Three-Dimensional Time Dependent Computation of Turbulent Flow

by

D. Kwak
W. C. Reynolds
and
J. H. Ferziger

Prepared from work done under Grant
NASA-NgR-05-020-622

Report No. TF-5

Thermosciences Division
Department of Mechanical Engineering
Stanford University
Stanford, California

May 1975

135

THREE-DIMENSIONAL TIME DEPENDENT COMPUTATION OF TURBULENT FLOW

by

D. Kwak, W. C. Reynolds, and J. H. Ferziger

Prepared from work done under Grant
NASA-NgR-05-020-622

Report No. TF-5

Thermosciences Division
Department of Mechanical Engineering
Stanford University
Stanford, California

May, 1975

ABSTRACT

The three-dimensional, primitive equations of motion have been solved numerically for the case of isotropic box turbulence and the distortion of homogeneous turbulence by irrotational plane strain at large Reynolds numbers. A Gaussian filter was applied to governing equations to define the large scale field. This gives rise to additional second order computed scale stresses (Leonard stresses). The residual stresses are simulated through an eddy viscosity. 16x16x16 and 32x32x32 uniform grids were used, with a fourth order differencing scheme in space and a second order Adams-Bashforth predictor for explicit time stepping. The results were compared to the experiments and statistical information was extracted from the computer generated data.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the useful comments of Drs. A. Leonard, M. Rubesin, H. Lomax and Mr. S. Shaanan. We appreciate Ms Nancy Silvis for the speedy typing of the final copy.

This work was sponsored by NASA Ames Research Center under Grant NASA-NgR-05-020-622.

TABLE OF CONTENTS

Page

LIST OF FIGURES

NOMENCLATURE

A,B	=	random unit vectors
A_o	=	constant in Rotta's model
C_s	=	constant in the Smagorinsky model
C_v	=	constant in the vorticity model
D	=	finite difference form of divergence operator
D_{ij}	$\equiv$	$2\nu \langle \frac{\partial u_i}{\partial x_k} \frac{\partial u_j}{\partial x_k} \rangle$
E	=	three-dimensional energy spectrum
E_{11}	=	one-dimensional energy spectrum
$\bar{E}$, $\bar{E}_{11}$	=	filtered spectra
$\hat{e}_x$, $\hat{e}_y$, $\hat{e}_z$	=	unit vecotrs in x-, y-, and z-direction
F_i	=	i-component of forcing term due to the mean strain
$\tilde{F}_i$	=	finite difference form of F_i
f	=	a general field variable; $\bar{f}$ = filtered field component; f' = residual field component
G	=	filter function; finite difference form of gradient operator
g_i	=	RHS of Poisson equation for pressure; $\tilde{g}_i$ finite difference form
H	=	heaviside step function
K_1	$\equiv$	$(\langle \overline{v}^2\rangle - \langle \overline{u}^2\rangle)/(\langle \overline{v}^2\rangle + \langle \overline{u}^2\rangle)$, structural parameter
k_i	=	wave number in i-direction
k_i'	=	modified wave number in i-direction (fourth order); k_i'' (second order)
$\underset{\sim}{k}^2$	=	finite difference form of ∇^2 operator
L	$\equiv$	$N\Delta$, computational box size; length scale of large eddies
M	=	grid size of turbulence generator in experiments
N	=	number of finite difference mesh points in one direction
Nc	$\equiv$	$\Delta t \sqrt{q^3/3}\,/\Delta$ for isotropic turbulence;
	$\equiv$	$\Delta t \sqrt{U_{max} + q^3/3}\,/\Delta$ for straining flow
P	$\equiv$	$p + \frac{1}{3} R_{ii}$; imposed pressure field

$\mathcal{P}$	=	production term
$\mathcal{P}_L$	=	Leonard production term
p	=	pressure
q^2	$\equiv$	$\overline{u}_i\overline{u}_i$
R	=	degree of anisotropy
R_T	$\equiv$	qL/ν
R_{ij}	$\equiv$	$\overline{u_i'u_j'} + \overline{u_i'\overline{u}_j} + \overline{u_j'\overline{u}_i}$, residual stress
S_{ij}	$\equiv$	$\frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)$
$\mathcal{S}_{ij}$	$\equiv$	$\frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)$
T_{ij}	$\equiv$	$\langle p\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\rangle$; $(T_{ij})_c$ = computed; $(T_{ij})_m$ = Rotta's model; $(T_{ij})_R$ obtained from R_{ij} history
t	=	time
U_o	=	free stream velocity in x-direction
U_i	=	i-components imposed mean velocity
u_i	=	i-components fluctuating velocity; $\overline{u}_i$ = filtered field component; u_i' = residual field component; $\hat{u}_i$ = Fourier transformed u_i ; $\underline{u}$ = vector notation
δ/δ_x, δ/δ_t	=	finite difference form of $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial t}$
ε	=	energy dissipation; ε_R = residual fields contribution; ε_L = Leonard terms contribution
Φ	=	spectrum function
ϕ,θ	=	random angle
ν	=	μ/ρ , kinematic viscosity
ν_T	=	eddy viscosity
ρ	=	fluid density
Γ	=	constant strain-rate
γ	=	constant in Gaussian filter function
η	=	Kolmogorov microscale
$\overline{\omega}$	=	filtered vorticity, curl $\overline{u}$
Δ_A	=	averaging length scale

Δ	=	computational mesh width
$< \ >$	=	ensemble average in real space; shell average in wave-number space

Superscripts

(n)	=	time step

Subscripts

(k,ℓ,m)	=	computational mesh number index

CHAPTER I

INTRODUCTION

1.1 Historical Background

As computer capabilities grow, the three-dimensional time-dependent computation of turbulence is becoming possible. However, the retention of all scales of motion is not yet feasible (and probably never will be), so the best one can hope for is the simulation of the large scale structures. The large scale structures are strongly dependent upon the nature of the flow, but there is considerable evidence that the structure of the smaller scales is independent of the large scale structure. This suggests a mixed approach in which one computes the large scale motions and models the small scales.

To define the large scale motions, some sort of averaging operator has to be applied to the governing equations to filter out the small scale motions. However, in three-dimensional computations to date, the definition of the small scale motions was not precisely related to the filtering operation and consequently the meaning of those motions was not very clear. In any case, the resulting equations for the filtered field contain so-called sub-grid scale or residual scale Reynolds stresses, which must be modeled in the computation.

Two distinctive solution methods have been used in solving the resulting equations. The first is a conventional finite difference mesh calculation. In this approach, the simplest and perhaps the most usual way of relating the residual scale turbulence to the filtered motion is by a local eddy viscosity model. Smagorinsky (1963, 1965) related eddy viscosity to the local strain-rate of the filtered field. Deardorff (1970a,b) applied this model to three-dimensional turbulent channel flow and planetary boundary layer problems, and Schumann (1973) applied it to plane channels and annuli. Deardorff (1973) and Schumann later introduced more sophisticated residual Reynolds stress transport equations. Other examples of this approach are given by Lilly (1964, 1967), Smagorinsky et al (1965), Fox and Lilly (1972), Fox and Deardorff (1972). Previous work mentioned above has not paid sufficient attention to the basic aspects of this type of simulation, so as yet this approach has not really progressed very far.

The second approach is the spectral (Fourier) method much advocated by Orszag (1969, 1971a,b). While this method has mathematically attractive features for certain problems, it is generally more difficult to extend to flows with interesting geometries. Moreover, work to date ignored the residual Reynolds stresses, and it is not clear how these could be incorporated in a Fourier calculation.

1.2 Motivation and Objectives

At the time this work was initiated there were some serious problems with work done previously:

(1) There was a need to define the large-scale field precisely, so that the equations can be systematically developed.

(2) There was a need to carefully evaluate the accuracy requirements to be sure that computational errors are higher order than the residual stresses.

(3) There was a need to carefully assess just what can really be learned from this type of turbulence simulation.

The main objective of this study was to carefully develop a numerical simulation method for turbulent flows away from solid or free boundaries, and to apply this method to study decaying isotropic turbulence and homogeneous turbulence with irrotational plane strain.

The present study is one in a systematic program investigating large eddy simulation of turbulence, and reports the details of the initial computations under this program.

1.3 Summary

The contribution of the present work includes

(a) a precise definition for the large-scale field (after Leonard (1973)),

(b) a study of the optimum averaging scale, as compared to the grid mesh scale,

(c) a study of two residual stress models, and evaluation of the model constant for each,

(d) a demonstration that the model constant is independent of mesh size,

(e) a fourth-order differencing scheme that properly conserves energy and momentum,

(f) a method for calculating the pressure so as to conserve mass at subsequent time steps,

(g) a demonstration that a coarse mesh can be used to obtain surprisingly good predictions for the Reynolds stresses in a straining flow,

(h) an evaluation of certain aspects of simple turbulence closure models.

Although many questions have been answered by this work, new ones have been raised. Suggestions for follow-on work in this project are made in Chapter VI.

Chapter II

Theoretical Foundations

2.1 Definition of the Filtered and Residual Fields

To resolve the smallest scales of turbulence in a grid-based calculation, the mesh size has to be smaller than the dissipation length, which is on the order of the Kolmogorov microscale, $\eta = (\nu^3/\varepsilon)^{1/4}$. Here ν is the kinematic viscosity and ε is the energy dissipation rate per unit mass. It is known that (see Tennekes and Lumley 1972) $\varepsilon \simeq q^3/L$ where q is the R.M.S. velocity and L is the length scale of large eddies. Thus the minimum number of mesh points that must be used in a three-dimensional grid computation that resolves both the large and small scales can be estimated as

$$N = \left(\frac{L}{\eta}\right)^3 = \left(\frac{qL}{\nu}\right)^{9/4} = R_T^{\,9/4} .$$

Using this estimate, we find that an R_T of 10^3, typical of turbulent flows, would require 2×10^7 worlds of storage for four variables. This is approximately 50 times the Large Core Memory of the CDC 7600, and about 10 times the available memory of the ILLIAC IV disk.

It is clear that one can not do a full simulation, except at extremely low Reynolds number. The best one can hope for is a computation that will yield the large-scale motions. Fortunately these contain most of the turbulence energy, and are responsible for most of the turbulent transport, and so a large-eddy simulation technique would be very useful, especially if it could handle arbitrary flows.

The first problem one faces is in defining the large scale field. Conventionally the large scale motions have been defined by volume-averaging in a continuous manner over computational grid boxes (e.g. Deardorff 1970a). Schumann (1973, 1974) applied a slightly different technique involving averaging over the surface of grid boxes.

A more general approach that recognizes the continuous nature of the flow variables is the "filter function" approach of Leonard (1973). Let $f(\underline{x})$ denote a field variable, for example velocity. f may contain large and very small components. Then, we define the filtered field $\overline{f}$ by

$$\overline{f}(\underline{x}) = \int G(\underline{x}-\underline{x}')\ f(\underline{x}')\ d\underline{x}' \tag{2.1a}$$

where G is a selected filter function. For $\overline{C} = C$, where C is a constant, the filter G must satisfy

$$\int G(\underline{x})\ d\underline{x} = 1 \tag{2.1b}$$

Now f can be decomposed into its filtered field (FF) component, $\overline{f}$, and residual-field (RF) components, f', by

$$f = \overline{f} + f' \tag{2.2}$$

Note that $\overline{f}$ is <u>not</u> the conventional mean used in the classic-turbulence literature.

In the present work we treat only flows that are homogeneous, for which the integration in (2.1) extends over all space. Careful consideration will have to be given to the domain of integration when one desires to treat a flow near a wall.

Now note that, if G is piecewise continuously differentiable and $G(r)$ goes to zero as $r \to \infty$ at least as fast as $1/r^4$,

$$\begin{aligned}
\overline{\frac{\partial f}{\partial \underline{x}}} &= \int_{-\infty}^{\infty} G(\underline{x}-\underline{x}')\ \frac{\partial f}{\partial \underline{x}'}\ d\underline{x}' \\
&= \underset{0}{\cancel{fG}}\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} f(\underline{x}')\ \frac{\partial}{\partial \underline{x}'}\ G(\underline{x}-\underline{x}')\ d\underline{x}' \\
&= \frac{\partial}{\partial \underline{x}} \int_{-\infty}^{\infty} G(\underline{x}-\underline{x}')\ f(\underline{x}')\ d\underline{x}' \\
&= \frac{\partial}{\partial \underline{x}}\ \overline{f}
\end{aligned} \tag{2.3a}$$

Also,

$$\overline{\frac{\partial f}{\partial t}} = \frac{\partial \overline{f}}{\partial t} \tag{2.3b}$$

However,

$$\overline{\bar{f}\bar{g}} \neq \bar{f}\bar{g} \tag{2.3c}$$

2.2 The Dynamical Equations

Applying (2.1) to the Navier-Stokes equations, and using (2.2) and (2.3), one obtains (for incompressible flows)

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{2.4}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j} \overline{u_i u_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \nu\nabla^2 \bar{u}_i \tag{2.5}$$

The advection term is

$$\begin{aligned} \overline{u_i u_j}\ (\underline{x}_0,t) &= \overline{\bar{u}_i\bar{u}_j} + \overline{u_i'\bar{u}_j} + \overline{\bar{u}_i u_j'} + \overline{u_i'u_j'} \\ &= \overline{\bar{u}_i\bar{u}_j} + R_{ij} \end{aligned} \tag{2.6}$$

where

$$R_{ij} = \overline{u_i'u_j'} + \overline{u_i'\bar{u}_j} + \overline{\bar{u}_i u_j'}$$

R_{ij} is the residual field contribution to the advection term. $-\rho R_{ij}$ is called the "residual stress."

To localize the first term on the right in (2.6) we carry out a Taylor series expansion,

$$\begin{aligned} \overline{\bar{u}_i\bar{u}_j}\ (\underline{x}_o,t) &= \int G(\underline{x}_o-\underline{x})\ \bar{u}_i\bar{u}_j(\underline{x})\ d\underline{x} \\ &= \int_{-\infty}^{\infty} \Big\{ \bar{u}_i(\underline{x}_o,t) + \frac{\partial \bar{u}_i}{\partial x_k}(x_k-x_{ko}) + \frac{1}{2}\frac{\partial^2 \bar{u}_i}{\partial x_k \partial x_\ell}(x_k-x_{ko})(x_\ell-x_{\ell o}) \\ &\quad + O(x-x_o)^3 \Big\} \cdot \Big\{ \bar{u}_j\ (\underline{x}_o,t) + \frac{\partial \bar{u}_j}{\partial x_k}(x_k-x_{ko}) \\ &\quad + \frac{1}{2}\frac{\partial^2 \bar{u}_j}{\partial x_k \partial x_\ell}(x_k-x_{ko})(x_\ell-x_{\ell o}) + O(x-x_o)^3 \Big\}\ G(\underline{x}_o-\underline{x})\ d\underline{x} \end{aligned} \tag{2.7}$$

For the above filtering of the dynamical equations to be useful, the integrals in (2.7) must exist. This requires that $G(r) \to 0$ exponentially as $r \to \infty$.

