L
16700      CE(I)=0.0
16750      CS(I)=0.0
16800      CDE(I)=0.0
16850      CSE(I)=0.0
16900      CZ(I)=0.0
16950      CDZ(I)=0.0
17000 210 CSZ(I)=0.0
17050      DO 211 J=1,L
17100      JJ=FJ2(J+L)
17150      DO 211 I=1,L
17200      KK=FI2(I)+JJ
17250      CE(I)=CE(I) + EE(I,J)
17300      CS(I)=CS(I) + SS(I,J)
17350      CDE(I)=CDE(I) - 2.0*QNU*(KK**NQNU)*EE(I,J)
17400      CSE(I)=CSE(I) - 2.0*RBF*EE(I,J)
17450      CZ(I)=CZ(I) + 2.0*KK*EE(I,J)

```



```

17500      CDZ(I)=CDZ(I) - 4.0*ONU*KK*(KK**NQNU)*EE(I,J)
17550      211 CSZ(I)=CSZ(I) - 4.0*RBF*KK*EE(I,J)
17600      DO 212 I=1,7
17650      212 TOT(I)=0.0
17700      DO 213 I=1,L
17750      TOT(1)=TOT(1)+CE(I)
17800      TOT(2)=TOT(2)+CS(I)
17850      TOT(3)=TOT(3)+CDE(I)
17900      TOT(4)=TOT(4)+CSE(I)
17950      TOT(5)=TOT(5)+CZ(I)
18000      TOT(6)=TOT(6)+CDZ(I)
18050      213 TOT(7)=TOT(7)+CSZ(I)
18100      C
18150      C      PRINT OUT K WAVENUMBER SPECTRA
18200      C
18250      WRITE(6,400) T
18300      400 FORMAT(' ',//,,' TIME = ',F11.3,/,',9X','KE',7X,
18350      1      'SD(KE)',4X,'DISS(KE)',5X,'SURF(KE)',7X,'Z',7X,
18400      2      'DISS(Z)',3X,'SURF(Z)',/)
18450      DO 214 I=1,L
18500      214 WRITE(6,403) I,CE(I),CS(I),CDE(I),CSE(I),CZ(I),CDZ(I),CSZ(I)
18550      403 FORMAT(' ',I3,7(1PE11.3))
18600      WRITE(6,404) (TOT(I),I=1,7)
18650      404 FORMAT(' TOT',7(1PE11.3))
18700      C
18750      C      CALCULATE CIRCULAR SPECTRA
18800      C
18850      KMAX=INT( FLOAT(LP)*SQRT(2.0) + 0.5 )
18900      DO 203 I=1,KMAX
18950      NC(I)=0
19000      CE(I)=0.0
19050      CS(I)=0.0
19100      CDE(I)=0.0
19150      CSE(I)=0.0
19200      CZ(I)=0.0
19250      CDZ(I)=0.0
19300      203 CSZ(I)=0.0
19350      DO 204 J=1,L
19400      JJ=FJ2(J+L)
19450      DO 204 I=1,L
19500      KK=FI2(I)+JJ
19550      IR=INT( SQRT(KK) + 0.5 )
19600      IF(IR.EQ.0 .OR. EE(I,J).EQ.0.0) GO TO 204
19650      NC(IR)=NC(IR) + 1
19700      CE(IR)=CE(IR) + EE(I,J)
19750      CS(IR)=CS(IR) + SS(I,J)
19800      CDE(IR)=CDE(IR) - 2.0*QNU*(KK**NQNU)*EE(I,J)
19850      CSE(IR)=CSE(IR) - 2.0*RBF*EE(I,J)
19900      CZ(IR)=CZ(IR) + 2.0*KK*EE(I,J)
19950      CDZ(IR)=CDZ(IR) - 4.0*QNU*KK*(KK**NQNU)*EE(I,J)
20000      204 CSZ(IR)=CSZ(IR) - 4.0*RBF*KK*EE(I,J)
20050      C
20100      C      PLOT CIRCULAR SPECTRA
20150      C
20155      IF(T.GE.0.0) GO TO 208
20200      DO 209 I=1,KMAX
20250      209 XPT(I)=FLOAT(I-1)
20300      FK=FLOAT(KMAX)
20350      ENCODE(20,601,LTIT) T
20400      601 FORMAT(12HE,Z,DZ (T =,F5.2,1H),2X)
20450      CALL ANOTAT(2H $,2H $,0,2,0,0)

```

```

20500      CALL DISPLA(2,1,1)
20550      CALL SET(0.15,0.95,0.7,0.9,1.0,FK,1.0,FK,1)
20600      CALL EZXY(XPT,CE,KMAX,0)
20650      CALL SET(0.15,0.95,0.4,0.6,1.0,FK,1.0,FK,1)
20700      CALL EZXY(XPT,CZ,KMAX,0)
20750      CALL SET(0.15,0.95,0.1,0.3,1.0,FK,1.0,FK,1)
20800      CALL EZXY(XPT,CDZ,KMAX,0)
20850      CALL PWRITY(512,975,LTIT,18,1,0,0)
20900      CALL FRAME
20905      208 CONTINUE
20950      C
21000      C      PRINT OUT CIRCULAR SPECTRA
21050      C
21100      WRITE(6,405)
21150      405 FORMAT(////)
21200      DO 205 I=1,KMAX
21250      205 WRITE(6,401) I,NC(I),CE(I),CS(I),CDF(I),CSE(I),
21300      1      CZ(I),CDZ(I),CSZ(I)
21350      401 FORMAT(' ',2I3,7(1PE11.3))
21400      C
21450      C      ZERO OUT STORAGE ARRAYS
21500      C
21550      IF(IOP.EQ.1) RETURN
21600      DO 206 J=1,N
21650      DO 206 I=1,LP
21700      E(I,J)=0.0
21750      206 S(I,J)=0.0
21800      RETURN
21850      END
21900      SUBROUTINE PRTU(U,LP,N)
21950      COMMON T
22000      DIMENSION U(2,LP,N)
22050      WRITE(6,400) T
22100      DO 200 J=1,N
22150      200 WRITE(6,401) J,((U(1,I,J),U(2,I,J)),I=1,LP)
22200      400 FORMAT(' ',//,' TIME = ',F11.3)
22250      401 FORMAT(' ',I3,13(10(1PE11.3),/,4X))
22300      RETURN
22350      END
22400      SUBROUTINE PLT(P,S,N,LP)
22450      DIMENSION P(2,LP,1),S(2,LP,1),LTIT(4)
22500      COMMON T
22550      COMMON /NAR/ ND,NM,NP,NP2,L,LM
22600      COMMON /WVI/ FI2(1)
22650      COMMON /WVJ/ FJ2(1)
22700      COMMON /FT/ NF(1)
22750      DATA IDASH /-682/
22800      NPTS=N
22850      NPRM=(NPTS/2) - 1
22900      FNPTS=FLOAT(NPTS)
22950      C
23000      DO 200 J=1,N
23050      DO 200 I=1,LP
23100      DO 200 K=1,2
23150      200 S(K,I,J)=P(K,I,J)
23200      CALL SHUF(S,LP,0)
23250      CALL FOUR2(S,NF,2,1,-1)
23300      ENCODE(16,600,LTIT) T
23350      600 FORMAT(9HPSI (T =,F5.2,1H),1X)
23400      CALL SET(0.1,0.9,0.1,0.9,1.0,FNPTS,1.0,FNPTS,1)
23450      CALL PERIM(1,NPRM,1,NPRM)

```



```

23500      CALL PWRITY(512,950,LTIT,15,1,0,0)
23550      CALL CONREC(S,N,NPTS,NPTS,0.0,0.0,-10.0,1,-1,IDASH)
23600      CALL FRAME
23650      C
23700      DO 201 J=1,N
23750      DO 201 I=1,LP
23800      DO 201 K=1,2
23850      201 S(K,I,J)=(-FI2(I)-FJ2(J))*P(K,I,J)
23900      CALL SHUF(S,LP,0)
23950      CALL FOUR2(S,NF,2,1,-1)
24000      ENCODE(16,601,LTIT) T
24050      601 FORMAT(9HVRT (T =,F5.2,1H),1X)
24100      CALL SET(0.1,0.9,0.1,0.9,1.0,FNPTS,1.0,FNPTS,1)
24150      CALL PERIM(1,NPRM,1,NPRM)
24200      CALL PWRITY(512,950,LTIT,15,1,0,0)
24250      CALL CONREC(S,N,NPTS,NPTS,0.0,0.0,-7.0,1,-1,IDASH)
24300      CALL FRAME
24350      RETURN
24400      FND
24450      SUBROUTINE JAC(S,ADV,U,V,Z,LP)
24500      DIMENSION S(2,LP,1),ADV(2,LP,1),U(2,LP,1),
24550      1 V(2,LP,1),Z(2,LP,1),FRC(64),IM(128),
24600      2 JM(128)
24650      REAL IM,JM,J1,J2,KK
24700      COMMON T,UB
24750      COMMON /PRM/ DT,ONU,BETA,RRF
24800      COMMON /NAR/ N,NM,NP,NP2,L,LM,LPD,LP2,NL,NLP,FAC
24850      COMMON /TRP/ CN,CNM1,MTRP
24900      COMMON /WVI/ FI2(1)
24950      COMMON /WVJ/ FJ2(1)
25000      COMMON /FT/ NF(2)
25050      COMMON /ADV/ NDEAL,NCUT
25100      DO 200 J=1,N
25150      200 JM(J)=FLOAT(J-LP)
25200      JM(1)=0.0
25250      DO 201 I=1,LP
25300      201 IM(I)=FLOAT(I-1)
25350      IM(LP)=0.0
25400      C
25450      DO 203 J=1,N
25500      IFN=MIN0(NCUT-IABS(J-LP),L)
25550      DO 204 I=1,IFN
25600      204 FRC(I)=FORCE(S,LP,I,J)
25650      DO 203 I=1,IFN
25700      ADV(1,I,J)=CNM1*ADV(1,I,J) - BETA*IM(I)*S(2,I,J)
25750      1 + FRC(I)*S(1,I,J)
25800      203 ADV(2,I,J)=CNM1*ADV(2,I,J) + BETA*IM(I)*S(1,I,J)
25850      1 + FRC(I)*S(2,I,J)
25900      C
25950      DEALIASING LOOP
26000      ILP=NDEAL+1
26050      FFAC=FAC/FLOAT(ILP)
26100      DO 210 II=1,ILP
26150      DO 211 J=1,N
26200      J1=JM(J)
26250      J2=FJ2(J)
26300      DO 212 I=1,LP
26350      U(1,I,J)= J1*S(2,I,J)
26400      U(2,I,J)=-J1*S(1,I,J)
26450      V(1,I,J)=-IM(I)*S(2,I,J)
26500      V(2,I,J)= IM(I)*S(1,I,J)
26550      Z(1,I,J)=(-FI2(I)-J2)*S(1,I,J)

```

```

26550 212 Z(2,I,J)=(-FI2(I)-J2)*S(2,I,J)
26600 211 CONTINUE
26650 JJ=II-1
26700 CALL SHUF(U,LP,JJ)
26750 CALL FOUR2(U,NF,2,1,-1)
26800 CALL SHUF(V,LP,JJ)
26850 CALL FOUR2(V,NF,2,1,-1)
26900 CALL SHUF(Z,LP,JJ)
26950 CALL FOUR2(Z,NF,2,1,-1)
27000 DO 220 J=1,N
27050 DO 220 I=1,LP
27100 DO 222 K=1,2
27150 U(K,I,J)=U(K,I,J)*Z(K,I,J)
27200 222 V(K,I,J)=V(K,I,J)*Z(K,I,J)
27250 220 CONTINUE
27300 JJ=2*(II-1)
27350 CALL FOUR2(U,NF,2,-1,0)
27400 CALL SHUF(U,LP,JJ)
27450 CALL FOUR2(V,NF,2,-1,0)
27500 CALL SHUF(V,LP,JJ)
27550 DO 230 J=1,N
27600 J1=JM(J)
27650 IFN=MINO( NCUT-IABS(J-LP) , L )
27700 DO 231 I=1,IFN
27750 ADV(1,I,J)=ADV(1,I,J) + FFAC*( -IM(I)*U(2,I,J)
27800 1 -J1*V(2,I,J) )
27850 231 ADV(2,I,J)=ADV(2,I,J) + FFAC*( +IM(I)*U(1,I,J)
27900 1 +J1*V(1,I,J) )
27950 230 CONTINUE
28000 210 CONTINUE
28050 DO 240 J=1,N
28100 IFN=MINO( NCUT-IABS(J-LP) , L )
28150 DO 241 I=1,IFN
28200 ADV(1,I,J)=CN*ADV(1,I,J)
28250 241 ADV(2,I,J)=CN*ADV(2,I,J)
28300 240 CONTINUE
28350 RETURN
28400 END
28450 SUBROUTINE SHUF(C,LP,II)
28500 COMPLEX C(LP,1),CI,CJ,TMP
28550 COMMON /NAR/ N
28600 COMMON /EPI/ CI(1)
28650 COMMON /EPJ/ CJ(1)
28700 NH=N/2
28750 IF(II.NE.1) GO TO 100
28800 DO 200 J=1,N
28850 DO 200 I=1,LP
28900 200 C(I,J)=CI(I)*CJ(J)*C(I,J)
28950 100 DO 201 I=1,LP
29000 DO 202 J=1,NH
29050 TMP=C(I,J)
29100 C(I,J)=C(I,J+NH)
29150 202 C(I,J+NH)=TMP
29200 201 CONTINUE
29250 IF(II.NE.2) RETURN
29300 DO 203 J=1,N
29350 DO 203 I=1,LP
29400 203 C(I,J)=C(I,J)/(CI(I)*CJ(J))
29450 RETURN
29500 END
29550 SUBROUTINE OPUT(T,IUNT,NW,NREC)

```



```

29600 DIMENSION T(NW)
29650 WRITE(IUNT,ERR=100) T
29700 NREC=NREC+1
29750 RETURN
29800 100 WRITE(6,400) X
29850 400 FORMAT('1','BUFFERING ERROR IN OPUT = ',F4.1)
29900 STOP
29950 END
30000 SUBROUTINE IPUT(T,IUNT,NW)
30050 DIMENSION T(NW)
30100 READ(IUNT,ERR=100) T
30150 RETURN
30200 100 WRITE(6,400) X
30250 400 FORMAT('1','BUFFERING ERROR IN IPUT = ',F4.1)
30300 STOP
30350 END
30400 SUBROUTINE INIT(U1,S2,LP)
30450 DIMENSION U1(2,LP,1),S2(2,LP,1)
30500 REAL II,JJ,KK
30550 INTEGER*4 IRAN
30600 COMMON /NAR/ N,NM,NP,NP2,L,LM,LPD,LP2,NL,NLP,FAC
30650 COMMON /FT/ NF(2)
30700 COMMON /WVI/ FI2(1)
30750 COMMON /WVJ/ FJ2(1)
30800 DATA PI /3.1415926535897931/
30850 C
30900 C SET STREAMFUNCTION FIELDS
30950 C
31000 IRAN=319567
31050 DO 202 I=1,LP,L
31100 DO 202 J=2,L
31150 AMP=ENRG(I,J,LP)
31200 ARG=2.0*PI*IRAN
31250 CARG=COS(ARG)
31300 SARG=SIN(ARG)
31350 U1(1,I,J)= AMP*CARG
31400 U1(2,I,J)= AMP*SARG
31450 U1(1,I,N+2-J)= AMP*CARG
31500 202 U1(2,I,N+2-J)=-AMP*SARG
31550 DO 203 J=1,N
31600 DO 203 I=2,L
31650 AMP=ENRG(I,J,LP)
31700 ARG=2.0*PI*IRAN
31750 CARG=COS(ARG)
31800 SARG=SIN(ARG)
31850 U1(1,I,J)=AMP*CARG
31900 203 U1(2,I,J)=AMP*SARG
31950 DO 204 I=1,LP,L
32000 DO 204 J=1,LP,L
32050 AMP=ENRG(I,J,LP)
32100 U1(1,I,J)=AMP
32150 204 U1(2,I,J)=0.0
32200 C
32250 RETURN
32300 END
32350 FUNCTION ENRG(I,J,LP)
32400 COMMON /WVI/ FI2(1)
32450 COMMON /WVJ/ FJ2(1)
32500 ENRG=0.0
32550 RR=SQRT( FI2(I) + FJ2(J) )
32600 IF(RR.GT.0.0 .AND. RR.LE.20.0)

```

```

32650      1      ENRG=SQRT(0.1E-03)*RR/( 1.0 + (RR**4.4)/600.0 )
32700      RETURN
32750      END
32800      FUNCTION FORCE(S,LP,I,J)
32850      DIMENSION S(2,LP,1)
32900      COMMON /FRC/  GAMMA
32950      COMMON /WVI/  FI2(1)
33000      COMMON /WVJ/  FJ2(1)
33050      FORCE=0.0
33100      IF(GAMMA.EQ.0.0) RETURN
33150      RR=SQRT( FI2(I)+FJ2(J) )
33200      IF(RR.GE.6.0 .AND. RR.LT.8.0) GO TO 100
33250      RETURN
33300      100 AMP=S(1,I,J)*S(1,I,J) + S(2,I,J)*S(2,I,J)
33350      IF(AMP.EQ.0.0) RETURN
33400      FORCE=GAMMA/AMP
33450      RETURN
33500      END

```

APPENDIX B.

This appendix lists the source code of the program ADVDIFF which advects and diffuses the tracer field, given the turbulent velocity field generated by program PROCESS (see Appendix A). The drop V02H06 is shown.

```

50      PROGRAM      ADVDIFF
100     C
150     C      ADVECTION-DIFFUSION IN A PERIODIC DOMAIN
200     C      ADVECTING VELOCITY FIELD READ FROM DISK
250     C
300     C
350     C      INPUT PARAMETERS:
400     C
450     C      N      2K (TWICE THE CUTOFF WAVENUMBER;
500     C              MUST BE A POWER OF 2)
550     C      LP      (N/2) + 1
600     C      NP2     N+2
650     C      DT      TIME STEP
700     C      RBF      SURFACE FRICTION COEFFICIENT
750     C      QNU      LATERAL MIXING COEFFICIENT
800     C      NONU     = 1  => LAPLACIAN FRICTION
850     C              = 2  => RIHARMONIC FRICTION
900     C              = 3,4,... => TRIHARMONIC AND BEYOND!
950     C      NTIMES   NUMBER OF TIME STEPS TO BE TAKEN IN THIS RUN
1000    C      NRST     NUMBER OF STEPS BETWEEN STORING RESTART DATA
1050    C      NDGN     HOW OFTEN TO CALCULATE TRACER DIAGNOSTICS
1100    C      NTRP     HOW OFTEN TO TAKE A LEAPFROG-TRAPEZOIDAL STEP
1150    C              (THIS OPTION IS NOT PRESENTLY AVAILABLE. TO
1200    C              AVOID PROBLEMS, SET NTRP TO A VERY LARGE
1250    C              INTEGER.)
1300    C      NPLLOT    HOW OFTEN TO MAKE CONTOUR PLOTS
1350    C      NREC      NUMBER OF RESTART FILES ALREADY WRITTEN
1400    C              FOR THIS EXPERIMENT
1450    C      NVLC      NUMBER OF VELOCITY RECORDS TO SKIP
1500    C      NDEAL     = 0  => DEALIASING OPTION OFF
1550    C              = 1  => DEALIASING OPTION ON
1600    C      NCUT      CUTOFF WAVENUMBER; I.E., /K+L/ < NCUT
1650    C
1700    C
1750    C      INPUT INITIAL CONDITIONS: SEE SUBROUTINE INIT
1800    C
1850    C
1900    C      TO RUN PROGRAM, YOU MUST ISSUE THESE THREE ASSIGN COMMANDS:
1950    C
2000    C      $ASSIGN FILE1.XXX FOR006
2050    C      $ASSIGN FILE2.XXX VELOCITY
2100    C      $ASSIGN FILE3.XXX RESTART
2150    C
2200    C      WHERE FILE1.XXX, FILE2.XXX AND FILE3.XXX ARE TO CONTAIN
2250    C      THE PRINT-OUT, VELOCITY AND RESTART FIELDS,
2300    C      RESPECTIVELY.
2350    C
2400    C      PARAMETER      LP=65,N=128,NP2=130
2450    C      COMMON        T,P(2,LP,N),TVLC,VLC(2,LP,N),PM(2,LP,N),
2500    C      1             S2(2,LP,N),S3(2,LP,N),S4(2,LP,N)
2550    C      COMMON /STR/    SC(401),SC2(401),CM(401,2),CM2(401,3)
2600    C      DIMENSION      CF1(LP),CI1(LP),OPRM(5),ID0(N)
2650    C      EQUIVALENCE     (DT,OPRM)
2700    C      REAL          JJ
2750    C      COMPLEX        CI,CJ,XSHFT,YSHFT,CARG
2800    C      COMMON /PRM/    DT,QNU,RBF,NQNU
2850    C      COMMON /NDO/    IOP,NDGN,NTST,NW3
2900    C      COMMON /NAR/    ND,NM,NP,NP2D,L,LM,LPD,LP2,NL,NLP,FAC
2950    C      COMMON /FT/     NF(2)
3000    C      COMMON /TRP/    CN,CNM1,MTRP
3050    C      COMMON /WVI/    FI2(LP)

```