2.3 Filter Selection

(a) Sub-Grid-Scale filter

Let us first seek a filter that makes the scales of motion in the residual field smaller than the scales in the filtered field, in the Fourier sense. Let

$$f = \int_{-\infty}^{\infty} \hat{f}(\underline{k})\, e^{i\underline{k}\cdot\underline{x}}\, d\underline{k} \tag{2.8}$$

Then,

$$\bar{f} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{f}(\underline{k})\, G(\underline{x}-\underline{x}')\, e^{i\underline{k}\cdot\underline{x}'}\, d\underline{x}'\, d\underline{k} \tag{2.9}$$

We want to have $\bar{f}$ contain all scales larger than a cut-off scale. Thus we want

$$\bar{f} = \int_{-k_c}^{+k_c} \hat{f}(\underline{k})\, e^{i\underline{k}\cdot\underline{x}}\, d\underline{k} \tag{2.10}$$

where k_c is the cut-off wave number. Hence, we can write

$$\int_{-\infty}^{\infty} H(k)\, \hat{f}(\underline{k})\, e^{i\underline{k}\cdot\underline{x}}\, d\underline{k} = \int_{-\infty}^{\infty} \hat{f}(\underline{k}) \int_{-\infty}^{\infty} G(\underline{x}-\underline{x}')\, e^{i\underline{k}\cdot\underline{x}'}\, d\underline{x}'\, d\underline{k} \tag{2.11a}$$

where

$$H(k;k_c) = \begin{cases} 0 & \text{if } k_i > k_c \text{ for any } i \\ 1 & \text{otherwise} \end{cases} \tag{2.11b}$$

So we have an integral equation for G ,

$$H(k;k_c) = \int_{-\infty}^{\infty} G(\underline{x}-\underline{x}')\, e^{ik(\underline{x}'-\underline{x})}\, d\underline{x}' \tag{2.12}$$

The solution to (2.12) is

$$G(\underline{x}-\underline{x}') = \prod_{i=1}^{3} \frac{\sin \pi(x_i-x_i')/\Delta_A}{\pi(x_i-x_i')} \tag{2.13}$$

where $\Delta_A = \pi/k_c$ is the averaging or filtering length scale. This is the proper filter if one wishes to have the residual field really be "sub-grid scale." A grid-based computation made using this filter would be equivalent to a Fourier computation.

The second moment of G involves integrals like

$$\int_{-\infty}^{\infty} \frac{x^2 \sin(\pi x/\Delta_A)}{\pi x}\, dx \tag{2.14}$$

This integral does not exist, and hence the expansion (2.7) could not be used. This filter is not suitable for a grid-based numerical method; hence one can not expect to really have the residual field be sub-grid scale.

(b) Top-hat filter

The filter used implicitly by many workers is the top-hat,

$$G(\underline{x}-\underline{x}') = \begin{cases} 1/\Delta_A & \text{for } |\underline{x}-\underline{x}'| < \frac{\Delta_A}{2} \\ 0 & |\underline{x}-\underline{x}'| \geq \frac{\Delta_A}{2} \end{cases} \tag{2.15}$$

Then the filtered velocity is

$$\bar{\underline{u}}(\underline{x}) = \frac{1}{\Delta_A^{\,3}} \int_{-\Delta_{A/2}}^{\Delta_{A/2}} \underline{u}(\underline{x}+\underline{\xi})\, d\underline{\xi} \tag{2.16}$$

This is equivalent to volume averaging. The Fourier transform of (2.16) is

$$\begin{aligned} \hat{\bar{\underline{u}}}(\underline{k}) &= \frac{1}{\Delta_A^{\,3}} \int_{-\infty}^{\infty} \int_{-\Delta_{A/2}}^{\Delta_{A/2}} \underline{u}(\underline{x}+\underline{\xi})\, e^{-\underline{k}\cdot\underline{x}}\, d\underline{x}\, d\underline{\xi} \\ &= \frac{1}{\Delta_A^{\,3}} \int_{-\Delta_{A/2}}^{+\Delta_{A/2}} \hat{\underline{u}}(\underline{k})\, e^{i\underline{k}\cdot\underline{\xi}}\, \underline{d\xi} \\ &= \left\{ \prod_{i=1}^{3} \frac{\sin(k_i \Delta_{A/2})}{k_i\, \Delta A/2} \right\} \hat{\underline{u}}(\underline{k}) \end{aligned} \tag{2.17}$$

Here $\hat{\underline{u}}(\underline{k})$ is the Fourier transform of $\underline{u}$.

Equation (2.17) shows that the spectrum of the filtered field will contain components of all wave-numbers. Moreover, at the wave-numbers for which the coefficient of $\underline{\hat{u}}(\underline{k})$ in (2.17) is zero, the inverse transform will be singular. This makes it impossible to predict the actual spectrum $\underline{\hat{u}}(\underline{k})$ from the filtered spectrum $\hat{\bar{\underline{u}}}(\underline{k})$, and this very undesirable feature of the top-hat filter renders it useless if we want to compute spectral features with a grid-based method. However, the top-hat filter could be used if spectral results were not sought.

Using (2.15) in (2.7), and carrying out the integration over $\underline{x}$,

$$\overline{\bar{u}_i\bar{u}_j}\,(\underline{x}_o,t) = \bar{u}_i\bar{u}_j(\underline{x}_o,t) + \frac{\Delta_A^2}{24}\nabla^2(\bar{u}_i\bar{u}_j) + O(\Delta_A^4) \qquad (2.18)$$

The second term on the right is called the Leonard term; $-\rho\Delta_A^2\,\nabla^2(\bar{u}_i\bar{u}_j)/24$ is called the "Leonard stress." As will be shown later, $R_{ij} = O(\Delta_A^2)$ and $\Delta_A = O(\Delta)$. So both the residual stresses and the Leonard stresses have to be included; moreover, the computational difference scheme must be accurate to $O(\Delta^2)$ to avoid introduction of numerical errors comparable with these stresses.

(c) <u>Gaussain filter</u>

A filter with much more desirable properties is

$$G(\underline{x}-\underline{x}') = \left\{\sqrt{\frac{\gamma}{\pi}}\,\frac{1}{\Delta_A}\right\}^3 \exp\left\{-\gamma(\underline{x}-\underline{x}')^2/\Delta_A^2\right\} \qquad (2.19)$$

where γ is a constant. Then the filtered velocity is

$$\bar{\underline{u}}(\underline{x}) = \left(\sqrt{\frac{\gamma}{\pi}}\,\frac{1}{\Delta_A}\right)^3 \int_{-\infty}^{\infty} \underline{u}(\underline{x}')\, e^{-\gamma(\underline{x}-\underline{x}')^2/\Delta_A^2}\, d\underline{x}' \qquad (2.20)$$

The Fourier transform of this is

$$\hat{\bar{\underline{u}}}(\underline{x}) = \left(\sqrt{\frac{\gamma}{\pi}}\,\frac{1}{\Delta_A}\right)^3 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \underline{u}(\underline{x}+\underline{\xi})\, e^{-i\underline{k}\cdot\underline{x}}\, e^{-\gamma\, \xi^2/\Delta_A^2}\, d\underline{x}\, d\underline{\xi}$$

$$= \left(\sqrt{\frac{\gamma}{\pi}}\,\frac{1}{\Delta_A}\right)^3 \int_{-\infty}^{\infty} \underline{\hat{u}}(\underline{k})\, e^{i\underline{k}\cdot\underline{\xi}}\, e^{-\gamma\, \xi^2/\Delta_A^2}\, d\underline{\xi}$$

$$= \left(\sqrt{\frac{\gamma}{\pi}}\,\frac{1}{\Delta_A}\right)^3 \underline{\hat{u}}(\underline{k})\, \exp\left\{-\frac{1}{4}\,\frac{\Delta_A^2}{\gamma}\,(k_1^2 + k_2^2 + k_3^3)\right\}$$

$$= \underline{\hat{u}}(\underline{k})\, \exp\left(-\frac{\Delta_A^2}{4\gamma}\, k^2\right) \tag{2.21}$$

Consider now the three-dimensional energy spectra of the actual and filtered fields:

$$E(k) = 2\pi\, k^2 < \underline{\hat{u}}(\underline{k}) \cdot \underline{\hat{u}}^*(\underline{k})> \tag{2.22a}$$

$$\bar{E}(k) = 2\pi\, k^2 < \hat{\bar{\underline{u}}}(\underline{k}) \cdot \hat{\bar{\underline{u}}}^*(\underline{k})> \tag{2.22b}$$

Here < > denotes an average over an ensemble of experiments, and * denotes a complex conjugate. Equations (2.21) and (2.22) show that

$$\bar{E}(k) = E(k)\, \exp\left(-\frac{\Delta_A^2}{2\gamma}\, k^2\right) \tag{2.23}$$

We see that the use of the Gaussian filter will result in a filtered field that misses only a very small amount of large scale motion; most of the small scale motions are placed in the residual field. Thus, in many respects this filter has the desirable properties of the sub-grid-scale filter. However, its behavior at $r \to \infty$ makes the integrals in (2.7) exist. Moreover, the conversion back and forth between the spectrum of the filtered field and the spectrum of the actual field is easily accomplished, and hence the Gaussian filter is perferable to the top-hat filter.

Using (2.19) and (2.7), one obtains

$$\overline{\bar{u}_i\bar{u}_j}\ (\underline{x}_o,t) = \bar{u}_i\bar{u}_j(\underline{x}_o,t) + \frac{\Delta_A^2}{4\gamma}\nabla^2(\bar{u}_i\bar{u}_j) + O(\Delta_A^4) \qquad (2.24)$$

When $\gamma = 6$, the Leonard term in (2.24) is exactly the same as in (2.18). Hence the Gaussian filter with $\gamma = 6$ was chosen for the present study. This filter is illustrated on Fig. 2.1 and an example of $\bar{E}$ and E relation (2.23) is shown on Fig. 2.2.

2.4 Residual Stress Models

The following eddy viscosity model is used for R_{ij} .

$$R_{ij} = \frac{1}{3} R_{kk}\,\delta_{ij} - 2\nu_T\bar{S}_{ij} \qquad (2.25)$$

where

$$\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i}\right)$$

is the strain-rate tensor, and ν_T is an effective viscosity associated with the residual field motions.

(a) Smagorinsky model

Smagorinsky (1963) suggested a model for ν_T ,

$$\nu_T = (c_S\Delta_A)^2\ (2\bar{S}_{ij}\ \bar{S}_{ij})^{1/2} \qquad (2.26)$$

where c_S is a constant. In experiments one observes a sharp separation of turbulent regions, containing vorticity and non-turbulent regions which are irrotational. A weakness of this model is that, in a non-turbulent irrotational region, ν_T will have a non-zero value. This will give rise to residual stresses in the non-turbulent flow outside of a boundary layer.

(b) Vorticity model

A way around this drawback is to relate ν_T directly to vorticity. A likely possibility is

$$\nu_T = (c_v \Delta_A)^2 \sqrt{\bar{\omega}_i \bar{\omega}_i} \tag{2.27}$$

where $\bar{\omega} = \text{curl } \bar{u}$ is the vorticity, and c_v is a constant.

2.5 Governing Equations for the Filtered Field

Now, neglicting the molecular viscosity term, and dropping terms of higher order than Δ_A^2, filtered momentum equations become

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\bar{u}_i \bar{u}_j + \frac{\Delta_A^2}{24} \nabla^2 \, \bar{u}_i \bar{u}_j - 2\nu_T \bar{S}_{ij}\right) = - \frac{\partial P}{\partial x_i} \tag{2.28}$$

where $P = \frac{p}{\rho} + \frac{1}{3} R_{ii}$. This may be written as

$$\frac{\partial \bar{u}_i}{\partial t} = h_i - \frac{\partial P}{\partial x_i} \triangleq H_i \tag{2.29}$$

where

$$h_i \triangleq - \frac{\partial}{\partial x_j}\left(\bar{u}_i \bar{u}_j + \frac{\Delta_A^2}{24} \nabla^2 \, \bar{u}_i \bar{u}_j - 2\nu_T \bar{S}_{ij}\right)$$

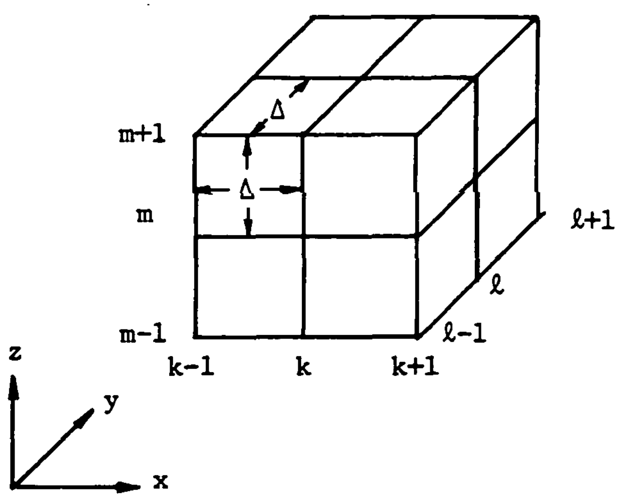
It is in this form that we shall deal with the problem computationally. The manner in which continuity was used to fix P is discussed in the next chapter.

CHAPTER III

NUMERICAL METHOD

3.1 Grid Layout and Notation

A uniform cubic mesh is used, as sketched below. The mesh width Δ need not be the same as the averaging scale Δ_A introduced in the previous chapter.



The i-component of the filtered velocity at the nth time step is written as

$$\bar{u}_{i\,(k,\ell,m)}^{(n)}$$

where (k,ℓ,m) is the meshpoint index for (x, y, z).

We now define the following operator notations:

$\delta/\delta x_i$ = finite difference operator corresponding to $\partial/\partial x_i$

$\delta/\delta t$ = finite difference operator corresponding to $\partial/\partial t$

G = finite difference form of gradient operator

D = finite difference form of divergence operator

$\frac{\delta}{\delta x_j}(u_i f)$ = transport operator corresponding to $\frac{\partial}{\partial x_j}(u_i f)$

Further details of these terms are given next.

3.2 Space Differencing

A fourth order differencing scheme is applied where fourth order accuracy is needed. Since the Leonard and residual stress terms are second order, they can be approximated by second order formula to give the same accuracy. The central difference fourth order scheme is, for example,

$$\frac{\delta \bar{u}}{\delta x} = \frac{1}{12\Delta}\left\{\bar{u}_{(k-2)} - 8\bar{u}_{(k-1)} + 8\bar{u}_{(k+1)} - \bar{u}_{(k+2)}\right\} \tag{3.1}$$

For simplicity, the subscripts ℓ and m are not shown.