```

3100      COMMON /WVJ/  FJ2(N)
3150      COMMON /ADV/  NDEAL,NCUT
3200      COMMON /EPI/  CI(LP)
3250      COMMON /EPJ/  CJ(N)
3300      COMMON /SHFT/  XSHFT(LP),YSHFT(N)
3350      DATA      PI      /3.1415926536/
3400  C
3450  C      INITIALIZE PARAMETEPS
3500  C
3550      ND=N
3600      NM=N-1
3650      NP=N+1
3700      NP2D=NP2
3750      L=LP-1
3800      LM=L-1
3850      LPD=LP
3900      LP2=L+2
3950      NL=N+L
4000      NLP=NL+1
4050      IOP=0
4100      NW1=2 + 4*LP*N
4150      NW3=1 + 2*LP*N
4200  C
4250      DT=0.005
4300      RBF=0.0
4350      ONU=0.005
4400      NONU=1
4450  C
4500      NTIMES=800
4550      NRST=20
4600      NDGN=100000
4650      NPLOT=100
4700      NTRP=17
4750      NREC=0
4800      NVLC=821
4850      NDEAL=1
4900      NCUT=4*L/3
4950      NTST=1
5000      IF(NDGN.EQ.1 .OR. NDGN.GT.NTIMES) NTST=0
5050  C
5100  C      SET SHIFT FOR VELOCITY FIELD
5150  C
5200      DELX=0.0
5250      DELY=PI
5300  C
5350  C      INITIALIZE FFT FOR NN X NN ARRAYS
5400  C
5450      FAC=1.0/FLOAT(N*N)
5500      NF(1)=N
5550      NF(2)=N
5600  C
5650  C      SET UP VECTORIZATION ARRAYS
5700  C
5750      DO 14  I=1,LP
5800      CARG=CMPLX( 0.0 , PI*FLOAT(I-1)/FLOAT(N) )
5850      CI(I)=CEXP(CARG)
5900      CARG=CMPLX( 0.0 , DELX*FLOAT(I-1) )
5950      XSHFT(I)=CEXP(CARG)
6000      14  FI2(I)=FLOAT(I-1)**2
6050      DO 15  J=1,N
6100      IDO(J)=1

```

```

6150      IF(J.EQ.LP) IDO(J)=2
6200      CARG=CMPLX( 0.0 , PI*FLOAT(J-LP)/FLOAT(N) )
6250      CJ(J)=CEXP(CARG)
6300      CARG=CMPLX( 0.0 , DELY*FLOAT(J-LP) )
6350      YSHFT(J)=CEXP(CARG)
6400      15 FJ2(J)=FLOAT(J-LP)**2
6450      C
6500      C      READ IN INITIAL VELOCITY FIELD
6550      C
6600      OPEN(UNIT=3,NAME='VELOCITY',TYPE='OLD',DISP='SAVE',
6650      1      ACCESS='SEQUENTIAL',FORM='UNFORMATTED',EXTENDS(1)=0)
6700      READ(3,ERR=107) (CF1(I),I=1,6)
6750      DO 44 I=1,NVLC
6800      44 CALL IPUT(TVLC,3,NW3)
6850      CALL SHIFT(VLC,XSHFT,YSHFT,LP)
6900      GO TO 106
6950      107 WRITE(6,407)
7000      407 FORMAT('1','OUTPUT ERROR 107 IN ADVDIFF')
7050      STOP
7100      106 CONTINUE
7150      C
7200      C      READ CURRENT INITIAL CONDITIONS
7250      C
7300      IF(NREC.NE.0) GO TO 30
7350      T=0.0
7400      CALL INIT(P,LP,N)
7450      OPEN(UNIT=1,NAME='RESTART',TYPE='NEW',DISP='SAVE',
7500      1      ACCESS='SEQUENTIAL',FORM='UNFORMATTED')
7550      WRITE(1,ERR=101) (DPRM(I),I=1,4)
7600      CALL OPUT(T,1,NW1,NREC)
7650      GO TO 31
7700      101 WRITE(6,402)
7750      402 FORMAT('1','ERROR 101 ON OUTPUT IN ADVDIFF')
7800      STOP
7850      30 OPEN(UNIT=1,NAME='RESTART',TYPE='OLD',DISP='SAVE',
7900      1      ACCESS='SEQUENTIAL',FORM='UNFORMATTED')
7950      READ(1,ERR=104) (CF1(I),I=1,4)
8000      GO TO 105
8050      104 WRITE(6,403)
8100      403 FORMAT('1','ERROR 104 ON INPUT TO ADVDIFF')
8150      STOP
8200      105 WRITE(6,434) NREC
8250      434 FORMAT(' ', 'READ FOLLOWING ',I5,' RECORDS')
8300      DO 32 I=1,NREC
8350      CALL IPUT(T,1,NW1)
8400      32 CONTINUE
8450      31 WRITE(6,401) T,DT,QNU,RBF
8500      401 FORMAT(' ', '      T = ',1PE10.3, '//, '      DT = ',1PE10.3, '//,
8550      1      '      QNU = ',1PE10.3, '//, '      RBF = ',1PE10.3)
8600      C
8650      C      PLOT INITIAL FIELDS
8700      C
8750      IF(T.EQ.0.0) CALL PLT(P,S2,S4,N,L,LP)
8800      C
8850      C      ZERO OUT WORK AND STORAGE ARRAYS
8900      C
8950      DO 11 J=1,N
9000      DO 11 I=1,LP
9050      PM(1,I,J)=P(1,I,J)
9100      PM(2,I,J)=P(2,I,J)
9150      S2(1,I,J)=0.0

```

```

9200      S2(2,I,J)=0.0
9250      S3(1,I,J)=0.0
9300      11 S3(2,I,J)=0.0
9350      C
9400      C      TAKE A MODIFIED EULER STEP FIRST, LEAPFROG THEREAFTER
9450      C
9500      MTRP=0
9550      DELT=DT
9600      CN=1.0
9650      CNM1=0.0
9700      IOP=0
9750      C
9800      C      MAIN COMPUTATIONAL LOOP
9850      C
9900      DO 61 IIC=1,NTIMES
9950      T=T+DT
10000      IOP=IOP+1
10050      19 CONTINUE
10100      C
10150      C      CALCULATE NONLINEAR TERMS
10200      C
10250      CALL JAC(P,VLC,S3,S2,S4,LP)
10300      C
10350      C      TIME STEP ON TRACER EQUATION
10400      C
10450      DO 22 J=2,N
10500      JJ=FJ2(J)
10550      IST=IND(J)
10600      IFN=MINO( NCUT-IARS(J-LP) , L )
10650      IFNP=IFN+1
10700      DO 23 I=IST,IFN
10750      CII(I)=(1.0/DELT) + 0.5*( RBF +
10800      1      ONU*((FI2(I)+JJ)**NONU) )
10850      23 CF1(I)=(1.0/DELT) - 0.5*( RBF +
10900      1      ONU*((FI2(I)+JJ)**NONU) )
10950      DO 24 I=IST,IFN
11000      S2(1,I,J)=( CF1(I)*PM(1,I,J)-S3(1,I,J) )/CII(I)
11050      24 S2(2,I,J)=( CF1(I)*PM(2,I,J)-S3(2,I,J) )/CII(I)
11100      DO 25 I=IFNP,LP
11150      S2(1,I,J)=0.0
11200      25 S2(2,I,J)=0.0
11250      22 CONTINUE
11300      DO 26 I=1,LP
11350      S2(1,I,1)=0.0
11400      26 S2(2,I,1)=0.0
11450      S2(1,1,LP)=PM(1,1,LP)
11500      S2(2,1,LP)=PM(2,1,LP)
11550      S2(2,1,1)=0.0
11600      S2(2,LP,1)=0.0
11650      S2(2,LP,LP)=0.0
11700      C
11750      C      SHUFFLE ARRAYS
11800      C
11850      IF(MTRP.EQ.1) GO TO 50
11900      DO 28 J=1,N
11950      DO 28 I=1,LP
12000      PM(1,I,J)=P(1,I,J)
12050      28 PM(2,I,J)=P(2,I,J)
12100      50 DO 20 J=1,N
12150      DO 20 I=1,LP
12200      P(1,I,J)=S2(1,I,J)

```



```

12250      20 P(2,I,J)=S2(2,I,J)
12300      C
12350      C      TEST FOR LEAPFROG TRAPEZOIDAL STEP
12400      C      SET TIME STEP FOR NEXT ITERATION
12450      C
12500      IF(MOD(IOP,NTRP).NE.0.OR.MTRP.EQ.1) GO TO 51
12550      52 CALL IPUT(TVLC,3,NW3)
12600      CALL SHIFT(VLC,XSHFT,YSHFT,LP)
12650      MTRP=1
12700      DELT=DT
12750      CN=0.5
12800      CNM1=1.0
12850      GO TO 19
12900      51 IF(MTRP.EQ.1) GO TO 53
12950      CALL IPUT(TVLC,3,NW3)
13000      CALL SHIFT(VLC,XSHFT,YSHFT,LP)
13050      53 MTRP=0
13100      DELT=2.0*DT
13150      CN=1.0
13200      CNM1=0.0
13250      C
13300      C      TEST FOR DIAGNOSTICS AND OUTPUT
13350      C
13400      IF(MOD(IOP,NRST).EQ.0) CALL OPUT(T,1,NW1,NREC)
13450      IF(MOD(IOP,NPLOT).EQ.0) CALL PLT(P,S2,S4,N,L,LP)
13500      C
13550      61 CONTINUE
13600      CLOSE(UNIT=1,DISP='SAVE')
13650      WRITE(6,33) NREC
13700      33 FORMAT(1H0,I5,' RESTART RECORDS WRITTEN')
13750      CLOSE(UNIT=3,DISP='SAVE')
13800      IF(NDGN.GT.NTIMES) STOP 'NO DIAGNOSTICS'
13850      CALL SHUF(P,LP,N)
13900      CALL FOUR2(P,NF,2,1,-1)
13950      CALL MOMENT(P,N)
14000      CALL DGNOUT
14050      STOP
14100      END
14150      SUBROUTINE MOMENT(T,N)
14200      DIMENSION T(N,1)
14250      COMMON /NAR/ NO,NM,NP,NP2,L,LM,LP
14300      COMMON /STR/ SC(401),SC2(401),CM(401,2),CM2(401,3)
14350      COMMON /DGN/ TPI,INDX
14400      DATA TPI,INDX /6.28318530,0/
14450      IF(INDX.GE.401) RETURN
14500      INDX=INDX+1
14550      SC(INDX)=0.0
14600      SC2(INDX)=0.0
14650      CM(INDX,1)=0.0
14700      CM(INDX,2)=0.0
14750      CM2(INDX,1)=0.0
14800      CM2(INDX,2)=0.0
14850      CM2(INDX,3)=0.0
14900      DO 200 J=1,N
14950      Y=TPI*FLOAT(J-LP)/FLOAT(N)
15000      DO 200 I=1,N
15050      X=TPI*FLOAT(I-LP)/FLOAT(N)
15100      SC(INDX)=SC(INDX)+T(I,J)
15150      SC2(INDX)=SC2(INDX)+T(I,J)*T(I,J)
15200      CM(INDX,1)=CM(INDX,1)+X*T(I,J)
15250      CM(INDX,2)=CM(INDX,2)+Y*T(I,J)

```



```

15300      200 CONTINUE
15350          CM(INDX,1)=CM(INDX,1)/SC(INDX)
15400          CM(INDX,2)=CM(INDX,2)/SC(INDX)
15450          DO 201 J=1,N
15500              Y= (TPI*FLOAT(J-LP)/FLOAT(N)) - CM(INDX,2)
15550              DO 201 I=1,N
15600                  X= (TPI*FLOAT(I-LP)/FLOAT(N)) - CM(INDX,1)
15650                  CM2(INDX,1)=CM2(INDX,1) + X*X*T(I,J)
15700                  CM2(INDX,2)=CM2(INDX,2) + X*Y*T(I,J)
15750                  CM2(INDX,3)=CM2(INDX,3) + Y*Y*T(I,J)
15800      201 CONTINUE
15850          CM2(INDX,1)=CM2(INDX,1)/SC(INDX)
15900          CM2(INDX,2)=CM2(INDX,2)/SC(INDX)
15950          CM2(INDX,3)=CM2(INDX,3)/SC(INDX)
16000          RETURN
16050          END
16100          SUBROUTINE SHIFT(V,XSHFT,YSHFT,LP)
16150          COMMON /NAR/ N
16200          COMPLEX V(LP,1),XSHFT(1),YSHFT(1)
16250          DO 200 J=1,N
16300              DO 200 I=1,LP
16350      200 V(I,J)=V(I,J)*XSHFT(I)*YSHFT(J)
16400          RETURN
16450          END
16500          SUBROUTINE DGNOUT
16550          COMMON T
16600          DIMENSION XPT(401),LTIT(6)
16650          COMMON /NAR/ N,NM,NP,NP2,L,LM,LP
16700          COMMON /STR/ SC(401),SC2(401),CM(401,2),CM2(401,3)
16750          COMMON /DGN/ TPI,INDX
16800          COMMON /PRM/ DT
16850          COMMON /NDO/ INP,NDOGN
16900          DO 200 I=1,INDX
16950      200 XPT(I)=FLOAT( (I-1)*NDOGN )*DT
17000          WRITE(6,400)
17050      400 FORMAT('1')
17100          TSC=SC(1)
17150          TSC2=SC2(1)
17200          TCM1=CM(1,1)
17250          TCM2=CM(1,2)
17300          TCM21=CM2(1,1)
17350          TCM22=CM2(1,2)
17400          TCM23=CM2(1,3)
17450          WRITE(6,402) TSC,TSC2,TCM1,TCM2,TCM21,TCM22,TCM23
17500      402 FORMAT(' ',12X,7(1PE12.4))
17550          DO 201 I=1,INDX
17600              SC(I)=SC(I)-TSC
17650              SC2(I)=SC2(I)-TSC2
17700              CM(I,1)=CM(I,1)-TCM1
17750              CM(I,2)=CM(I,2)-TCM2
17800              CM2(I,1)=CM2(I,1)-TCM21
17850              CM2(I,2)=CM2(I,2)-TCM22
17900              CM2(I,3)=CM2(I,3)-TCM23
17950      201 WRITE(6,401) XPT(I),SC(I),SC2(I),CM(I,1),CM(I,2),
18000      1 CM2(I,1),CM2(I,2),CM2(I,3)
18050      401 FORMAT(' ',8(1PE12.4))
18100          CALL ANOTAT(2H $,2H $,0,2,0,0)
18150          CALL DISPLA(2,1,1)
18200          ENCODE(24,600,LTIT) T
18250      600 FORMAT(15HS(T),S(T2) (T=,F5.2,1H),3X)
18300          FINDX=FLOAT(INDX)

```

```

18350      CALL SET(0.15,0.95,0.5,0.9,1.0,FINDX,1.0,FINDX,1)
18400      CALL EZXY(XPT,SC,INDX,0)
18450      CALL SET(0.15,0.95,0.05,0.45,1.0,FINDX,1.0,FINDX,1)
18500      CALL EZXY(XPT,SC2,INDX,0)
18550      CALL PWRITY(512,950,LTIT,21,1,0,0)
18600      CALL FRAME
18650      ENCODE(24,601,LTIT) T
18700 601  FORMAT(12HCMX,CMY (T =,F5.2,1H),6X)
18750      CALL SET(0.15,0.95,0.5,0.9,1.0,FINDX,1.0,FINDX,1)
18800      CALL EZXY(XPT,CM(1,1),INDX,0)
18850      CALL SET(0.15,0.95,0.05,0.45,1.0,FINDX,1.0,FINDX,1)
18900      CALL EZXY(XPT,CM(1,2),INDX,0)
18950      CALL PWRITY(512,950,LTIT,18,1,0,0)
19000      CALL FRAME
19050      ENCODE(24,602,LTIT) T
19100 602  FORMAT(16HCMXX,CMXY,CMYY (,F5.2,1H),2X)
19150      CALL SET(0.15,0.95,0.7,0.9,1.0,FINDX,1.0,FINDX,1)
19200      CALL EZXY(XPT,CM2(1,1),INDX,0)
19250      CALL SET(0.15,0.95,0.4,0.6,1.0,FINDX,1.0,FINDX,1)
19300      CALL EZXY(XPT,CM2(1,2),INDX,0)
19350      CALL SET(0.15,0.95,0.1,0.3,1.0,FINDX,1.0,FINDX,1)
19400      CALL EZXY(XPT,CM2(1,3),INDX,0)
19450      CALL PWRITY(512,950,LTIT,22,1,0,0)
19500      CALL FRAME
19550      RETURN
19600      END
19650      SUBROUTINE PRTP(P,LP,N)
19700      COMMON T
19750      DIMENSION P(2,LP,N)
19800      WRITE(6,400) T
19850      DO 200 J=1,N
19900 200  WRITE(6,401) J,((P(1,I,J),P(2,I,J)),I=1,LP)
19950 400  FORMAT(' ',//,' TIME = ',F11.3)
20000 401  FORMAT(' ',I3,13(10(1PE11.3),/,4X))
20050      RETURN
20100      END
20150      SUBROUTINE PLT(P,S,R,N,L,LP)
20200      DIMENSION P(2,LP,1),S(2,LP,1),LTIT(5)
20250      DIMENSION R(L,1),CBR(101),CN(101),XPT(101)
20300      COMMON T
20350      COMMON /NAR/ ND,NM,NP,NP2,LD,LM
20400      COMMON /PRM/ DT,QNU,RBF,NQNU
20450      COMMON /WVI/ FI2(1)
20500      COMMON /WVJ/ FJ2(1)
20550      COMMON /FT/ NF(1)
20600      DATA IDASH /-990099/
20650      DATA DCON,DR /0.05,0.1/
20700      NPTS=N
20750      NTCKS=N/2 - 1
20800      FNPTS=FLOAT(NPTS)
20850  C
20900      DO 200 J=1,N
20950      DO 200 I=1,LP
21000      DO 200 K=1,2
21050 200  S(K,I,J)=P(K,I,J)
21100      CALL SHUF(S,LP,N)
21150      CALL FOUR2(S,NF,2,1,-1)
21200      ENCODE(16,600,LTIT) T
21250 600  FORMAT(9HTRCR (T =,F5.2,1H),1X)
21300      CALL SET(0.1,0.9,0.1,0.9,1.0,FNPTS,1.0,FNPTS,1)
21350      IF(T.GE.0.0) GO TO 100

```