Suppose we represent $\bar{u}$ by a discrete Fourier expansion,

$$\bar{u} = \sum_{n} \hat{\bar{u}}(\underline{k})\, e^{i\underline{k}\,\underline{x}} \tag{3.2}$$

where $k_1 = \frac{2\pi}{N\Delta} n_1$ = wave number in the x_1 direction

$n_1 = -N/2,\ldots,0,1,\ldots,(\frac{N}{2} - 1)$

N = Number of mesh points in one direction

The sum extends over all n_1, n_2, and n_3. Substituting (3.2) in (3.1), the Fourier transform of $\delta\bar{u}/\delta x$ is identified as

$$\widehat{\frac{\delta \bar{u}}{\delta x}} = \frac{1}{12\Delta}\left(e^{-i2\Delta k_1} - 8e^{-i\Delta k_1} + 8e^{i\Delta k_1} - e^{i2\Delta k_1}\right)\hat{\bar{u}}$$

$$= \frac{i}{6\Delta}\left\{8\sin(\Delta k_1) - \sin(2\Delta k_1)\right\}\hat{\bar{u}} \tag{3.3}$$

If a modified wave number, k_i', is defined by

$$k_i' \triangleq \frac{1}{6\Delta}\left\{8\sin(\Delta k_i) - \sin(2\Delta k_i)\right\} \tag{3.4}$$

then the Fourier transform of $D\bar{\underline{u}} = 0$ can be written as

$$k_i'\hat{\bar{u}}_i = 0 \tag{3.5}$$

Note that the exact transform of div $\bar{\underline{u}}$ is $k_i\hat{\bar{u}}_i$. Hence, k_i' may be interpreted as the wave number that allows continuity to be satisfied in grid space.

If instead one were to use a second-order central difference scheme,

$$\frac{\delta \bar{u}}{\delta x} = \frac{1}{2\Delta}\left(\bar{u}_{(k+1)} - \bar{u}_{(k-1)}\right) \tag{3.6}$$

The modified wave number , k_i'' , would be

$$k_i'' \triangleq \frac{1}{\Delta}\sin(\Delta k_i) \tag{3.7}$$

k_1, k_1' and k_1'' are compared in Fig. 3.1.

The fourth-order D and G operators are therefore (again only subscripts different from k, ℓ, or m are explicitly shown).

$$\begin{aligned}
D(\bar{\underline{u}}) &= \frac{\delta \bar{u}}{\delta x} + \frac{\delta \bar{v}}{\delta y} + \frac{\delta \bar{w}}{\delta z} \\
&= \frac{1}{12\Delta}\left\{\bar{u}_{(k-2)} - 8\bar{u}_{(k-1)} + 8\bar{u}_{(k+1)} - \bar{u}_{(k+2)}\right\} \\
&\quad + \frac{1}{12\Delta}\left\{\bar{v}_{(\ell-2)} - 8\bar{v}_{(\ell-1)} + 8\bar{v}_{(\ell+1)} - \bar{v}_{(\ell+2)}\right\} \\
&\quad + \frac{1}{12\Delta}\left\{\bar{w}_{(m-2)} - 8\bar{w}_{(m-1)} + 8\bar{w}_{(m+1)} - \bar{w}_{(m+2)}\right\} \\
&= \text{div}\ \bar{\underline{u}} + O(\Delta^4)
\end{aligned} \tag{3.8}$$

$$\begin{aligned}
G(P) &= \left(\hat{e}_x \frac{\delta}{\delta x} + e_y \frac{\delta}{\delta y} + e_z \frac{\delta}{\delta z}\right) P \\
&= \hat{e}_x \frac{1}{12\Delta}\left\{P_{(k-2)} - 8P_{(k-1)} + 8P_{(k+1)} - P_{(k+2)}\right\} \\
&\quad + \hat{e}_y \frac{1}{12\Delta}\left\{P_{(\ell-2)} - 8P_{(\ell-1)} + 8P_{(\ell+1)} - P_{(\ell+2)}\right\} \\
&\quad + \hat{e}_z \frac{1}{12\Delta}\left\{P_{(m-2)} - 8P_{(m-1)} + 8P_{(m+1)} - P_{(m+2)}\right\} \\
&= \text{grad}\ P + O(\Delta^4)
\end{aligned} \tag{3.9}$$

where $\hat{e}_x$, $\hat{e}_y$, $\hat{e}_z$ are unit vectors in the x-, y-, and z-directions.

3.3 DG Operator

$$DG(P) = \frac{\delta}{\delta x_i} \left(\frac{\delta}{\delta x_i} P \right) \tag{3.10}$$

Expanding one term of (3.10), using the fourth-order difference scheme,

$$\frac{\delta}{\delta x}\left(\frac{\delta}{\delta x} P\right) = \frac{1}{(12\Delta)^2} \Big\{ P_{(k-4)} - 16P_{(k-3)} + 64P_{(k-2)} + 16P_{(k-1)} - 130P_{(k)} + 16P_{(k+1)} + 64P_{(k+2)} - 16P_{(k+3)} + P_{(k+4)} \Big\} \tag{3.11}$$

If P is expanded in a discrete Fourier series, similar to (3.2), the Fourier Transform of (3.11) is identified as,

$$\widehat{\left\{\frac{\delta}{\delta x}\left(\frac{\delta P}{\delta x}\right)\right\}} = \frac{1}{(12\Delta)^2} \Big\{ e^{i4\Delta k_1} - 16e^{i3\Delta k_1} + 64e^{i2\Delta k_1} + 16e^{i\Delta k_1} - 130 + 16e^{-i\Delta k_1} + 64e^{-i2\Delta k_1} - 16e^{-i3\Delta k_1} + e^{-i4\Delta k_1} \Big\} \hat{P}$$

$$= \frac{1}{72\Delta^2} \Big\{ -65 + 16\cos(\Delta k_1) + 64\cos(2\Delta k_1) - 16\cos(3\Delta k_1) + \cos(4\Delta k_1) \Big\} \hat{P}$$

$$= -k_1'^2 \hat{P} \tag{3.12}$$

(3.12) may be obtained directly from (3.4). Therefore, in Fourier space DG operator becomes

$$\widehat{DG} = -k_i' \, k_i' \tag{3.13}$$

Compare (3.12) to the following fourth order central differencing scheme, which is a commonly used approximation to $\partial^2/\partial x^2$ operator;

$$\frac{\delta^2}{\delta x^2} P = \frac{1}{12\Delta^2} \left\{ -P_{(k-2)} + 16P_{(k-1)} - 30P_{(k)} + 16P_{(k+1)} - P_{(k+2)} \right\} \tag{3.14}$$

$$\widehat{\left(\frac{\delta^2 P}{\delta x^2}\right)} = -\frac{1}{6\Delta^2} \left\{ 15 - 16 \cos(\Delta k_1) + \cos(2\Delta k_1) \right\} \hat{p}$$

$$\triangleq -\tilde{k}_1^2 \hat{p} \tag{3.15}$$

where $\tilde{k}_1$ is defined by (3.15). k_1^2, $k_1'^2$, and $\tilde{k}_1^2$ are compared in Fig. 3.2 for $N = 16$.

3.4 Transport Difference Operator

Differencing the transport terms in the form of (2.28) will automatically conserve momentum in an inviscid flow. But the computation becomes unstable and the kinetic energy increases. This happens in real flows in spite of the dissipative nature of R_{ij} and the Leonard term. This non-linear instability, first reported by Phillips (1959), arises because the momentum conservative form does not necessarily guarantee energy conservation, and truncation errors in the energy equation are not negligible.

Arakawa (1966) devised a differencing scheme that conserves both mean square vorticity and energy in two-dimensional calculations that use the vorticity and stream function as dependent variables. This is not useful in a three-dimensional flow.

A fourth-order transport differencing scheme that does conserve energy and momentum was developed for the present work:

$$\frac{\delta}{\delta x}(\bar{u}\bar{f}) = \frac{1}{3\Delta}\Big\{(\bar{u}\bar{f})_{(k+1)} - (\bar{u}\bar{f})_{(k-1)}$$

$$+ \bar{u}(\bar{f}_{(k+1)} - \bar{f}_{(k-1)}) + \bar{f}(\bar{u}_{(k+1)} - \bar{u}_{(k-1)})\Big\}$$

$$- \frac{1}{24\Delta}\Big\{(\bar{u}\bar{f})_{(k+2)} - (\bar{u}\bar{f})_{(k-2)} + \bar{u}(\bar{f}_{(k+2)} - \bar{f}_{(k-2)})$$

$$+ \bar{f}(\bar{u}_{(k+2)} - \bar{u}_{(k-2)})\Big\} \qquad (3.16)$$

Again we only show subscripts that differs from k, ℓ, or m. For details see Appendix I. In the present work this was used for the terms $\partial(\bar{u}_i\bar{u}_j)/\partial x_j$; for the Leonard term a second-order version of (3.16) was used,

$$\frac{\delta}{\delta x}(\bar{u}\bar{f}) = \frac{1}{4\Delta}\Big\{(\bar{u}\bar{f})_{(k+1)} - (\bar{u}\bar{f})_{(k-1)}$$

$$+ \bar{u}(\bar{f}_{(k+1)} - \bar{f}_{(k-1)}) + \bar{f}(\bar{u}_{(k+1)} - \bar{u}_{(k-1)})\Big\} \qquad (3.17)$$

The familiar second-order central difference approximation was used for ∇^2 in the Leonard term,

$$\frac{\delta^2\bar{f}}{\delta x^2} = \frac{1}{\Delta^2}(\bar{f}_{(k+1)} - 2\bar{f}_{(k)} + \bar{f}_{(k-1)}) \qquad (3.18)$$

For the residual stress terms, (3.17) was used. The strain-rates and vorticity were computed from the second-order central difference (3.6).

3.5 Time Differencing

A second order Adams-Bashforth method was used for the time integration. As shown by Lilly (1965), this method is very weakly unstable, but the total spurious computational production of kinetic energy is small.

The Adams-Bashforth formula for $\bar{u}_i$ at time step $n+1$ is

$$\bar{u}_i^{(n+1)} = \bar{u}_i^{(n)} + \Delta t \left(\frac{3}{2} H_i^{(n)} - \frac{1}{2} H_i^{(n-1)} \right) + 0(\Delta t^3) \qquad (3.19)$$

where H_i is defined by (2.29). Note that this is an explicit scheme.

3.6 Pressure Field Solution

To study this problem in detail, let us rewrite (2.29) as

$$\frac{\partial \bar{u}_i}{\partial t} - h_i = - \frac{\partial P}{\partial x_i} \qquad (3.20a)$$

Again, continuity is

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \qquad (3.20b)$$

Taking the divergence of (3.20a)

$$\nabla^2 P = \frac{\partial}{\partial x_i} h_i - \frac{\partial}{\partial t} \frac{\partial \bar{u}_i}{\partial x_i}$$

$$= g_1 \qquad (3.21)$$

The usual computational procedure involves choosing the pressure field at the current time step such that continuity is satisfied at the next time step, i.e. so that the new flow field will be divergence free. This must be done very carefully. Let's look at three possibilities. These take advantage of Fourier transformation, for it is known that fast Fourier transforms provide an excellent way to solve the Poisson equation, at least in a rectangular domain.

(a) Method 1

The Fourier transform of (3.21) is

$$-k^2 \hat{P} = \hat{\tilde{g}}_1 \qquad (3.22)$$

where $\tilde{g}_1$ is the difference approximation to g_1 and $k^2 = k_x^2 + k_y^2 + k_z^2$. k_x, k_y, and k_z are wave numbers in x-, y-, and z-directions. By inverse transformation, P can be obtained. However, as will be shown shortly, this method does not give a next velocity field that is divergence-free in grid space. Hence this approach is unsatisfactory.

(b) Method 2

A second approach is to use the difference form of (3.21),

$$\frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2} + \frac{\delta^2}{\delta z^2} \; P = \tilde{g}_1 \tag{3.23}$$

Then P can be obtained by Fourier transforming (3.23),

$$-\tilde{k}_i \tilde{k}_i \hat{P} = \hat{\tilde{g}}_1 \tag{3.24}$$

As will be shown shortly, the pressure from this does not give a divergence-free field in grid space at the next time step. Hence, this method, which was used by Jain (1967), is also unsatisfactory.

(c) Method 3

The finite difference forms of (3.20a) and (3.20b) are

$$\frac{\delta \bar{u}_i}{\delta t} - \tilde{h}_i = -\frac{\delta}{\delta x_i} P \tag{3.25a}$$

$$D\bar{u}_i = 0 \tag{3.25b}$$

where $\tilde{h}_i$ is difference approximation to h_i. If we apply the D operator to (3.25a),

$$DG\ P = \frac{-\delta}{\delta t} (D\bar{u}_i) + D\tilde{h}_i = g_1' \tag{3.26}$$

Then, taking the Fourier transform of (3.26), and using (3.13),

$$-k_i' k_i' \hat{P} = \hat{g}_1' \tag{3.27}$$

Now let's compare the three methods. The Fourier transform of (3.25a) is

$$\frac{\delta\hat{\bar{u}}_i}{\delta t} - \hat{\tilde{h}}_i = -i\, k_i'\, \hat{P} \qquad (3.28)$$

To satisfy continuity in grid space, we want to have

$$\frac{\delta}{\delta t}\, \frac{\delta\bar{u}_i}{\delta x_i} = 0$$

for a flow that has $D\bar{u}_i = 0$ to start. From (3.5), this is equivalent to $k_i'\bar{u}_i = 0$ at the first and next time steps. Using (3.21), and substituting $\hat{P}$ in (3.28) by $\hat{P}$ from (3.22), (3.24) or (3.27) and operating with D on the resulting equations, one obtains,

<u>for method 1:</u>

$$\frac{\delta}{\delta t}\,(k_i'\hat{\bar{u}}_i) = \left(\hat{\tilde{h}}_i - \frac{k_i'k_j'}{k^2}\,\hat{\tilde{h}}_j\right)k_i' \neq 0 \qquad (3.29)$$

<u>for method 2:</u>

$$\frac{\delta}{\delta t}\,(k_i'\hat{\bar{u}}_i) = \left(\hat{\tilde{h}}_i - \frac{k_i'k_j'}{\tilde{k}^2}\,\hat{\tilde{h}}_j\right)k_i' \neq 0 \qquad (3.30)$$

<u>for method 3:</u>

$$\frac{\delta}{\delta t}\,(k_i'\hat{\bar{u}}_i) = \left(\hat{\tilde{h}}_i - \frac{k_i'k_j'}{k'^2}\,\hat{\tilde{h}}_j\right)k_i' \equiv 0 \qquad (3.31)$$

The error introduced by method 1 and method 2 can be seen clearly by observing the magnitude of the ratio k'^2/k^2 or $k'^2/\tilde{k}^2$ as illustrated in Fig. 3.2. Method 3 satisfies continuity at the next time step in grid space, and hence is chosen for pressure field solution here.