```

21400      CALL PERIM(1,NTCK5,1,NTCK5)
21450      CALL PWRITY(512,950,LTIT,15,1,0,0)
21500      CALL CONREC(S,N,NPTS,NPTS,-1.45,1.55,0.1,1,-1,IDASH)
21550      CALL FRAME
21600      100 CONTINUE
21650      CALL PERIM(1,NTCK5,1,NTCK5)
21700      CALL PWRITY(512,950,LTIT,15,1,0,0)
21750      CALL CONREC(S,N,NPTS,NPTS,0.0,0.0,-10.0,1,-1,IDASH)
21800      CALL FRAME
21850      IF(T.GE.0.0) RETURN
21900      C
21950      CALL CBAR(S,N,CBR,DR,NCBR,CN,DCON,NCN)
22000      CALL ANOTAT(2H $,2H $,0,2,0,0)
22050      CALL DISPLA(2,1,1,1)
22100      ENCODE(20,603,LTIT) T
22150      603 FORMAT(14HC(R),N(C) (T =,F5.2,1H))
22200      DO 204 I=1,NCBR
22250      204 XPT(I)=FLOAT(I-1)*DR
22300      CALL SET(0.15,0.95,0.55,0.95,1.0,FLOAT(NCBR),1.0,
22350      1      FLOAT(NCBR),1)
22400      CALL EZXY(XPT,CBR,NCBR,0)
22450      DO 205 I=1,NCN
22500      205 XPT(I)=FLOAT(I-1)*DCON
22550      CALL SET(0.15,0.95,0.05,0.45,1.0,FLOAT(NCN),1.0,
22600      1      FLOAT(NCN),1)
22650      CALL EZXY(XPT,CN,NCN,0)
22700      CALL PWRITY(512,990,LTIT,20,1,0,0)
22750      CALL FRAME
22800      C
22850      CON=1.0
22900      DO 202 I=1,L
22950      R(I,1)=CON*( P(1,I,LP)**2 + P(2,I,LP)**2 )
23000      DO 203 J=2,L
23050      203 R(I,J)=CON*( P(1,I,J+L)**2 + P(2,I,J+L)**2
23100      1      + P(1,I,L+2-J)**2 + P(2,I,L+2-J)**2 )
23150      CON=2.0
23200      202 CONTINUE
23250      ENCODE(16,602,LTIT) T
23300      602 FORMAT(9HT(K) (T =,F5.2,1H),1X)
23350      FL=FLOAT(L)
23400      CALL SET(0.1,0.9,0.1,0.9,1.0,FL,1.0,FL,1)
23450      CALL PERIM(1,LM,1,LM)
23500      CALL PWRITY(512,950,LTIT,15,1,0,0)
23550      CALL CONREC(R,L,L,L,0.0,0.0,-8.0,1,-1,IDASH)
23600      CALL FRAME
23650      C
23700      DO 201 J=1,N
23750      DO 201 I=1,LP
23800      DO 201 K=1,2
23850      201 S(K,I,J)=-QNU*( (FI2(I)-FJ2(J))**NQNU ) *P(K,I,J)
23900      CALL SHUF(S,LP,N)
23950      CALL FOUR2(S,NF,2,1,-1)
24000      ENCODE(16,601,LTIT) T
24050      601 FORMAT(9HDIFF (T =,F5.2,1H),1X)
24100      CALL SET(0.1,0.9,0.1,0.9,1.0,FNPTS,1.0,FNPTS,1)
24150      CALL PERIM(1,NTCK5,1,NTCK5)
24200      CALL PWRITY(512,950,LTIT,15,1,0,0)
24250      CALL CONREC(S,N,NPTS,NPTS,0.0,0.0,-8.0,1,-1,IDASH)
24300      CALL FRAME
24350      RETURN
24400      END

```

```

24450      SUBROUTINE      CBR(P,N,CBR,DR,NCBR,CN,DCON,NCN)
24500      DIMENSION      P(N,1),CBR(1),CN(1),LC(101)
24550      COMMON          T
24600      COMMON /DGN/    TPI
24650      C      DETERMINE CENTER OF MASS
24700      TC=0.0
24750      CMX=0.0
24800      CMY=0.0
24850      DO 200  J=1,N
24900      Y=TPI*FLOAT(J-LP)/FLOAT(N)
24950      DO 200  I=1,N
25000      X=TPI*FLOAT(I-LP)/FLOAT(N)
25050      TC=TC+P(I,J)
25100      CMX=CMX+X*P(I,J)
25150      200  CMY=CMY+Y*P(I,J)
25200      CMX=CMX/TC
25250      CMY=CMY/TC
25300      C      NUMBER OF POINTS IN GIVEN CONCENTRATION INTERVAL
25350      DO 201  I=1,101
25400      201  CN(I)=0.0
25450      NCN=0
25500      DO 202  J=1,N
25550      DO 202  I=1,N
25600      IF(P(I,J).LE.(DCON/10.0)) GO TO 202
25650      K=1 + INT(P(I,J)/DCON)
25700      IF(K.GT.101) GO TO 202
25750      IF(K.GT.NCN) NCN=K
25800      CN(K)=CN(K) + 1.0
25850      202  CONTINUE
25900      C      AVERAGE CONCENTRATION IN CIRCULAR RINGS
25950      DO 203  I=1,101
26000      LC(I)=0
26050      203  CBR(I)=0.0
26100      NCBR=0
26150      DO 204  J=1,N
26200      DY=TPI*FLOAT(J-LP)/FLOAT(N) - CMY
26250      DO 204  I=1,N
26300      DX=TPI*FLOAT(I-LP)/FLOAT(N) - CMX
26350      CR=SQRT(DY*DY + DX*DX)
26400      IR=1 + INT(CR/DR)
26450      IF(IR.GT.101) GO TO 204
26500      IF(IR.GT.NCBR) NCBR=IR
26550      LC(IR)=LC(IR)+1
26600      CBR(IR)=CBR(IR) + P(I,J)
26650      204  CONTINUE
26700      DO 205  I=1,NCBR
26750      IF(LC(I).LE.0) GO TO 205
26800      CBR(I)=CBR(I)/FLOAT(LC(I))
26850      205  CONTINUE
26900      C      PRINT OUT
26950      WRITE(6,400) T
27000      400  FORMAT('1','TIME =',F6.2,/)
27050      IMX=MAX0(NCBR,NCN)
27100      DO 206  I=1,IMX
27150      TC=FLOAT(I)*DCON
27200      CR=FLOAT(I)*DR
27250      206  WRITE(6,401) I,TC,CN(I),CR,CBR(I)
27300      401  FORMAT(' ',I3,5X,2(1PE11.3),5X,2(1PE11.3))
27350      RETURN
27400      END
27450      SUBROUTINE      JAC(P,S,ADV,V,C,LP)

```



```

27500      DIMENSION      P(2,LP,1),ADV(2,LP,1),V(2,LP,1),
27550      1                C(2,LP,1),S(2,LP,1),
27600      2                IM(128),JM(128)
27650      REAL            IM,JM,J1,J2,KK
27700      COMMON          T
27750      COMMON /PRM/     DT,QNU,RBF
27800      COMMON /NAR/     N,NM,NP,NP2,L,LM,LPO,LP2,NL,NLP,FAC
27850      COMMON /TRP/     CN,CNM1,MTRP
27900      COMMON /WVI/     FI2(1)
27950      COMMON /WVJ/     FJ2(1)
28000      COMMON /FT/      NF(2)
28050      COMMON /NDO/     IOP,NDGN,NTST,NW3
28100      COMMON /ADV/     NDEAL,NCUT
28150      DO 200 J=1,N
28200      200 JM(J)=FLOAT(J-LP)
28250      JM(1)=0.0
28300      DO 201 I=1,LP
28350      201 IM(I)=FLOAT(I-1)
28400      IM(LP)=0.0
28450      DO 202 J=1,N
28500      IFN=MINO( NCUT-IABS(J-LP) , L )
28550      DO 202 I=1,IFN
28600      ADV(1,I,J)=CNM1*ADV(1,I,J)
28650      202 ADV(2,I,J)=CNM1*ADV(2,I,J)
28700      C      DEALIASING LOOP
28750      ILP=NDEAL+1
28800      FFAC=FAC/FLOAT(ILP)
28850      DO 210 II=1,ILP
28900      DO 211 J=1,N
28950      J1=JM(J)
29000      DO 211 I=1,LP
29050      C(1,I,J)=P(1,I,J)
29100      C(2,I,J)=P(2,I,J)
29150      V(1,I,J)= J1*S(2,I,J)
29200      211 V(2,I,J)=-J1*S(1,I,J)
29250      JJ=II-1
29300      CALL SHUF(C,LP,JJ)
29350      CALL FOUR2(C,NF,2,1,-1)
29400      IF(MOD(IOP,NDGN).EQ.NTST .AND. II.EQ.1)
29450      1      CALL MOMENT(C,N)
29500      CALL SHUF(V,LP,JJ)
29550      CALL FOUR2(V,NF,2,1,-1)
29600      DO 212 J=1,N
29650      DO 212 I=1,LP
29700      DO 212 K=1,2
29750      212 V(K,I,J)=V(K,I,J)*C(K,I,J)
29800      JJ=2*JJ
29850      CALL FOUR2(V,NF,2,-1,0)
29900      CALL SHUF(V,LP,JJ)
29950      DO 213 J=1,N
30000      IFN=MINO( NCUT-IABS(J-LP) , L )
30050      DO 213 I=1,IFN
30100      ADV(1,I,J)=ADV(1,I,J) + FFAC*(-IM(I)*V(2,I,J))
30150      213 ADV(2,I,J)=ADV(2,I,J) + FFAC*( IM(I)*V(1,I,J))
30200      DO 221 J=1,N
30250      DO 221 I=1,LP
30300      V(1,I,J)=-IM(I)*S(2,I,J)
30350      221 V(2,I,J)= IM(I)*S(1,I,J)
30400      JJ=II-1
30450      CALL SHUF(V,LP,JJ)
30500      CALL FOUR2(V,NF,2,1,-1)

```



```

30550      DO 222 J=1,N
30600      DO 222 I=1,LP
30650      DO 222 K=1,2
30700      222 V(K,I,J)=V(K,I,J)*C(K,I,J)
30750      JJ=2*JJ
30800      CALL FOUR2(V,NF,2,-1,0)
30850      CALL SHUF(V,LP,JJ)
30900      DO 223 J=1,N
30950      J1=JM(J)
31000      IFN=MINO( NCUT-IABS(J-LP) , L )
31050      DO 223 I=1,IFN
31100      ADV(1,I,J)=ADV(1,I,J) + FFAC*(-J1*V(2,I,J))
31150      223 ADV(2,I,J)=ADV(2,I,J) + FFAC*( J1*V(1,I,J))
31200      210 CONTINUE
31250  C
31300      DO 230 J=1,N
31350      IFN=MINO( NCUT-IABS(J-LP) , L )
31400      DO 230 I=1,IFN
31450      ADV(1,I,J)=CN*ADV(1,I,J)
31500      230 ADV(2,I,J)=CN*ADV(2,I,J)
31550      RETURN
31600      END
31650      SUBROUTINE SHUF(C,LP,II)
31700      COMPLEX C(LP,1),CI,CJ,TMP
31750      COMMON /NAR/ N
31800      COMMON /EPI/ CI(1)
31850      COMMON /EPJ/ CJ(1)
31900      NH=N/2
31950      IF(II.NE.1) GO TO 100
32000      DO 200 J=1,N
32050      DO 200 I=1,LP
32100      200 C(I,J)=CI(I)*CJ(J)*C(I,J)
32150      100 DO 201 I=1,LP
32200      DO 202 J=1,NH
32250      TMP=C(I,J)
32300      C(I,J)=C(I,J+NH)
32350      202 C(I,J+NH)=TMP
32400      201 CONTINUE
32450      IF(II.NE.2) RETURN
32500      DO 203 J=1,N
32550      DO 203 I=1,LP
32600      203 C(I,J)=C(I,J)/(CI(I)*CJ(J))
32650      RETURN
32700      END
32750      SUBROUTINE OPUT(T,IUNT,NW,NREC)
32800      DIMENSION T(NW)
32850      COMMON /NAR/ N,NM,NP,NP2,L,LM,LP,LP2,NL,NLP,FAC
32900      WRITE(IUNT,ERR=101) T
32950      NREC=NREC+1
33000      RETURN
33050      101 WRITE(6,400) X
33100      400 FORMAT('1','BUFFERING ERROR IN OPUT = ',F4.1)
33150      STOP
33200      END
33250      SUBROUTINE IPUT(T,IUNT,NW)
33300      DIMENSION T(NW)
33350      COMMON /NAR/ N,NM,NP,NP2,L,LM,LP,LP2,NL,NLP,FAC
33400      READ(IUNT,ERR=101) T
33450      RETURN
33500      101 WRITE(6,400) X
33550      400 FORMAT('1','BUFFERING ERROR IN IPUT = ',F4.1)

```

```

33600      STOP
33650      END
33700      SUBROUTINE      INIT(P,LP,N)
33750      DIMENSION      P(N,1)
33800      REAL            II,JJ,KK
33850      COMMON /NAR/    NO,NM,NP,NP2,L,LM,LPD,LP2,NL,NLP,FAC
33900      COMMON /FT/      NF(2)
33950      COMMON /WVI/     FI2(1)
34000      COMMON /WVJ/     FJ2(1)
34050      DATA PI        /3.1415926535897931/
34100      C
34150      C      SET TRACER FIELD
34200      C
34210      SIG=1.777
34220      SIG2=SIG*SIG
34250      DO 200 I=1,N
34300      DX2=FLOAT(LP-I)**2
34350      DO 200 J=1,N
34400      DY2=FLOAT(LP-J)**2
34450      200 P(I,J)=FAC*EXP( -(DX2+DY2)/(2.0*SIG2) )
34500      CALL FOUR2(P,NF,2,-1,0)
34550      CALL SHUF(P,LP,N)
34600      C
34650      RETURN
34700      END

```

APPENDIX C.

This appendix contains a listing of the Fast Fourier Transform routine used by programs PROCESS and ADVDIFF.


```

100      SUBROUTINE FOUR2 (DATA,N,NDIM,ISIGN,IFORM)
200      COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN.
300      MULTI-DIMENSIONAL TRANSFORM, EACH DIMENSION A POWER OF TWO,
400      COMPLEX OR REAL DATA.
500      TRANSFORM(K1,K2,...) = SUM(DATA(J1,J2,...)*EXP(ISIGN*2*PI*SORT(-1)
600      *((J1-1)*(K1-1)/N(1)+(J2-1)*(K2-1)/N(2)+...))), SUMMED FOR ALL
700      J1 AND K1 FROM 1 TO N(1), J2 AND K2 FROM 1 TO N(2),
800      ETC. FOR ALL NDIM SUBSCRIPTS. NDIM MUST BE POSITIVE AND
900      EACH N(IDIM) MUST BE A POWER OF TWO. ISIGN IS +1 OR -1.
1000     LET NTOT = N(1)*N(2)*...*N(NDIM). THEN A -1 TRANSFORM
1100     FOLLOWED BY A +1 ONE (OR VICE VERSA) RETURNS NTOT
1200     TIMES THE ORIGINAL DATA. IFORM = 1, 0 OR -1, AS DATA IS
1300     COMPLEX, REAL OR THE FIRST HALF OF A COMPLEX ARRAY. TRANSFORM
1400     VALUES ARE RETURNED TO ARRAY DATA. THEY ARE COMPLEX, REAL OR
1500     THE FIRST HALF OF A COMPLEX ARRAY, AS IFORM = 1, -1 OR 0.
1600     THE TRANSFORM OF A REAL ARRAY (IFORM = 0) DIMENSIONED N(1) BY N(2)
1700     BY ... WILL BE RETURNED IN THE SAME ARRAY, NOW CONSIDERED TO
1800     BE COMPLEX OF DIMENSIONS N(1)/2+1 BY N(2) BY .... NOTE THAT IF
1900     IFORM = 0 OR -1, N(1) MUST BE EVEN, AND ENOUGH ROOM MUST BE
2000     RESERVED. THE MISSING VALUES MAY BE OBTAINED BY COMPLEX CONJUGA-
2100     TION. THE REVERSE TRANSFORMATION, OF A HALF COMPLEX ARRAY DIMEN-
2200     SIONED N(1)/2+1 BY N(2) BY ..., IS ACCOMPLISHED BY SETTING IFORM
2300     TO -1. IN THE N ARRAY, N(1) MUST BE THE TRUE N(1), NOT N(1)/2+1.
2400     THE TRANSFORM WILL BE REAL AND RETURNED TO THE INPUT ARRAY.
2500     RUNNING TIME IS PROPORTIONAL TO NTOT*LOG2(NTOT), RATHER THAN
2600     THE NAIVE NTOT**2. FURTHERMORE, LESS ERROR IS BUILT UP.
2700     WRITTEN BY NORMAN BRENNER OF MIT LINCOLN LABORATORY, JANUARY 1969.
2800     C SEE-- IEEE AUDIO TRANSACTIONS (JUNE 1967), SPECIAL ISSUE ON FFT.
2900     DIMENSION DATA(1), N(1)
3000     NTOT=1
3100     DO 10 IDIM=1,NDIM
3200     10  NTOT=NTOT*N(IDIM)
3300     IF (IFORM) 70,20,20
3400     20  NREM=NTOT
3500     DO 60 IDIM=1,NDIM
3600     60  NREM=NREM/N(IDIM)
3700     NPREV=NTOT/(N(IDIM)*NREM)
3800     NCURR=N(IDIM)
3900     IF (IDIM-1+IFORM) 30,30,40
4000     30  NCURR=NCURR/2
4100     40  CALL BITRV (DATA,NPREV,NCURR,NREM)
4200     CALL COOL2 (DATA,NPREV,NCURR,NREM,ISIGN)
4300     IF (IDIM-1+IFORM) 50,50,60
4400     50  CALL FIXRL (DATA,N(1),NREM,ISIGN,IFORM)
4500     NTOT=(NTOT/N(1))*(N(1)/2+1)
4600     60  CONTINUE
4700     RETURN
4800     70  NTOT=(NTOT/N(1))*(N(1)/2+1)
4900     NREM=1
5000     DO 100 JDIM=1,NDIM
5100     100  IDIM=NDIM+1-JDIM
5200     NCURR=N(IDIM)
5300     IF (IDIM-1) 80,80,90
5400     80  NCURR=NCURR/2
5500     CALL FIXRL (DATA,N(1),NREM,ISIGN,IFORM)
5600     NTOT=NTOT/(N(1)/2+1)*N(1)
5700     90  NPREV=NTOT/(N(IDIM)*NREM)
5800     CALL BITRV (DATA,NPREV,NCURR,NREM)
5900     CALL COOL2 (DATA,NPREV,NCURR,NREM,ISIGN)
6000     100  NREM=NREM*N(IDIM)
6100     RETURN

```