3.7 <u>Summary of Difference Equations</u>

Momentum equation:

$$\frac{\delta\bar{u}_i}{\delta t} = -\frac{\delta}{\delta x_j}\left\{\bar{u}_i\bar{u}_j + \frac{\Delta_A^2}{24}\,\frac{\delta^2}{\delta x_k \delta x_k}\,(\bar{u}_i\bar{u}_j) - 2\nu_T\tilde{S}_{ij}\right\} - \frac{\delta P}{\delta x_i}$$

$$= \tilde{h}_i - \frac{\delta P}{\delta x_i} \qquad (3.32)$$

Poisson equation for P:

$$DG(P) = D(\tilde{h}_i) \tag{3.33}$$

where

$$P = \frac{\bar{p}}{\rho} + \frac{1}{3} R_{ii}$$

$$\tilde{S}_{ij} = \frac{1}{2}\left(\frac{\delta \bar{u}_i}{\delta x_j} + \frac{\delta \bar{u}_j}{\delta x_j}\right)$$

$$\nu_T = (c_S \Delta_A)^2 \ (2S_{ij}S_{ij})^{1/2} \quad : \text{ Smagorinsky model}$$

$$= (c_v \Delta_A)^2 \ (\bar{\omega}_i \bar{\omega}_i)^{1/2} \quad : \text{ Vorticity model}$$

$$\overline{\omega}_i = \varepsilon_{ijk} \frac{\delta \bar{u}_j}{\delta x_k}$$

$$c_S, c_v = \text{constant}$$

CHAPTER IV

DECAY OF ISOTROPIC TURBULENCE

4.1 Problem Description

Perhaps the most basic problem in turbulence is the decay of incompressible homogeneous isotropic turbulence. It is with this primitive turbulent flow that our study began. This flow was used to determine the value of the residual stress model constant (C_s or C_v), for use in subsequent calculations of other flows. It also provided a basic testing ground for the computational methods being developed in this program.

The experimental grid turbulence data of Comte-Bellot and Corrsin (1971) were used as the "target" for these predictions. Such experiments closely approximate homogeneous isotropic turbulence, when viewed in a coordinate frame translating at the mean flow velocity.

4.2 The Benchmark Experiment

The pertinent information from Comte-Bellot and Corrsin's (1971) experiments will be reviewed now. The wind tunnel test section, which has a slight secondary contraction to isotropize the turbulence, is sketched in Fig. 4.1 . The turbulence was generated by a biplane square rod grid with mesh size, M , of 5.08 cm. The free-stream air speed, U_o , was 10 m/sec, giving grid mesh Reynolds number U_oM/ν of 34,000 . The streamwise ($\langle u^2\rangle$) and transverse ($\langle v^2\rangle$, $\langle w^2\rangle$) turbulent energy components remained nearly equal to each other during the decay along the test section. These were closely fit by

$$\frac{U_o^2}{\langle u^2\rangle} \simeq 21\left(\frac{U_o t}{M} - 3.5\right)^{1.25} \tag{4.1a}$$

$$\frac{U_o^2}{\langle v^2\rangle} \simeq \frac{U_o^2}{\langle w^2\rangle} \simeq 20\left(\frac{U_o t}{M} - 3.5\right)^{1.25} \tag{4.1b}$$

Correlations, energy spectra and other quantities were measured using hot wire anemometry at $U_o t/M = 42$, 98 and 171. The Reynolds number based on the Taylor microscale, $R_\lambda = \sqrt{(u^2)}\lambda/\nu$, was 71.6, 65.3 and 60.7 at these points.

4.3 Mesh Size Selection

The choice of the computational mesh size requires consideration of the turbulence spectrum (Fig. 4.3). The smallest scales that can be resolved have wave number π/Δ, where Δ is the mesh width. If there are N points in one direction, the largest scales that will be represented in the computation have wave number $2\pi/(N\Delta)$. N and Δ must be chosen such that the computation captures as much of the turbulence energy as possible. It is also desirable that the computation extend to the so-called inertial subrange (Tennekes and Lumley 1972).

The mesh systems used were as follows:

16^3 mesh

$$\Delta = 1.5 \text{ cm}, \; N = 16, \quad \Delta t = 6.25 \times 10^{-3} \text{sec}$$

32^3 mesh

$$\Delta = 1.0 \text{ cm}, \; N = 32, \quad \Delta t = 6.25 \times 10^{-3} \text{sec}$$

The model constants were first evaluated using the 16^3 mesh; the 32^3 calculation then verified that the constants are independent of mesh size.

The corresponding Courant numbers were:

$$Nc \equiv \sqrt{q^3/3} \quad \Delta t/\Delta x, \quad q^2 = \langle \bar{u}^2 \rangle + \langle \bar{v}^2 \rangle + \langle \bar{w}^2 \rangle$$

16^3 mesh

$$Nc \leq 0.06$$

32^3 mesh

$$Nc \leq 0.1$$

4.4 Initial Conditions

We want to prescribe an initial profile that has the proper energy content and spectrum, and is isotropic. A technique for generating a random field that meets these conditions, and also satisfies continuity, was developed for this purpose. Since the computation will treat the filtered field, we matched the initial filtered spectrum and not the initial measured spectrum.

We generated initial filtered field, $\bar{u}_i(\underline{x})$, by first establishing its discrete Fourier transform, $\hat{\bar{u}}_i(\underline{k}_n)$,

$$\bar{\underline{u}}(\underline{x}) = \sum\sum\sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{\bar{\underline{u}}}(\underline{k}_n) e^{i\underline{k}_n \cdot \underline{x}} \tag{4.2}$$

Here $\underline{k}_n$ is the wave number vector defined by $(k_n)_i = \frac{2\pi}{N\Delta} n_i$, where n_i is an integer ranging from $-1/2N$ to $1/2N-1$ for an N^3 mesh system. Note that the maximum wave number is $k_{max} = \pi/\Delta$. If $\bar{\underline{u}}$ is discretized at N points, then the Fourier transform u can only be evaluated at N discrete wave numbers, and that is why the summation must have non-symmetric limits.

The commonly used fast Fourier transform requires N to be 2^m, where m is an integer (see Cochran et al. 1967). Physically N has to be large enough so that wave-number spectra can be treated as smooth functions. As will be shown later, 16^3 or 32^3 mesh systems gave fairly smooth three-dimensional energy spectra.

Now, we can approximate the spectrum function, Φ_{ij} of the filtered field (see Tennekes and Lumley 1972) as,

$$\Phi_{ij}(k_n) \simeq \langle \hat{\bar{u}}_i(\underline{k}_n) \hat{\bar{u}}_j^*(\underline{k}_n) \rangle \tag{4.3}$$

where $< >$ denotes an average over an esemble of experiments or, alternatively, over a spherical shell in k-space with radius k_n (see Section 4.6).

The filtered 3-D energy spectrum, $\overline{E}$, is given by

$$\overline{E}(k) = 2\pi k^2 \Phi_{ii}(\underline{k}) \tag{4.4}$$

E(k)dk is the energy content of a differentially thick spherical shell in wave-number space. Using (4.3), $\overline{E}(k)$ is approximated by

$$\overline{E}(k_n) \approx 2\pi k_n^2 < \hat{u}_i(k_n)\hat{u}_i^*(k_n)> \tag{4.5}$$

To establish the initial field we need to fix the Fourier amplitudes $\hat{u}_i(k_n)$. Equation (4.5) was used to fix $\hat{u}_i(k_n)\hat{u}_i^*(k_n)$ for each k . The vector components $\hat{u}_i$ were chosen by a technique described below.

To get $\hat{u}_i$ to satisfy continuity in grid space, the real and imaginary parts of the transformed velocity vector, $\hat{\underline{u}}$, must be perpendicular to the modified wave-number vector, $\underline{k}'$ (see Equation (3.5)). In the actual computation, we have N^3 points in $\underline{k}$-space, and, for any $\underline{k}$, $\underline{k}'$ can be obtained by (3.4). Then, $\hat{\underline{u}}$ has to be selected on a perpendicular plane to $\underline{k}'$ in $\underline{k}$-space.

To ensure statistical isotropy, the real and imaginary parts of $\hat{\underline{u}}$ must be chosen randomly. First we picked a unit vector A , perpendicular to $\underline{k}'$, by turning a random angle, ϕ , from a reference frame (Fig. 4.2). Here ϕ was selected with uniform probability over the interval 0 to 2π . We then repeated to get a second random unit vector $\underline{B}$, also perpendicular to $\underline{k}'$. The real part of $\hat{\underline{u}}$ was made proportional to the vector $\underline{A}$ and the imaginary to $\underline{B}$, and hence continuity was satisfied. We still needed to fix the relative magnitudes of the real and imaginary parts of $\hat{\underline{u}}(k)$, which we did by a random choice of an angle, θ . Then we defined a and b by

$$a = \cos\theta , \qquad b = \sin\theta \tag{4.6}$$

Finally, we set

$$\hat{u}_j(\underline{k}_n) = |\hat{u}_\ell(\underline{k}_n)\hat{u}_\ell^*(\underline{k}_n)|^{1/2} (aA_j + ibB_j) \tag{4.7}$$

Now, by inverse transforming $\hat{u}_i$, we obtained $\bar{u}_i$, which must be real and will be real if the Fourier transform satisfies

$$\hat{\underline{u}}(-\underline{k}_n) = \hat{\underline{u}}^*(\underline{k}_n) \tag{4.8}$$

In essence, the imaginary contribution for each negative k_n exactly cancels the imaginary contribution of the same positive k_n . Hence, we only needed to generate $\hat{u}_i$ by (4.7) for the upper half of the $\underline{k}$-space, i.e. for $0 \le n \le \frac{N}{2} - 1$. However, this won't fix $\hat{\underline{u}}$ for $n_i = -\frac{N}{2}$. Moreover, if $\hat{\underline{u}}$ is not zero, the velocity field will have an imaginary part (see (4.2)). If instead we wrote (4.2) as

$$\bar{u}_i(\underline{x}) = \sum_{-N/2}^{N/2}\sum\sum \hat{u}_i(\underline{k}_n)e^{i\underline{k}_n \cdot \underline{x}}$$

then $\bar{u}_i$ would be real. However, then we could not take advantage of the FFT routine to invert $\hat{u}_i$. As a practical solution to this dilemma we set $\hat{u}_j$ equal to zero for the wave numbers corresponding to n_i = $-1/2N$. Then (4.2) is essentially the same as

$$\bar{\underline{u}}(\underline{x}) = \sum_{-(\frac{N}{2}-1)}^{(\frac{N}{2}-1)}\sum\sum \hat{\underline{u}}(\underline{k}_n)e^{i\underline{k}_n \cdot \underline{x}}$$

and $\bar{u}_i$ will be real, and an FFT routine may be used. The resulting energy spectrum was therefore slightly low at the highest wave number. However, the effect of this discrepancy was insignificant and became invisible after a few time steps in the computation.

We remark that the field generated by this procedure is quite isotropic. However, as will be shown it has zero skewness, whereas real turbulence has a non-zero skewness. As will be seen, this condition corrected itself in only a few time steps.

4.5 Boundary Conditions

The computational problem can only extend over a part of the experimental region. To get around this difficulty we have used "periodic" boundary conditions, which are of course not really correct. However, if the computational grid system extends over a distance large compared to the scale of the energy-containing motions, the periodic boundary conditions should not introduce appreciable error. The periodic boundary conditions have a great advantage in dealing with the Poisson equation for the pressure by fast Fourier transform (FFT) methods.

4.6 Extraction of Statistical Information from the Computation

The statistical quantities of interest are averages over ensembles of experiments. Since we made only one computational realization in each case, the statistical quantities had to be inferred from appropriate ergodic hypotheses.

In physical space the ensemble average $< >$ was replaced by an average over the flow field. This was done by taking a mean value over N^3 mesh points, i.e.

$$<f(\underline{x})> = \frac{1}{N^3} \sum\sum\sum_{k,\ell,m=1}^{N} f_{(k,\ell,m)} \tag{4.9}$$

The differencing schemes described in Chapter III were used to calculate these quantities.

In wave number space, the ensemble average was replaced by an average over a shell in k-space ("shell average"). Since we have only N^3 discrete points in k-space, the $< >$ average was made by taking a mean value over the points between the two shells with radius $(k-1/2\Delta k)$ and $(k+1/2\Delta k)$.

To get the filtered spectrum, $\overline{E}(k)$, the transformed velocity, $\hat{\underline{u}}$, was obtained by FFT (see 4.2). Then, $<\hat{u}_i(\underline{k}_n)\hat{u}_i^*(\underline{k}_n)>$ was calculated by shell-averaging. The choice of the band width, Δk, is somewhat arbitrary and was set to be 0.1 cm^{-1} here. Then, from (4.5),

$$\overline{E}(k) \simeq \frac{2\pi k^2}{N_k} \sum_{k_n=k-\Delta k}^{k+\Delta k} \hat{\overline{u}}_i(\underline{k}_n)\hat{\overline{u}}_i^*(\underline{k}_n) \tag{4.10}$$

where N_k is the number of points between the two shells with radius $(k-1/2\Delta k)$ and $(k+1/2\Delta k)$. The resulting spectra, evaluated at 0.1 cm^{-1} wave-number intervals, were smooth enough to be represented by continuous curves as shown by solid lines in Fig. 4.4, 4.5, and 4.7 . The filtered spectrum, $\overline{E}(k)$, was then compared to the filtered experimental spectrum, which we obtained using (2.23).

4.7 Selection of the Averaging Scale

Considerable thought was given to the choice of the averaging scale Δ_A . Our failures are as important as our successes, and both will now be discussed.

Consider first the computation with $\Delta_A = 0$. This zero averaging length is equivalent to the unfiltered calculations used in laminar flow, and implies that we are trying to resolve the complete spectrum by a finite difference method. The Leonard term in this case is equal to zero, i.e. $\overline{\overline{u}_i\,\overline{u}_j} = \overline{u}_i\,\overline{u}_j$.

The unfiltered initial energy spectrum is plotted in Fig. 4.3 . The amounts of unfiltered energy for the 16^3 mesh and 32^3 mesh systems are also shown. Figure 4.4 shows the computation for a value of C_s that gives the proper rate of energy decay. Note that for $k > 1/2k_{max}$ the spectrum is distorted considerably at $tU_o/M = 86.5$ and become worse as time increases.

In an instantaneously fluctuating field, higher derivatives are not small and the convergence of the Taylor series is expected to be slow. Use of $\Delta_A = 0$ and the consequent exclusion of the Leonard term caused much distortion of the spectrum, i.e. aliasing error. Indeed, the finite difference method with N mesh points in one direction can only resolve the unfiltered field up to $k = \pi/(2\Delta) = k_{max}/2$, which is a half the maximum wave number in one direction (see Orszag 1969, Orszag and

Israelli 1974). Judging from Fig. 4.4, in a computation without filtering the non-linear interactions transfer too much energy from large to small scales. This excess up-scale energy transfer could be somewhat reduced by using a larger coefficient in the residual stress model. However, this brings an unreasonably high energy decay rate. We conclude that one should never use $\Delta_A = 0$ in a turbulence simulation; filtering is essential.

Let's look next at what happens when the averaging scale Δ_A is equal to the mesh scale Δ. Figure 4.5 shows a computation with a value of C_s that gives the proper energy decay. Note that the errors in the predictions of the filtered spectrum are significant at high wave numbers.

Consider now the computations with $\Delta_A = 2\Delta$, shown in Fig. 4.6, run for values of C_s and C_v that give the proper energy decay. The predictions of the filtered spectrum for both residual stress models are remarkably accurate, even in the coarse 16^3 calculation!