```

6200      END
6300      SUBROUTINE BITRV (DATA,NPREV,N,NREM)
6400      C      SHUFFLE THE DATA BY BIT REVERSAL.
6500      C      DIMENSION DATA(NPREV,N,NREM)
6600      C      COMPLEX DATA
6700      C      EXCHANGE DATA(J1,J4REV,J5) WITH DATA(J1,J4,J5) FOR ALL J1 FROM 1
6800      C      TO NPREV, ALL J4 FROM 1 TO N (WHICH MUST BE A POWER OF TWO), AND
6900      C      ALL J5 FROM 1 TO NREM. J4REV-1 IS THE BIT REVERSAL OF J4-1. E.G.
7000      C      SUPPOSE N = 32. THEN FOR J4-1 = 10011, J4REV-1 = 11001, ETC.
7100      DIMENSION DATA(1)
7200      IPO=2
7300      IP1=IPO*NPREV
7400      IP4=IP1*N
7500      IP5=IP4*NREM
7600      I4REV=1
7700      C      I4REV = 1+(J4REV-1)*IP1
7800      DO 60 I4=1,IP4,IP1
7900      C      I4 = 1+(J4-1)*IP1
8000      IF (I4-I4REV) 10,30,30
8100      10      I1MAX=I4+IP1-IPO
8200      DO 20 I1=I4,I1MAX,IP0
8300      C      I1 = 1+(J1-1)*IPO+(J4-1)*IP1
8400      DO 20 I5=I1,IP5,IP4
8500      C      I5 = 1+(J1-1)*IPO+(J4-1)*IP1+(J5-1)*IP4
8600      I5REV=I4REV+I5-I4
8700      C      I5REV = 1+(J1-1)*IPO+(J4REV-1)*IP1+(J5-1)*IP4
8800      TEMPR=DATA(I5)
8900      TEMPI=DATA(I5+1)
9000      DATA(I5)=DATA(I5REV)
9100      DATA(I5+1)=DATA(I5REV+1)
9200      DATA(I5REV)=TEMPR
9300      20      DATA(I5REV+1)=TEMPI
9400      C      ADD ONE WITH DOWNWARD CARRY TO THE HIGH ORDER BIT OF J4REV-1.
9500      30      IP2=IP4/2
9600      40      IF (I4REV-IP2) 60,60,50
9700      50      I4REV=I4REV-IP2
9800      IP2=IP2/2
9900      IF (IP2-IP1) 60,40,40
10000      60      I4REV=I4REV+IP2
10100      RETURN
10200      END
10300      SUBROUTINE COOL2 (DATA,NPREV,N,NREM,ISIGN)
10400      C      DISCRETE FOURIER TRANSFORM OF LENGTH N. IN-PLACE COOLEY-TUKEY
10500      C      ALGORITHM, BIT-REVERSED TO NORMAL ORDER, SANDE-TUKEY PHASE SHIFTS.
10600      C      DIMENSION DATA(NPREV,N,NREM)
10700      C      COMPLEX DATA
10800      C      DATA(J1,K4,J5) = SUM(DATA(J1,J4,J5)*EXP(ISIGN*2*PI*I*(J4-1)*
10900      C      (K4-1)/N)), SUMMED OVER J4 = 1 TO N FOR ALL J1 FROM 1 TO NPREV,
11000      C      K4 FROM 1 TO N AND J5 FROM 1 TO NREM. N MUST BE A POWER OF TWO.
11100      C      METHOD--LET IPREV TAKE THE VALUES 1, 2 OR 4, 4 OR 8, ..., N/16,
11200      C      N/4, N. THE CHOICE BETWEEN 2 OR 4, ETC., DEPENDS ON WHETHER N IS
11300      C      A POWER OF FOUR. DEFINE IFACT = 2 OR 4, THE NEXT FACTOR THAT
11400      C      IPREV MUST TAKE, AND IREM = N/(IFACT*IPREV). THEN--
11500      C      DIMENSION DATA(NPREV,IPREV,IFACT,IREM,NREM)
11600      C      COMPLEX DATA
11700      C      DATA(J1,J2,K3,J4,J5) = SUM(DATA(J1,J2,J3,J4,J5)*EXP(ISIGN*2*PI*I*
11800      C      (K3-1)*((J3-1)/IFACT+(J2-1)/(IFACT*IPREV)))), SUMMED OVER J3 = 1
11900      C      TO IFACT FOR ALL J1 FROM 1 TO NPREV, J2 FROM 1 TO IPREV, K3 FROM
12000      C      1 TO IFACT, J4 FROM 1 TO IREM AND J5 FROM 1 TO NREM. THIS IS
12100      C      A PHASE-SHIFTED DISCRETE FOURIER TRANSFORM OF LENGTH IFACT.
12200      C      FACTORING N BY FOURS SAVES ABOUT TWENTY FIVE PERCENT OVER FACTOR-

```

```

12300 C      ING BY TWOS. DATA MUST BE BIT-REVERSED INITIALLY.
12400 C      IT IS NOT NECESSARY TO REWRITE THIS SUBROUTINE INTO COMPLEX
12500 C      NOTATION SO LONG AS THE FORTRAN COMPILER USED STORES REAL AND
12600 C      IMAGINARY PARTS IN ADJACENT STORAGE LOCATIONS. IT MUST ALSO
12700 C      STORE ARRAYS WITH THE FIRST SUBSCRIPT INCREASING FASTEST.
12800      DIMENSION DATA(1)
12900      TWOPI=6.2831853072*FLOAT(ISIGN)
13000      IPO=2
13100      IP1=IPO*NPREV
13200      IP4=IP1*N
13300      IP5=IP4*NREM
13400      IP2=IP1
13500 C      IP2=IP1*IPROD
13600      NPART=N
13700 10 IF (NPART-2) 60,30,20
13800 20 NPART=NPART/4
13900      GO TO 10
14000 C      DO A FOURIER TRANSFORM OF LENGTH TWO
14100 30 IF (IP2-IP4) 40,160,160
14200 40 IP3=IP2*2
14300 C      IP3=IP2*IFACT
14400      DO 50 I1=1,IP1,IPO
14500 C      I1 = 1+(J1-1)*IPO
14600      DO 50 I5=I1,IP5,IP3
14700 C      I5 = 1+(J1-1)*IPO+(J4-1)*IP3+(J5-1)*IP4
14800      I3A=I5
14900      I3B=I3A+IP2
15000 C      I3 = 1+(J1-1)*IPO+(J2-1)*IP1+(J3-1)*IP2+(J4-1)*IP3+(J5-1)*IP4
15100      TEMPR=DATA(I3B)
15200      TEMPI=DATA(I3B+1)
15300      DATA(I3B)=DATA(I3A)-TEMPR
15400      DATA(I3B+1)=DATA(I3A+1)-TEMPI
15500      DATA(I3A)=DATA(I3A)+TEMPR
15600 50 DATA(I3A+1)=DATA(I3A+1)+TEMPI
15700      IP2=IP3
15800 C      DO A FOURIER TRANSFORM OF LENGTH FOUR (FROM BIT REVERSED ORDER
15900 60 IF (IP2-IP4) 70,160,160
16000 70 IP3=IP2*4
16100 C      IP3=IP2*IFACT
16200 C      COMPUTE TWOPI THRU WR AND WI IN DOUBLE PRECISION, IF AVAILABLE
16300      THETA=TWOPI/FLOAT(IP3/IP1)
16400      SINTH=SIN(THETA/2.)
16500      WSTPR=-2.*SINTH*SINTH
16600      WSTPI=SIN(THETA)
16700      WR=1.
16800      WI=0.
16900      DO 150 I2=1,IP2,IP1
17000 C      I2 = 1+(J2-1)*IP1
17100      IF (I2-1) 90,90,80
17200 80 W2R=WR*WR-WI*WI
17300      W2I=2.*WR*WI
17400      W3R=W2R*WR-W2I*WI
17500      W3I=W2R*WI+W2I*WR
17600 90 I1MAX=I2+IP1-IPO
17700      DO 140 I1=I2,I1MAX,IPO
17800 C      I1 = 1+(J1-1)*IPO+(J2-1)*IP1
17900      DO 140 I5=I1,IP5,IP3
18000 C      I5 = 1+(J1-1)*IPO+(J2-1)*IP1+(J4-1)*IP3+(J5-1)*IP4
18100      I3A=I5
18200      I3B=I3A+IP2
18300      I3C=I3B+IP2

```



```

18400      I3D=I3C+IP2
18500      C      I3 = 1+(J1-1)*IP0+(J2-1)*IP1+(J3-1)*IP2+(J4-1)*IP3+(J5-1)*IP4
18600      IF (I2-1) 110,110,100
18700      C      APPLY THE PHASE SHIFT FACTORS
18800      100      TEMPR=DATA(I3B)
18900      DATA(I3B)=W2R*DATA(I3B)-W2I*DATA(I3B+1)
19000      DATA(I3B+1)=W2R*DATA(I3B+1)+W2I*TEMPR
19100      TEMPR=DATA(I3C)
19200      DATA(I3C)=WR*DATA(I3C)-WI*DATA(I3C+1)
19300      DATA(I3C+1)=WR*DATA(I3C+1)+WI*TEMPR
19400      TEMPR=DATA(I3D)
19500      DATA(I3D)=W3R*DATA(I3D)-W3I*DATA(I3D+1)
19600      DATA(I3D+1)=W3R*DATA(I3D+1)+W3I*TEMPR
19700      110      TOR=DATA(I3A)+DATA(I3B)
19800      TOI=DATA(I3A+1)+DATA(I3B+1)
19900      T1R=DATA(I3A)-DATA(I3B)
20000      T1I=DATA(I3A+1)-DATA(I3B+1)
20100      T2R=DATA(I3C)+DATA(I3D)
20200      T2I=DATA(I3C+1)+DATA(I3D+1)
20300      T3R=DATA(I3C)-DATA(I3D)
20400      T3I=DATA(I3C+1)-DATA(I3D+1)
20500      DATA(I3A)=TOR+T2R
20600      DATA(I3A+1)=TOI+T2I
20700      DATA(I3C)=TOR-T2R
20800      DATA(I3C+1)=TOI-T2I
20900      IF (ISIGN) 120,120,130
21000      120      T3R=-T3R
21100      T3I=-T3I
21200      130      DATA(I3B)=T1R-T3I
21300      DATA(I3B+1)=T1I+T3R
21400      DATA(I3D)=T1R+T3I
21500      140      DATA(I3D+1)=T1I-T3R
21600      TEMPR=WR
21700      WR=WSTPR*TEMPR-WSTPI*WI+TEMPR
21800      150      WI=WSTPR*WI+WSTPI*TEMPR+WI
21900      IP2=IP3
22000      GO TO 60
22100      160      RETURN
22200      END
22300      SUBROUTINE FIXRL (DATA,N,NREM,ISIGN,IFORM)
22400      C      FOR IFORM = 0, CONVERT THE TRANSFORM OF A DOUBLED-UP REAL ARRAY,
22500      C      CONSIDERED COMPLEX, INTO ITS TRUE TRANSFORM. SUPPLY ONLY THE
22600      C      FIRST HALF OF THE COMPLEX TRANSFORM, AS THE SECOND HALF HAS
22700      C      CONJUGATE SYMMETRY. FOR IFORM = -1, CONVERT THE FIRST HALF
22800      C      OF THE TRUE TRANSFORM INTO THE TRANSFORM OF A DOUBLED-UP REAL
22900      C      ARRAY. N MUST BE EVEN.
23000      C      USING COMPLEX NOTATION AND SUBSCRIPTS STARTING AT ZERO, THE
23100      C      TRANSFORMATION IS--
23200      C      DIMENSION DATA(N,NREM)
23300      C      ZSTP = EXP(ISIGN*2*PI*I/N)
23400      C      DO 10 I2=0,NREM-1
23500      C      DATA(0,I2) = CONJ(DATA(0,I2))*(1+I)
23600      C      DO 10 I1=1,N/4
23700      C      Z = (1+(2*IFORM+1)*I*ZSTP**I1)/2
23800      C      I1CNJ = N/2-I1
23900      C      DIF = DATA(I1,I2)-CONJ(DATA(I1CNJ,I2))
24000      C      TEMP = Z*DIF
24100      C      DATA(I1,I2) = (DATA(I1,I2)-TEMP)*(1-IFORM)
24200      C 10      DATA(I1CNJ,I2) = (DATA(I1CNJ,I2)+CONJ(TEMP))*(1-IFORM)
24300      C      IF I1=I1CNJ, THE CALCULATION FOR THAT VALUE COLLAPSES INTO
24400      C      A SIMPLE CONJUGATION OF DATA(I1,I2).

```