To investigate the effect of the Leonard term separately from the filtering, an additional calculation with $\Delta_A = 2\Delta$ was run with the vorticity model, excluding the Leonard terms (Fig. 4.7). The prediction is poor on high wave number side. Evidently the Leonard terms assists in removal of energy from high wave numbers. We conclude that good results will be obtained with $\Delta_A = 2\Delta$, and that the Leonard terms must be included.

4.8 Selection of C_s and C_v

An analytical way of determining the residual stress model constants, C_s or C_v, is not known. Lilly (1966) estimated $C_s = 0.2$ using several ad-hoc assumptions. Later workers (Deardorff 1971, Fox and Deardorff 1972) calibrated this constant to get the best computational results. In these cases, the required C_s was between 0.10 and 0.22.

In the present study a series of 16^3 mesh calculations were run with different values of each constant, and values selected that gave the best prediction for the filtered rate of energy decay as judged by consideration of the slope of the curve (Fig. 4.10). The constants obtained were as follows;*

$$C_s = 0.206 \quad , \qquad C_v = 0.254$$

Figure 4.9 shows the sensitivity of the predicted filtered energy decay to C_s . Figure 4.10 shows the excellent agreement of the energy history with the data for the final constants.

Figure 4.11 shows the energy decay rate from the 32^3 calculation with these same constants. The spectral results are shown in Fig. 4.8 . The excellent agreement with data confirms that the model constants do not vary with the mesh size, at least in the range covered.

In comparing Fig. 4.10 and 4.11, it must be remembered that these are the filtered energies, which are different in these two mesh systems. Because of the discrete Fourier approximation, not all the turbulent energy is captured (see Fig. 4.3). Filtering improves the situation, because less energy is omitted from the filtered field at high wave number. However, the energy in the discrete approximation to the filtered field was still less than that in the filtered experimental field. To facilitate selection of the constants, the filtered experimental history was shifted as shown in Fig. 4.10 and 4.11 .

4.9 Energetics of the Filtered Field

Multiplying (2.28) by $\overline{u}_i$, and taking an ensemble average, and assuming homogeneity, one finds

$$\frac{d}{dt}\frac{q^2}{2} = -\varepsilon = -(\varepsilon_R + \varepsilon_L) \tag{4.11a}$$

where $q^2 = \langle \overline{u}_i \, \overline{u}_i \rangle$.

The dissipation ε is seen to have two parts, a part representing transfer to the residual field,

*The three digits are not meant to imply accuracy. We actually ran with

$$(2C_s)^2 = 0.17 \quad , \qquad (2C_v)^2 = 0.26 \quad .$$

$$\varepsilon_R = -\langle \bar{u}_i \frac{\partial}{\partial x_j} (2\nu_T \bar{S}_{ij})\rangle \tag{4.11b}$$

and a Leonard term part,

$$\varepsilon_L = +\langle \bar{u}_i \frac{\partial}{\partial x_j} (\frac{\Delta_A^2}{24} \nabla^2 \bar{u}_i \bar{u}_j) \tag{4.11c}$$

To see the relative contributions of the Leonard term and the residual scale motions to the energy decay rate, $\varepsilon_L/\varepsilon_o$ and $\varepsilon_R/\varepsilon_o$ are shown in Fig. 4.14. Here ε_o is the sum of ε_L and ε_R at $(U_o t/M - 3.5) = 42$ where the computation started. ε_L is much smaller than Leonard estimated (1973). As shown by Leonard (1975), ε_L takes energy mostly from the large wave number side thus preventing the damming up of energy in the smaller eddies.

4.10 Other Aspects

No significant difference is observed between Smagorinsky and vorticity models. However, some differences are expected in future applications to unbounded flow problems with turbulent and non-turbulent regions.

The skewness, which is a measure of vorticity production in the energy cascade process, is shown in Fig. 4.12 and 4.13. Since the initial field is randomly generated, the skewness is zero initially, but quickly adjusts to essentially a constant value. For the 16^3 calculation the value is clearly too low (the experimental skewness is about -0.4). For the 32^3 calculation the skewness seems slightly high.

We have emphasized the need for a fourth order differencing scheme, and wonder why others have been able to do so well with second order schemes. The reason may be that the second order difference form of the advection term implicitly includes Leonard-like second order truncation terms and thus the Leonard term is partially taken care of by the truncation. If a fourth order scheme is to be used, the Leonard terms should be included explicitly. We have seen that they are important, particularly at the high wave numbers. We conclude that, for a grid calculation of the type run here, the best results will be given by the fourth-order difference scheme that incorporates the Leonard terms.

A question arose as to the behavior of the vorticity under the difference scheme used here. A two-dimensional irrotational flow was input, ν_T was set equal to zero, and two time steps were taken. The vorticity remained exactly zero, indicating that, at least in a two-dimensional flow, the differencing scheme will not produce unwanted vorticity. This aspect of the computation should receive further study in the future.

4.11 Computational Details

The calculations described above were executed on the CDC 7600 at Lawrence Berkeley Laboratory, using programs written in FORTRAN (Appendix II). The total storage requirements (octal) for 60 bit words were as follows:

16^3 calculation

Large Core Memory: 230,360

Field Length (Small Core) required to load: 121,200

32^3 calculation

Large Core Memory: 1,100,234

Field Length (Small Core) required to load: 121,200

The computer time per computational step was approximately as follows:

16^3 calculation: CPU time $\cong$ 3 sec

32^3 calculation: CPU time $\cong$ 20 sec

The calculation program was carefully checked before these production runs. To check each term in the difference equation (3.32) and (3.33), we imposed systematically artificial flow fields. For the terms involving first derivatives of velocities such as $\overline{S}_{ij}$, $\overline{\omega}_i$, ν_T , $\frac{\delta}{\delta x_i}(\overline{u}_i\,\overline{u}_j)$ and $D\overline{u}_i$, the following linear velocity field was used:

$$\overline{u} = x + 2y + 3z$$

$$\overline{v} = 4x + 5y + 6z$$

$$\overline{w} = 7x + 8y + 9z$$

Then the computed results were compared to the exact values. For the terms with second derivatives, the following quadratic expressions were used:

$$\bar{u} = x^2 + 2y^2 + 3z^2$$

$$\bar{v} = 4x^2 + 5y^2 + 6z^2$$

$$\bar{w} = 7x^2 + 8y^2 + 9z^2$$

Then, at randomly picked mesh points, the computer results for the advection terms, the Leonard terms, $\tilde{h}_i$, and $D(\tilde{h}_i)$ were compared to the exact values obtained analytically.

For the Poisson solver, a sinusoidal pressure field was used to generate $DG(p)$, then the computer results were checked against the imposed pressure field. The initial field was generated as described in Section 4.4 and two time steps were advanced. The subsequent results provided a testing ground for time stepping, the maintenance of a divergence-free velocity field, the overall sequence of computing, and input, output, tape handling, and data reduction routines.

The computer program is given in Appendix II.

CHAPTER V

DISTORTION OF HOMOGENEOUS TURBULENCE BY IRROTATIONAL PLANE STRAIN

5.1 Problem Description

Shear may be viewed as a combination of pure strain and rotation. Therefore, a basic problem is that of homogeneous turbulence acted upon by an imposed uniform homogeneous irrotational strain. Tucker and Reynolds (1968) approximated such a flow experimentally by passing grid-generated turbulence through a passage designed to produce uniform strain in a coordinate system translating with the mean flow velocity. This experimental flow approximates the problem of box turbulence with a constant rate of strain, shown in Fig. 5.1b.

In this chapter we discuss the computation of an idealized homogeneous flow with irrotational pure strain, comparable to the Tucker-Reynolds laterally strained flow. In addition, we treat the return to isotropy following the removal of strain, which roughly corresponds to the experiment in the uniform channel downstream of the straining section.

Tucker and Reynolds did not measure the energy spectrum and hence we cannot make an exact comparison with their data. However, for a qualitative comparison, the initial turbulent intensities in the computation were set to be equal to the experimental values at the beginning of the strained section. Two cases were run. The first case was run with approximately the same initial anisotropy as the experiments. However, there are problems in that the anisotropic field so generated had improper shearing stresses. Therefore, a second calculation was made with an initially isotropic field, and this flow has been used to study the effects of pure strain on homogeneous turbulence.

The initial field for the computation was based on an energy spectrum similar to that used in Chapter IV. However, the Tucker-Reynolds initial energy level was much higher than the energy in the grid flow studied in Chapter IV. To adjust the energy, the amplitude of the Fourier coefficients were multiplied by a constant. The initial one-dimensional energy spectra are shown in Figs. 5.11 and 5.12 by solid curves.

The strain-rate used in the calculation was different from that of the Tucker-Reynolds experiments. In an initial calculation their strain rate was used, but the energy-decay rate in the relaxation section did not match their experiments. The difference was attributed to differences between their (unmeasured) spectrum and that used to start the calculation. We first computed the energy-decay rate in the absence of strain as shown by the solid lines in Fig. 5.5 and 5.8. Then, the strain-rate was determined to get the same total strain as in the Tucker-Reynolds experiments at a point in the flow that would have the same energy in the absence of strain. The final calculations were performed using this strain rate. Therefore, the calculation should be regarded as a "Tucker-Reynolds-like" flow, and not as a simulation of their flow.

5.2 Governing Equations

To handle the imposed mean strain, we express the local velocity and pressure field as*

$$u_i(\underline{x},t) = U_i(\underline{x}) + u_i''(\underline{x},t) \qquad (5.1a)$$

$$p(\underline{x},t) = P(\underline{x}) + p''(\underline{x},t) \qquad (5.1b)$$

where

$$U_i = (U,V,0) = (\Gamma x,-\Gamma y,0) \qquad (5.2a)$$

$$P = -\frac{1}{2}\rho\Gamma^2 (x^2 + y^2) \qquad (5.2b)$$

Γ is the constant strain-rate. With this decomposition, the Navier-Stokes equations for incompressible flow become

$$\frac{\partial}{\partial t}(U_i+u_i'') + \frac{\partial}{\partial x_j}\left\{(U_i+u_i'')(U_j+u_j'')\right\}$$
$$= -\frac{1}{\rho}\frac{\partial}{\partial x_i}(P+p'') + \frac{\partial}{\partial x_j}\left\{\nu\frac{\partial}{\partial x_j}(U_i+u_i'')\right\} \qquad (5.3a)$$

*Note that we place the strain in the x_1-x_2 plane, while Tucker and Reynolds placed it in the x_1-x_3 plane.

$$\frac{\partial}{\partial x_i}(U_i+u_i'') = 0 \tag{5.3b}$$

Now, using the definition (5.2), and noting that $\partial^2 U_i/\partial x_k \partial x_\ell = 0$, this reduces to

$$\frac{\partial}{\partial t} u_i'' + \frac{\partial}{\partial x_j} u_i'' u_j'' + \frac{\partial}{\partial x_j}(U_i u_j'' + U_j u_i'')$$
$$= -\frac{\partial}{\partial x_i}\frac{p''}{\rho} + \frac{\partial}{\partial x_j}\left(\nu \frac{\partial u_i''}{\partial x_j}\right) \tag{5.4a}$$

$$\frac{\partial u_i''}{\partial x_i} = 0 \tag{5.4b}$$

These are the equations that will solve by the methods presented previously. The only modification (compare (2.5)) comes from the third term on the left; this term may be regarded as a forcing function due to the mean strain.

Now, we express each variable quantity, f'' , as

$$f'' = \bar{f} + f' \tag{5.5a}$$

where

$$\bar{f}(\underline{x}) = \int G(\underline{x}-\underline{x}')\, f''(\underline{x}')\, d\underline{x}' \tag{5.5b}$$

Note that $\bar{f}$ is now the filtered f'' field. The $U_i u_j''$ term in (5.4a) is filtered using the method described in chapter II, giving

$$\overline{U_i u_j''}(x_o) = \int G(\underline{x},\underline{x}_o)\left[U_i(\underline{x}_o) + \frac{\partial U_i(x_o)}{\partial x_k}(x_k-x_{ko})\right]$$
$$\left[\bar{u}_j(\underline{x}_o) + \frac{\partial \bar{u}_j}{\partial x_k}(x_o)(x_k-x_{ko}) + \frac{1}{2}\frac{\partial^2 \bar{u}_j}{\partial x_k \partial_\ell}(x_o)\,(x_k-x_{ko})(x_\ell-x_{\ell o})\right.$$
$$\left. + \ldots u_j'\right] d\underline{x}$$
$$= U_i\bar{u}_j + \frac{\Delta_A^2}{24}\nabla^2 U_i\bar{u}_j + O(\Delta_A^4) \tag{5.6}$$

The model for the residual stress, R_{ij}, is taken as

$$R_{ij} = \frac{1}{3} R_{\ell\ell}\, \delta_{ij} - \nu_T\, 2S_{ij} \tag{5.8}$$

where now S_{ij} is the total strain-rate,

$$S_{ij} = \frac{1}{2}\left\{\frac{\partial}{\partial x_j}(U_i+\bar{u}_i) + \frac{\partial}{\partial x_i}(U_j+\bar{u}_j)\right\} \tag{5.9a}$$

Since we expect the strain-rate to be dominated by small scales, the eddy viscosity, ν_T, was evaluated using the vorticity model,

$$\nu_T = (C_v \Delta_A)^2\, (\bar{\omega}_i \bar{\omega}_i)^{1/2} \tag{5.9b}$$

$$\underline{\bar{\omega}} = \text{curl}\ \underline{\bar{u}} \tag{5.9c}$$

Since the imposed flow is irrotational, ν_T is based on the total RMS vorticity.

Then, filtering (5.4), and again neglecting the viscous term, the following equations are obtained (compare (2.28)).

$$\frac{\partial \bar{u}_i}{\partial t} = -\frac{\partial}{\partial x_j}\left\{\bar{u}_i \bar{u}_j + \frac{\Delta_A^2}{24}\nabla^2\,(\bar{u}_i\bar{u}_j) - 2\nu_T(\bar{S}_{ij}+\mathcal{S}_{ij})\right\}$$

$$-\frac{\partial P}{\partial x_i} + F_i \tag{5.10a}$$

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{5.10b}$$

where

$$P = \frac{\bar{p}}{\rho} + \frac{1}{3} R_{ii} \tag{5.11a}$$

and

$$\mathcal{S}_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) \tag{5.11b}$$

F_i are the terms through which the major effects of the mean strain come in. These are

$$F_i = \frac{\partial}{\partial x_j} \left[U_i \overline{u}_j + U_j \overline{u}_i + \frac{\Delta_A^2}{24} \nabla^2 \left(U_i \overline{u}_j + U_j \overline{u}_i \right) \right] \tag{5.11c}$$

Note the appearance of a Leonard correction term.