```

24500      DIMENSION DATA(1)
24600      TWOPI=6.283185307*FLOAT(ISIGN)
24700      IPO=2
24800      IP1=IPO*(N/2)
24900      IP2=IP1*NREM
25000      IF (IFORM) 10,70,70
25100 C      PACK THE REAL INPUT VALUES (TWO PER COLUMN)
25200      10  J1=IP1+1
25300          DATA(2)=DATA(J1)
25400          IF (NREM-1) 70,70,20
25500      20  J1=J1+IPO
25600          I2MIN=IP1+1
25700          DO 60 I2=I2MIN,IP2,IP1
25800          DATA(I2)=DATA(J1)
25900          J1=J1+IPO
26000      IF (N-2) 50,50,30
26100      30  I1MIN=I2+IPO
26200          I1MAX=I2+IP1-IPO
26300          DO 40 I1=I1MIN,I1MAX,IPO
26400          DATA(I1)=DATA(J1)
26500          DATA(I1+1)=DATA(J1+1)
26600      40  J1=J1+IPO
26700      50  DATA(I2+1)=DATA(J1)
26800      60  J1=J1+IPO
26900      70  DO 80 I2=1,IP2,IP1
27000          TEMPR=DATA(I2)
27100          DATA(I2)=DATA(I2)+DATA(I2+1)
27200      80  DATA(I2+1)=TEMPR-DATA(I2+1)
27300          IF (N-2) 200,200,90
27400      90  THETA=TWOPI/FLOAT(N)
27500          SINTH=SIN(THETA/2.)
27600          ZSTPR=-2.*SINTH*SINTH
27700          ZSTPI=SIN(THETA)
27800          ZR=(1.-ZSTPI)/2.
27900          ZI=(1.+ZSTPR)/2.
28000          IF (IFORM) 100,110,110
28100      100 ZR=1.-ZR
28200          ZI=-ZI
28300      110 I1MIN=IPO+1
28400          I1MAX=IPO*(N/4)+1
28500          DO 190 I1=I1MIN,I1MAX,IPO
28600          DO 180 I2=I1,IP2,IP1
28700          I2CNJ=IPO*(N/2+1)-2*I1+I2
28800          IF (I2-I2CNJ) 150,120,120
28900      120 IF (ISIGN*(2*IFORM+1)) 130,140,140
29000      130 DATA(I2+1)=-DATA(I2+1)
29100      140 IF (IFORM) 170,180,180
29200      150 DIFR=DATA(I2)-DATA(I2CNJ)
29300          DIFI=DATA(I2+1)+DATA(I2CNJ+1)
29400          TEMPR=DIFR*ZR-DIFI*ZI
29500          TEMPI=DIFR*ZI+DIFI*ZR
29600          DATA(I2)=DATA(I2)-TEMPR
29700          DATA(I2+1)=DATA(I2+1)-TEMPI
29800          DATA(I2CNJ)=DATA(I2CNJ)+TEMPR
29900          DATA(I2CNJ+1)=DATA(I2CNJ+1)-TEMPI
30000          IF (IFORM) 160,180,180
30100      160 DATA(I2CNJ)=DATA(I2CNJ)+DATA(I2CNJ)
30200          DATA(I2CNJ+1)=DATA(I2CNJ+1)+DATA(I2CNJ+1)
30300      170 DATA(I2)=DATA(I2)+DATA(I2)
30400          DATA(I2+1)=DATA(I2+1)+DATA(I2+1)
30500      180 CONTINUE

```



```

30600      TEMPR=ZR-.5
30700      ZR=ZSTPR*TEMPR-ZSTPI*ZI+ZR
30800      190  ZI=ZSTPR*ZI+ZSTPI*TEMPR+ZI
30900      C    RECURSION SAVES TIME, AT A SLIGHT LOSS IN ACCURACY.  IF AVAILABLE,
31000      C    USE DOUBLE PRECISION TO COMPUTE ZR AND ZI.
31100      200  IF (IFORM) 270,210,210
31200      C    UNPACK THE REAL TRANSFORM VALUES (TWO PER COLUMN)
31300      210  I2=IP2+1
31400          I1=I2
31500          J1=IPO*(N/2+1)*NREM+1
31600          GO TO 250
31700      220  DATA(J1)=DATA(I1)
31800          DATA(J1+1)=DATA(I1+1)
31900          I1=I1-IPO
32000          J1=J1-IPO
32100      230  IF (I2-I1) 220,240,240
32200      240  DATA(J1)=DATA(I1)
32300          DATA(J1+1)=0.
32400      250  I2=I2-IP1
32500          J1=J1-IPO
32600          DATA(J1)=DATA(I2+1)
32700          DATA(J1+1)=0.
32800          I1=I1-IPO
32900          J1=J1-IPO
33000          IF (I2-I1) 260,260,230
33100      260  DATA(2)=0.
33200      270  RETURN
33300      END

```

APPENDIX D.

This appendix contains plots of 10 drops using advective field V01 ($\hat{\beta} = 0$) and 10 drops using advective field V02 ($\hat{\beta} = 10$). These are marked V01H01 through V01H10 and V02H01 through V02H10, respectively.

V01 H01

 $\hat{\beta} = 0$

V01 H02

V01 H03

$$\hat{\beta} = 0$$

V01 H04

V01 H05

V01 H06

$$\hat{\beta} = 0$$

V01 H07

$$\hat{\beta} = 0$$

V01 H08

V01 H09

$$\hat{\beta} = 0$$

V01 H10

$$\hat{\beta} = 0$$

V02 H01

$$\hat{\beta} = 10$$

V02 H02

V02 H03

$$\hat{\beta} = 10$$

V02 H04

V02 H05

$$\hat{\beta} = 10$$

V02 H06

$$\hat{\beta} = 10$$

V02 H07

V02 H08

$$\beta = 10$$

V02 H09

V02 H10

$$\hat{\beta} = 10$$

REPORT DOCUMENTATION PAGE	1. REPORT NO. WHOI-84-24	2.	3. Recipient's Accession No.
4. Title and Subtitle Models and Simulations of Tracer Dispersal by Mid-Ocean Mesoscale Eddies			5. Report Date May 1984
7. Author(s) Dale B. Haidvogel and Thomas Keffer			6.
9. Performing Organization Name and Address Woods Hole Oceanographic Institution Woods Hole, Massachusetts 02543			8. Performing Organization Rept. No. WHOI-84-24
12. Sponsoring Organization Name and Address National Science Foundation			10. Project/Task/Work Unit No.
			11. Contract(C) or Grant(G) No. (C) OCE81-09486 (G)
15. Supplementary Notes This report should be cited as: Woods Hole Oceanog. Inst. Tech Rept. WHOI-84-24.			13. Type of Report & Period Covered Technical
			14.
16. Abstract (Limit: 200 words) Haidvogel and Keffer (1984) have examined the stirring and mixing of a passive tracer in a turbulent mesoscale eddy field for environmental parameters characteristic of a homogeneous mid-ocean region. This study utilized two numerical models, based respectively on the barotropic vorticity and advective-diffusive equations, to simulate the evolving mid-ocean eddy field and the associated dispersion of an embedded tracer spot. Here, these models are described in detail and the full suite of tracer release simulations are shown in the form of contour plots. (<u>Dyn. Atm. Oceans</u> , 8, 1 - 40.)			
17. Document Analysis a. Descriptors 1. tracers 2. modelling 3. ocean circulation b. Identifiers/Open-Ended Terms c. COSATI Field/Group			
18. Availability Statement: Approved for publication; distribution unlimited.		19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 69
		20. Security Class (This Page)	22. Price

DOCUMENT LIBRARY

August 3, 1984

DISTRIBUTION LIST FOR TECHNICAL REPORT EXCHANGE

Institute of Marine Sciences Library
University of Alaska
O'Neill Building
905 Koyukuk Ave. North
Fairbanks, AK

Attn: Stella Sanchez-Wade
Documents Section
Scripps Institution of Oceanography
Library, Mail Code C-075C
La Jolla, CA 92093

Hancock Library of Biology & Oceanography
Alan Hancock Laboratory
University of Southern California
Los Angeles, CA 90007

Gifts & Exchanges
Library
Bedford Institute of Oceanography
P.O. Box 1006
Dartmouth, NS, B2Y 4A2, CANADA

Office of the International
Ice Patrol
c/o Coast Guard R&D Center
Avery Point
Groton, CT 06340

Library
Physical Oceanographic Laboratory
Nova University
8000 N. Ocean Drive
Dania, FL 33304

NOAA/EDIS Miami Library Center
4301 Rickenbacker Causeway
Miami, FL 33149

Library
Skidaway Institute of Oceanography
P.O. Box 13687
Savannah, GA 31406

Institute of Geophysics
University of Hawaii
Library Room 252
2525 Correa Road
Honolulu, HI 96822

Library
Chesapeake Bay Institute
4800 Atwell Road
Shady Side, MD 20876

MIT Libraries
Serial Journal Room 14E-210
Cambridge, MA 02139

Director, Ralph M. Parsons Laboratory
Room 48-311
MIT
Cambridge, MA 02139

Marine Resources Information Center
Bldg. E38-320
MIT
Cambridge, MA 02139

Library
Lamont-Doherty Geological Observatory
Columbia University
Palisades, NY 10964

Library
Serials Department
Oregon State University
Corvallis, OR 97331

Pell Marine Science Library
University of Rhode Island
Narragansett Bay Campus
Narragansett, RI 02882

Working Collection
Texas A&M University
Dept. of Oceanography
College Station, TX 77843

Library
Virginia Institute of Marine Science
Gloucester Point, VA 23062

Fisheries-Oceanography Library
151 Oceanography Teaching Bldg.
University of Washington
Seattle, WA 98195

Library
R.S.M.A.S.
University of Miami
4600 Rickenbacker Causeway
Miami, FL 33149