For the computation, the difference form of (5.10) is used,

$$\frac{\delta \overline{u}_i}{\delta t} = -\frac{\delta}{\delta x_j} \left\{ \overline{u}_i \overline{u}_j + \frac{\Delta_A^2}{24} \frac{\delta^2}{\delta x_k \delta x_k} \overline{u}_i \overline{u}_j - 2\nu_T (\tilde{S}_{ij} + \mathcal{S}_{ij}) \right\}$$

$$- G(P) + \tilde{F}_i \tag{5.12}$$

where $\tilde{F}_i$ is the difference form of F_i and

$$\tilde{S}_{ij} = \frac{1}{2} \left(\frac{\delta \overline{u}_i}{\delta x_j} + \frac{\delta \overline{u}_j}{\delta x_i} \right) \tag{5.13}$$

The exact expressions for $\mathcal{S}_{ij}$ were used. As before, the Poisson equation for P is obtained by operating with D on (5.10a),

$$DG(P) = D(\tilde{h}_i) + D(\tilde{F}_i) - \frac{\delta}{\delta t} D(\overline{u}_i) \tag{5.14}$$

where
$$\tilde{h}_i = -\frac{\delta}{\delta x_j} \left\{ \overline{u}_i \overline{u}_j + \frac{\Delta_A^2}{24} \frac{\delta^2}{\delta x_k \delta x_k} \overline{u}_i \overline{u}_j - 2\nu_T (\tilde{S}_{ij} + \mathcal{S}_{ij}) \right\} \tag{5.15}$$

The space differencing and the time advancing schemes are the same as those explained in Chapter III. Periodic boundary conditions in all three directions were imposed, and the same solution procedure as described in Chapter IV was applied.

5.3 Anisotropic Initial Condition

Anisotropy in grid generated turbulence is not negligible in many experiments (e.g. Grant and Nisbet (1957), Uberoi (1963), Tucker and Reynolds (1968)). Therefore, to make the initial condition reasonably close to the experiments, it was felt desirable to generate an initial field with anisotropy. A method for this will now be described.

Suppose that the u and v components of turbulence energy are equal while the w component is different,

$$\langle \overline{u}^2 \rangle = \langle \overline{v}^2 \rangle \tag{5.16a}$$

$$\langle \overline{w}^2 \rangle = (1+R) \langle \overline{u}^2 \rangle \tag{5.16b}$$

where $\langle\ \rangle$ denotes an average over an ensemble of experiments. For the Tucker-Reynolds experiments, w is in the mean flow direction and $R \simeq 0.45$. Now let us decompose $\overline{w}$ into its isotropic part, $\overline{w}_I$, and anisotropic part, $\overline{w}_A$, such that

$$\langle \overline{u}^2 \rangle = \langle \overline{v}^2 \rangle = \langle \overline{w}_I^{\,2} \rangle \tag{5.17a}$$

and

$$\overline{w} = \overline{w}_I + \overline{w}_A \tag{5.17b}$$

If we assume that the isotropic part can be generated by the method in Chapter IV, then, for continuity to be satisfied,

$$D\,(\overline{w}_A) = 0 \tag{5.18}$$

This is a crude assumption. However, unless we know more about the initial turbulence structure, this is perhaps the best we can do. Now (5.18) in Fourier transformed space can be written as,

$$\hat{\overline{w}}_A\, k_3' = 0 \tag{5.19}$$

where ^ denotes a Fourier transform and k_3' is defined by (3.4). Therefore, $\hat{\overline{w}}$ can have a non-zero values only when $k_3' = 0$. Then, following the same procedure discussed in Section 4.4, we get

$$\hat{\hat{w}}_A(k_1,k_2,0) = \left(\frac{R}{3+R}\,\hat{q}^2\right)^{1/2} (a+ib) \qquad (5.20)$$

where

$$\hat{q}^2 = \hat{u}\hat{u}^* + \hat{v}\hat{v}^* + \hat{w}_I\hat{w}_I^* + \hat{w}_A\hat{w}_A^*$$

and $a = \cos\theta$, and $b = \sin\theta$ are obtained from a random angle, θ, with uniform probability from 0 to 2π.

The initial condition for the first run was generated by this procedure and, for an input R of 0.45, the generated field emerged with $R \simeq 0.43$. This field had shearing stresses not present in the actual flow, and so the second run was made using an isotropic initial field generated by the method described in Chapter IV. Further studies in Section 5.5 and 5.6 are based on the second run. Both runs are reported for completeness.

5.4 Results

The results of the following two cases are presented.

(1) Anisotropic initial field

$$\frac{\langle \overline{w}^2 \rangle}{\langle \overline{u}^2 \rangle} \simeq \frac{\langle \overline{w}^2 \rangle}{\langle \overline{v}^2 \rangle} \simeq 1.43$$

(2) Isotropic initial field

$$\langle \overline{u}^2 \rangle \simeq \langle \overline{v}^2 \rangle \simeq \langle \overline{w}^2 \rangle$$

For both cases, the mean stream speed was taken as $w_o = 240$ in/sec, $\Gamma = 1.457$/sec., $\Delta = 0.59$ in and $\Delta t = 5.36\times10^{-3}$ sec. This corresponds to a Courant number $\sqrt{q^2/3 + U_{max}^2}\ \Delta t/\Delta \le 0.15$. The 16^3 mesh system was run using the vorticity model, with $C_v = 0.206$ as obtained from the isotropic decay studies.

For convenience of reader, three of the Tucker-Reynolds measurements for the turbulent intensities, $\langle \overline{u_1^2} \rangle / W_o^2$, the turbulent energy ratios, $\langle \overline{u_1^2} \rangle / q^2$, and the structural parameter, $K_1 = \{\langle \overline{v^2} \rangle - \langle \overline{u^2} \rangle\} / \{\langle \overline{v^2} \rangle + \langle \overline{u^2} \rangle\}$, are replotted in Figs. 5.2, 5.3 and 5.4 For the case 1 , the same quantities are plotted in Figs. 5.5, 5.6 and 5.7. Here, for comparison, the time in computed results, t , are converted into the downstream distance, Z , by $Z = W_o t$. The behavior of these are comparable to Tucker and Reynolds results. However, when the strain is turned off, the rate of return to isotropy is much slower than in the experiments. It is interesting that this is consistent with the feelings of some workers that the return to isotropy in the Tucker-Reynolds flow is too rapid (Reynolds 1975), perhaps because of defects in the experimental simulation of homogeneity. The same quantities for case 2 are shown in Figs. 5.8, 5.9 and 5.10.

One-dimensional energy spectra were obtained in a similar manner to that described in Chapter IV (see Tennekes and Lumley (1972)). The only difference is that the shell average in Chapter IV is replaced by a plane average in wave-number space, i.e.

$$\overline{E}_{11}(k_1) = \frac{1}{N^2} \sum_{k_2} \sum_{k_3} \hat{\overline{u}}(\underline{k}_n)\ \hat{\overline{u}}^*(\underline{k}_n) \qquad (5.21)$$

Here, the notations are the same as before. These spectra were computed at three different times or downstream locations (Figs. 5.11 and 5.12). At zeroth time step, $\overline{E}_{11}(k_1)$, $\overline{E}_{22}(k_2)$ and $\overline{E}_{33}(k_3)$ are almost identical, as they should be. By the end of the straining period (75th time step), E_{11} and E_{22} have become quite different. E_{11} is flatter on the large eddy side while both small-scale spectra are nearly the same. Over the last period, in which there is no strain, the spectra approached one another very slowly, as seen by the spectra at the 125th time step.

The calculations were run on a CDC 7600, and required approximately 7 minutes for each case. Storage requirements were similar to those for isotropic decay.

5.5 Energetics of the Filtered Field

Multiplying (5.10) by $\bar{u}_i$, taking the ensemble average, and assuming homogeneity, one finds (compare (4.11))

$$\frac{\partial q^2/2}{\partial t} = \mathcal{P} + \mathcal{P}_L - \varepsilon_R - \varepsilon_L \tag{5.22}$$

where

$$q^2 = \langle \bar{u}_i \bar{u}_i \rangle \tag{5.23a}$$

$$\mathcal{P} = -\langle \bar{u}_i \bar{u}_j \rangle \frac{\partial U_i}{\partial x_j} \tag{5.23b}$$

$$\mathcal{P}_L = -\frac{\Delta_A^2}{24} \left\{ \langle \bar{u}_i \frac{\partial}{\partial x_j} \nabla^2 (U_i \bar{u}_j) \rangle + \langle \bar{u}_i \frac{\partial}{\partial x_j} \nabla^2 (U_j \bar{u}_i) \rangle \right\} \tag{5.23c}$$

ε_R and ε_L are the dissipation terms discussed in Section 4.9. $\mathcal{P}$ and $\mathcal{P}_L$ are the production terms. Note the appearance of a Leonard production term, $\mathcal{P}_L$, that comes from the non-linear interaction between the mean strain and the filtered field.

The computed behavior of these terms is shown in Fig. 5.13. Note that $\mathcal{P}_L$ contributes significantly, particularly where the anisotropy is large near the end of straining period.

5.6 Assessment of Turbulence Closure Models

Turbulence computation of the conventionally averaged (ensemble or space) quantities has been based on some ad hoc closure models with a number of adjustable constants. As pointed out by Reynolds (1974a,b), more systematic approaches are desirable for generalized turbulence models. Even though laboratory experiments provide actual quantities, experiments are limited because important properties like the pressure-strain correlations are difficult to measure directly. On the other hand, computer-generated experiments provide a vast amount of data on the flow field, and hence the numerical experiments can be used to study the closure models. We have attempted to study the pressure-strain terms and other statistical quantities using the present computation. Even though the 16x16x16 mesh calculation gives good results for the energy components, as will be shown shortly we cannot use the computer generated field to compute the pressure-strain term directly, at least not in the present calculation.

The exact Reynolds stress equations for homogeneous flow without mean deformation are

$$\frac{dR_{ij}}{dt} = T_{ij} - D_{ij} \tag{5.24}$$

where

$$T_{ij} = < p\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)>$$

$$D_{ij} = 2\nu < \frac{\partial u_i}{\partial x_k}\frac{\partial u_j}{\partial x_k}>$$

Here u_i denotes the turbulent components of velocity and p is the turbulent pressure divided by the fluid density.

The pressure-strain term T_{ij} is responsible for the return to isotropy following removal of strain. The modeling of T_{ij} has been the subject of much discussion. Since no direct measurement of this term is known, we tried to estimate this term using the present computation in the return-to-isotropy portion of the computation.

The computed pressure-strain term, (T_{ij}) , was obtained from

$$(T_{ij})_c = < P\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)> \tag{5.25}$$

Here < > denotes an average over the flow field. The fourth order central differencing scheme was used for the $\delta/\delta x_j$ terms.

For comparison T_{ij} was obtained a second way. Using $D_{ij} = 2/3\ \varepsilon$ (see Reynolds (1974b), (1975)), T_{ij} was obtained from the computed R_{ij} history as

$$(T_{ij})_R = \frac{d}{dt}<\bar{u}_i\bar{u}_j> + \frac{2}{3}\varepsilon\delta_{ij} \tag{5.26}$$

ε was estimated from the energy equation, which in the absence of strain is

$$\varepsilon = -\frac{d \langle \bar{u}_i \bar{u}_i / 2 \rangle}{dt} \tag{5.27}$$

ε agreed well with $\varepsilon_R + \varepsilon_L$, computed directly. The time derivatives were approximated by a second order central differencing formula.

Finally, we predicted T_{ij} by Rotta's (1951) model for T_{ij} in the absence of mean strain,

$$(T_{ij})_m = -A_o \, \varepsilon \, b_{ij} \tag{5.28}$$

where A_o is a constant and

$$b_{ij} = \frac{\langle \bar{u}_i \bar{u}_j \rangle}{q^2} - \frac{\delta_{ij}}{3}$$

We used $A_o = 2.5$, as suggested by Reynolds (1975).

These three results are shown in Fig. 5.14. It appears that $(T_{ij})_c$ is quantitatively poor. This is attributed to the coarseness of the 16^3 mesh. We conclude that T_{ij} undoubtedly contains some Leonard-like terms, and this must account for the difference between $(T_{ij})_c$ and $(T_{ij})_R$. While T_{ij} can be estimated from the R_{ij} equation à la (5.26), it cannot be computed directly from the calculated field with such a coarse grid. A repeat of this work using a 32^3 grid is recommended. This should be accompanied by a careful analysis of the Leonard terms arising in the R_{ij} equations.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

In this thesis we have developed the basic approach to computation of three-dimensional time-dependent turbulent flows. We have seen that, with a modest 16x16x16 mesh and a residual scale stress model, many interesting features of experimental flows can be computed. Work remains to be done in the development of better approaches for using this type of computation to assess turbulence model equations, and to extend the procedure to other flows.

It would be informative to study the effect of the initial spectrum on the rate of return to isotropy. This might be done by removing the strain at different points along the straining portion of the Tucker-Reynolds flow, allowing isotropy to be restored from points with different spectra.

In extending the method to other interesting flows, problems to be resolved include the handling of non-periodic boundary conditions, solid boundaries, and free boundaries connecting the region of computation to irrotational flow outside.

One useful problem to study would be the case of homogeneous turbulence near a wall, without mean shear, for which some experimental data exist (Uzkan and Reynolds (1967)), and would be the diffusion of turbulence into a non-turbulent region, again without shear. It is recommended that experience with these simple problems be gained before a more complex flow is attempted.

When one moves on to handle flows like jets, wakes, and mixing regions, it should be possible to take advantage of the fact that the flow outside of the superlayer is irrotational, and to use the exact solution for unsteady irrotational flow to extend the calculation to infinity out beyond the mesh. Care must be taken that the numerical scheme does not produce vorticity in an irrotational flow; the diffusion of vorticity by ν_T will also have to be handled in a way that prevents its diffusion into the irrotational external field.

Eventually it may be possible to treat practical flows, such as boundary layers, wakes, combustion, etc., by these methods. But much more effort should first be devoted to fully understanding the nuances of the residual scale models, grid-schemes, differencing schemes, filters, etc. that are the bases for this type of numerical simulation.

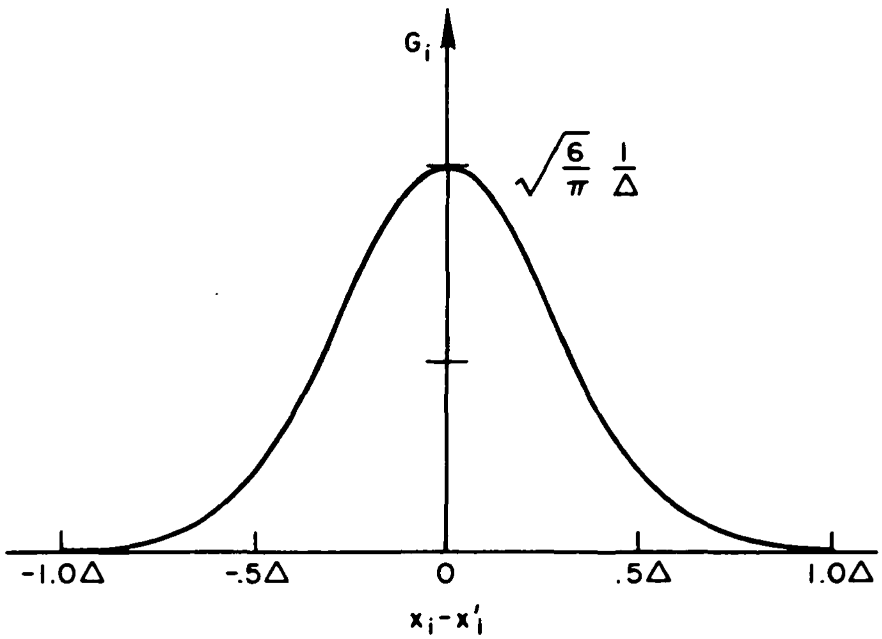


Fig. 2.1. Gaussian Filter

$$G_i = \sqrt{\frac{6}{\pi}}\ \frac{1}{\Delta} \exp\{ -6(x_i - x_i')^2/\Delta^2\}$$

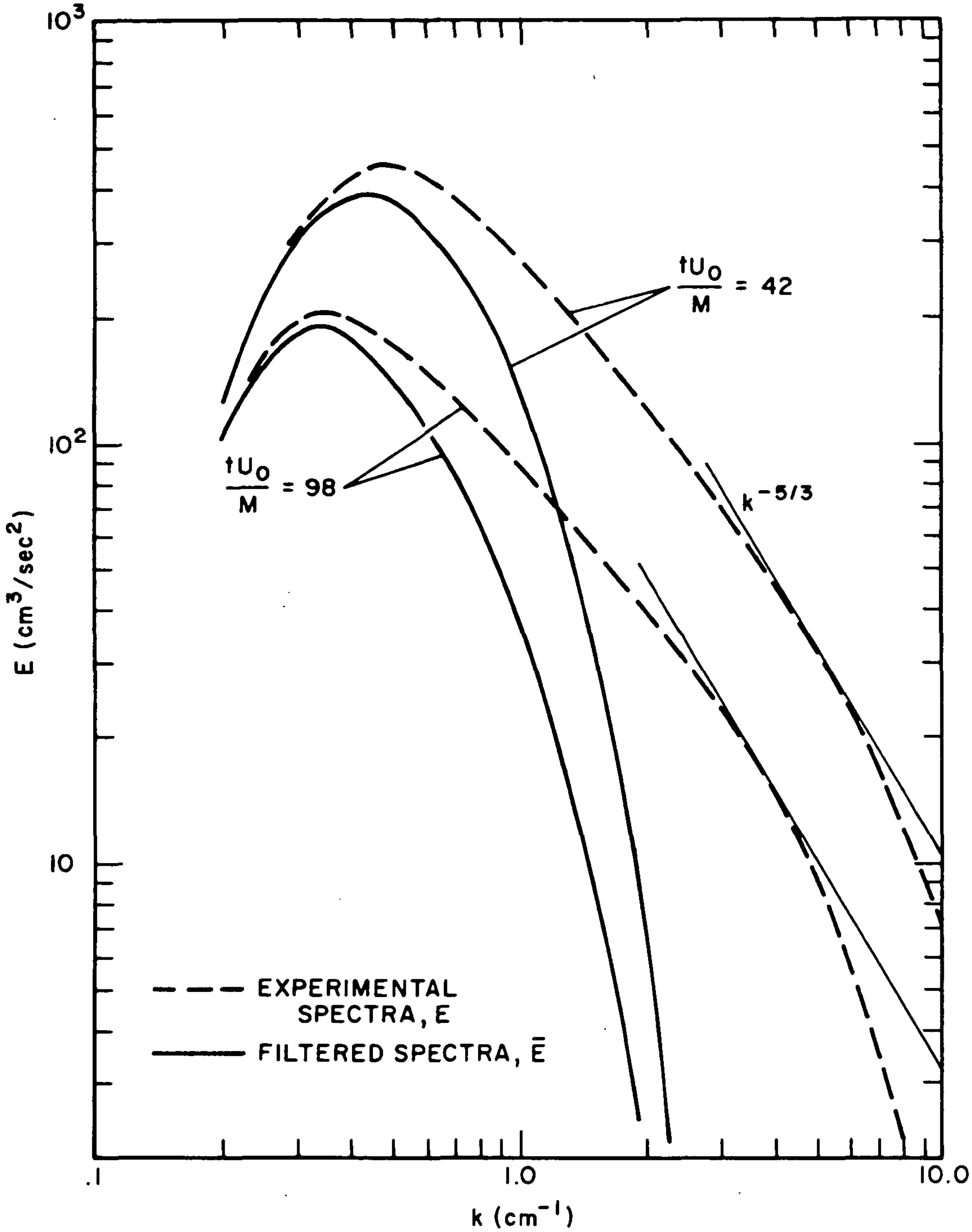


Fig. 2.2. Filtered Energy Spectra, $\overline{E}(k,t)$

$\Delta_A \quad = \quad 2\Delta \quad$ (see Eqn. 2.23)

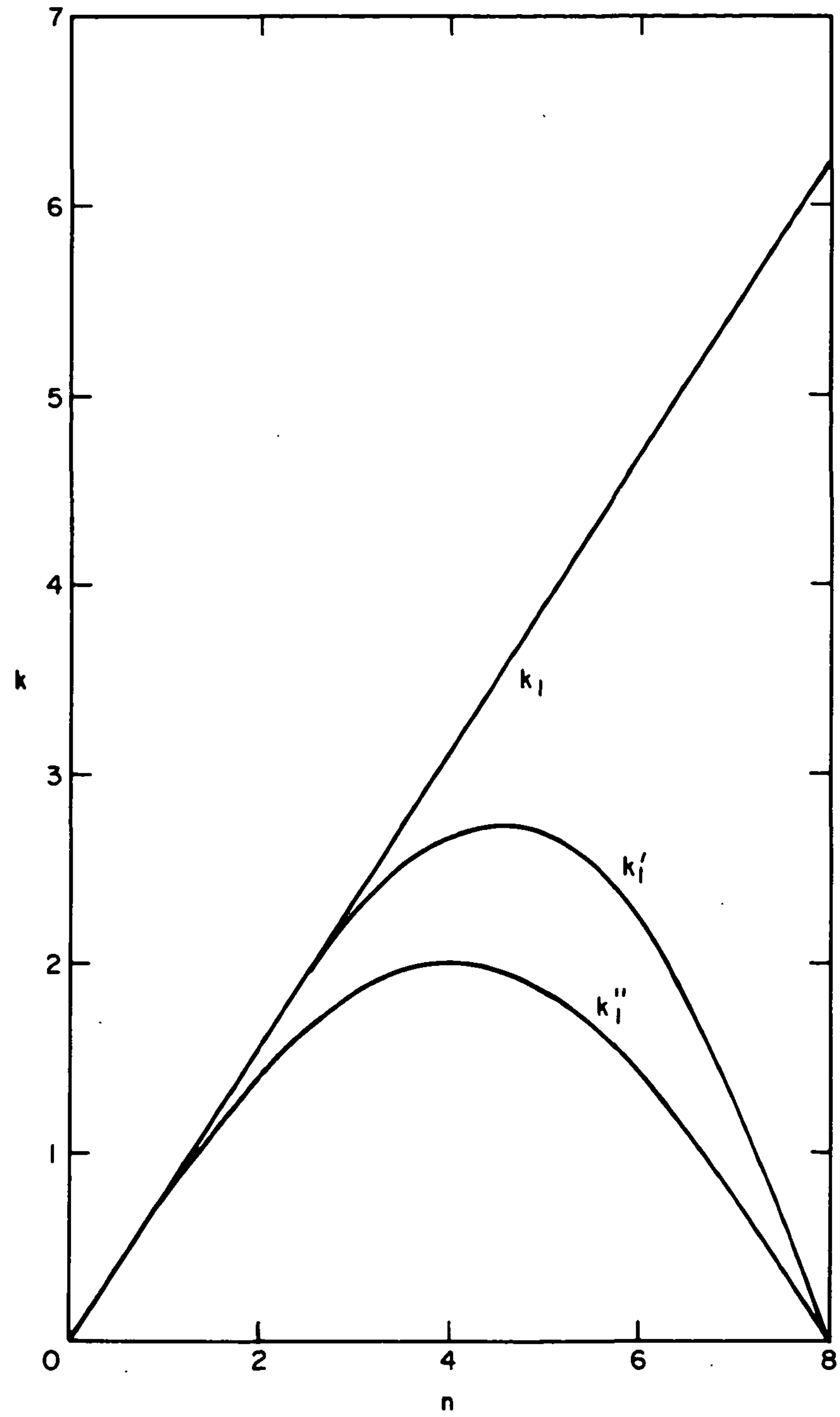


Fig. 3.1. Comparison of Modified Wave Numbers
(see Eqn. 3.4 and 3.7)

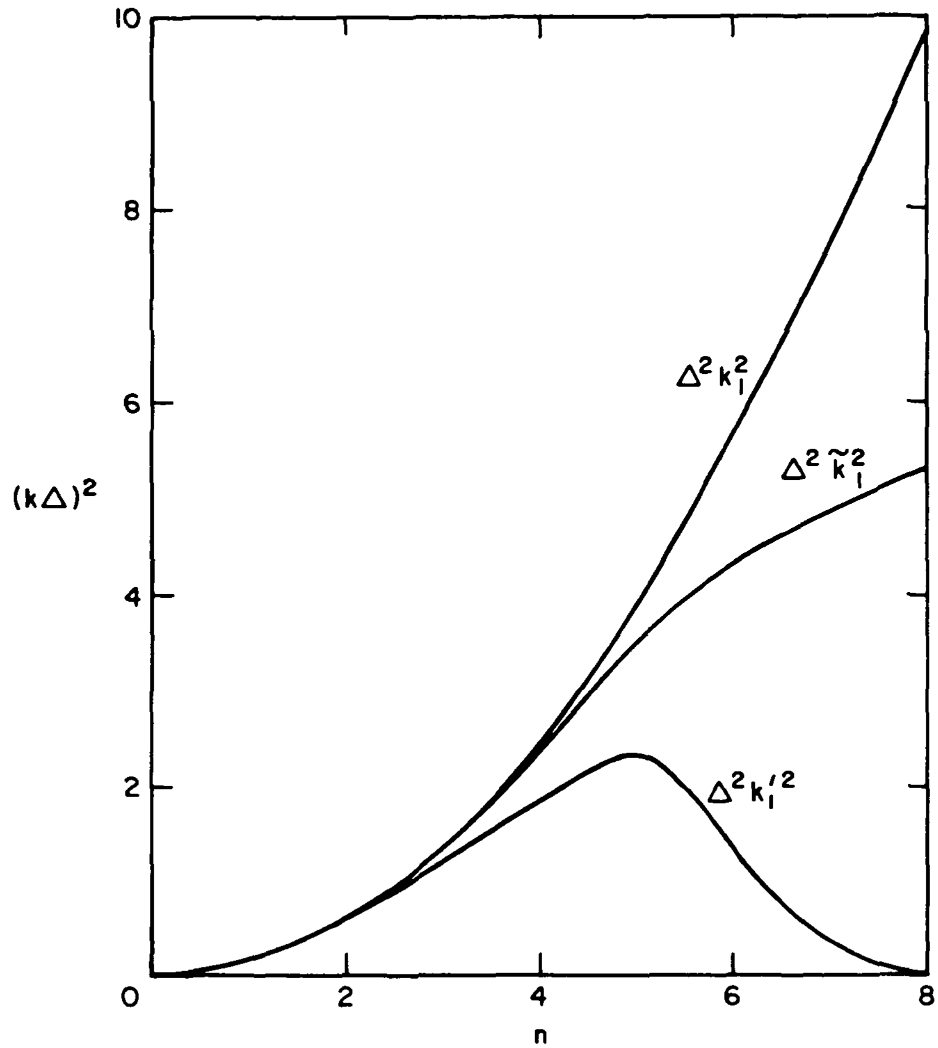


Fig. 3.2. Comparison of Div(Grad) Operator

k_1^2 = exact

$\tilde{k}_1^2$ = fourth order (see Eqn. 3.15)

$k_1'^2$ = fourth order (see Eqn. 3.12)

N = 16

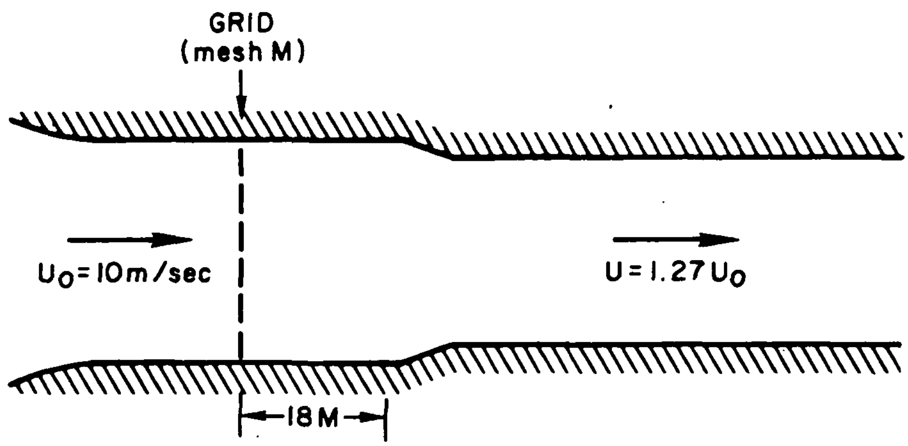


Fig. 4.1. Sketch of Wind Tunnel Test Section for a Generation of Isotropic Flow.

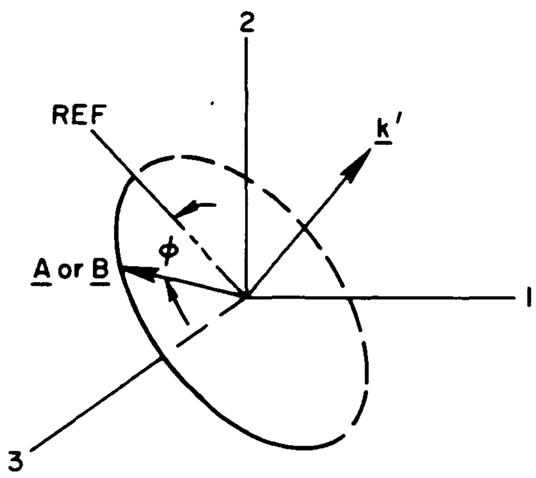


Fig. 4.2. Determination of A or B Vector

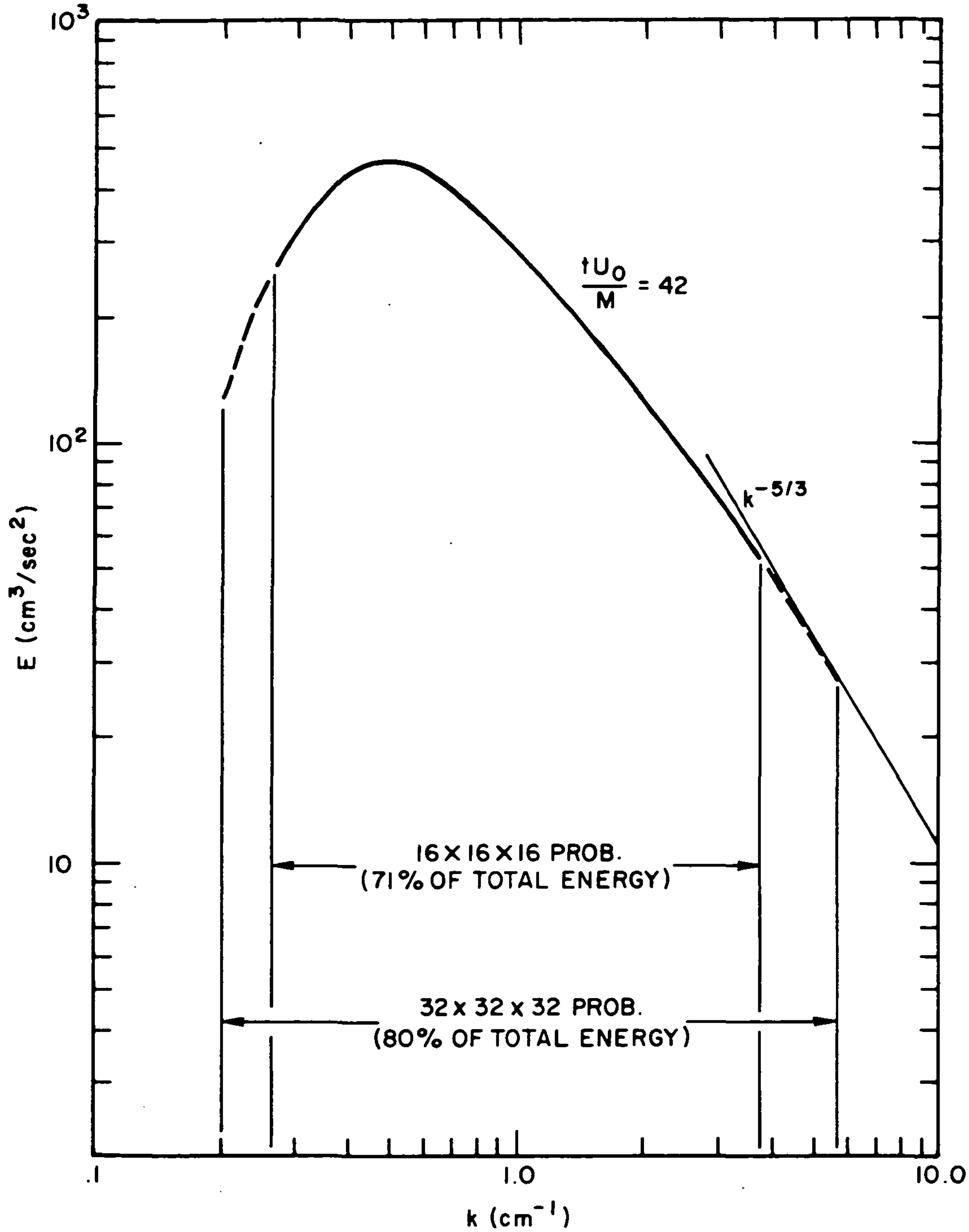


Fig. 4.3. Unfiltered Initial Energy Spectrum
(Data by G. Comte-Bellot and S. Corrsin 1971)

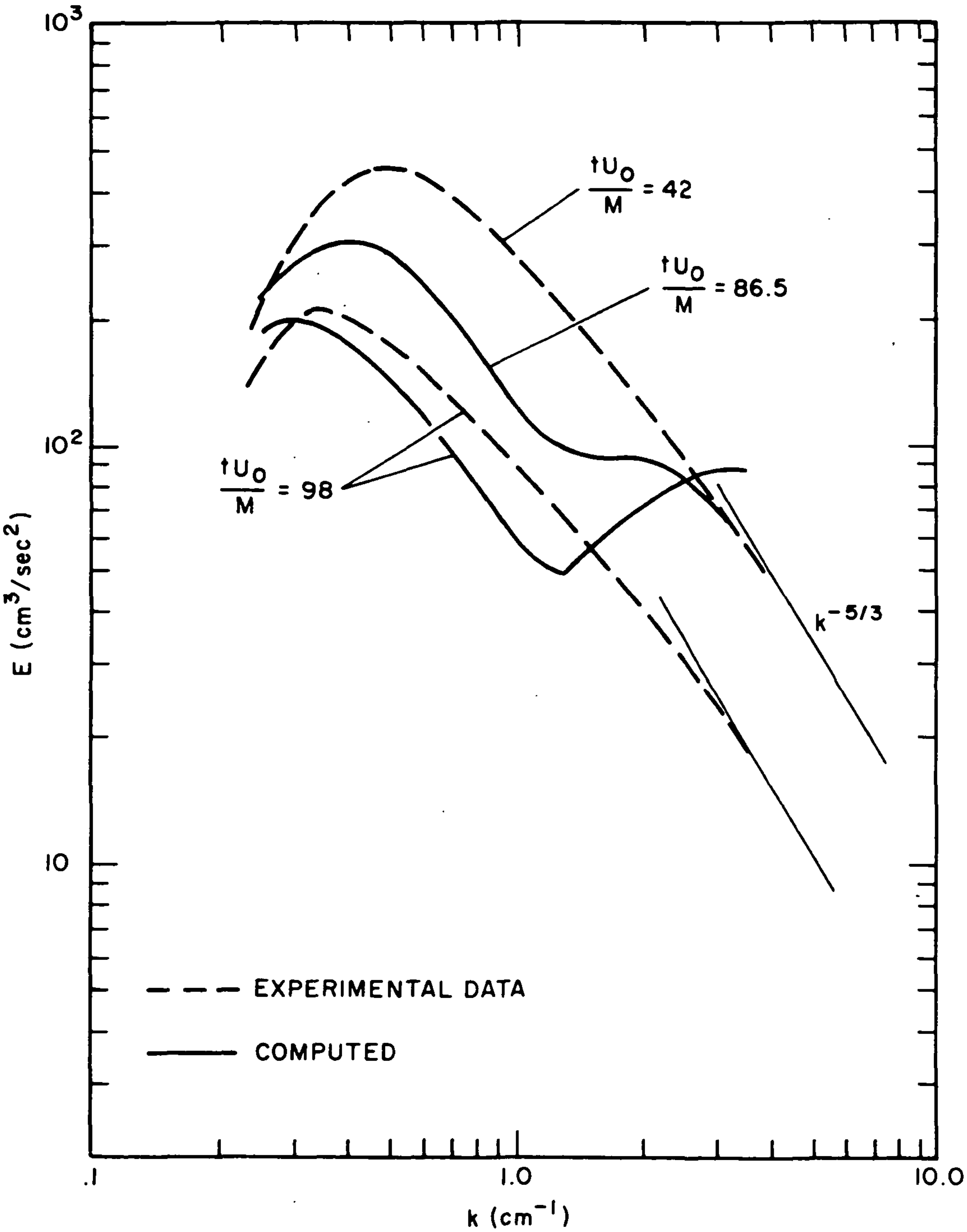


Fig. 4.4. Evolution of Energy Spectra
16x16x16 Mesh: Smagorinsky Model:

$\Delta_A = 0$; $\Delta = 1.5$ cm

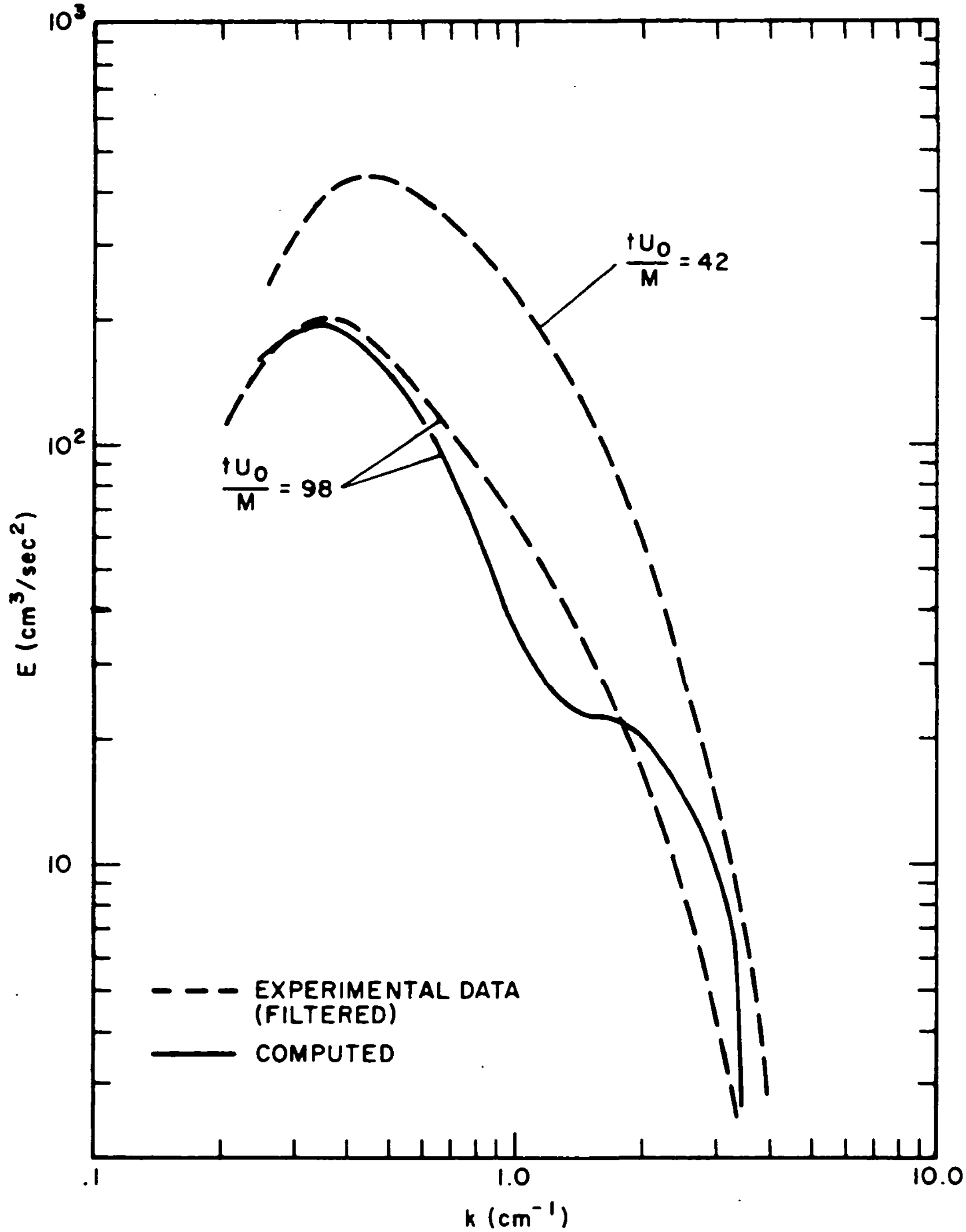


Fig. 4.5. Filtered Energy Spectra
16x16x16 Mesh: Smagorinsky Model:

$\Delta_A = \Delta$; $\Delta = 1.5$ cm

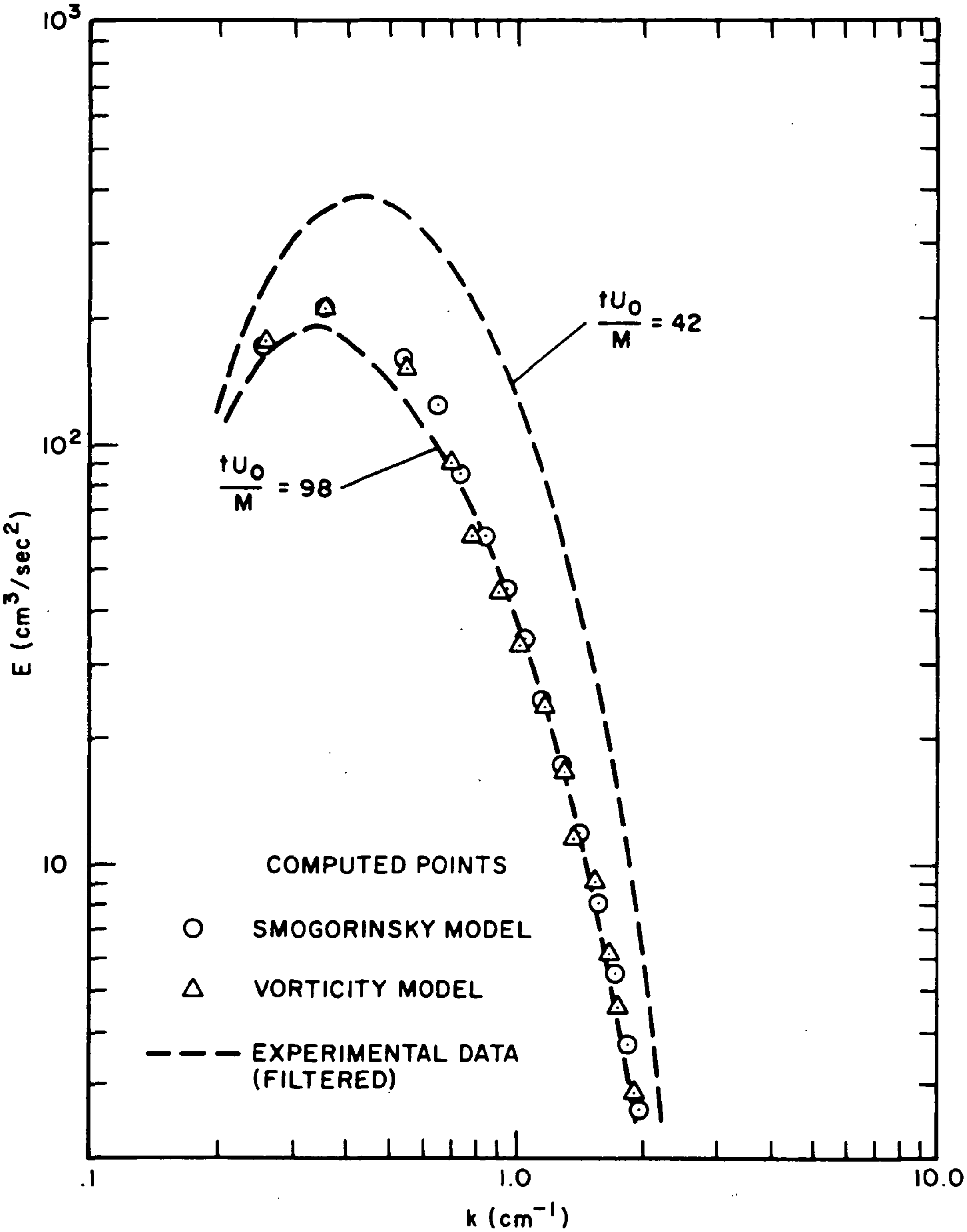


Fig. 4.6. Filtered Energy Spectra -- A Comparison of Smagorinsky and Vorticity Model
16x16x16 Mesh: $\Delta_A = 2\Delta$; $\Delta = 1.5$ cm

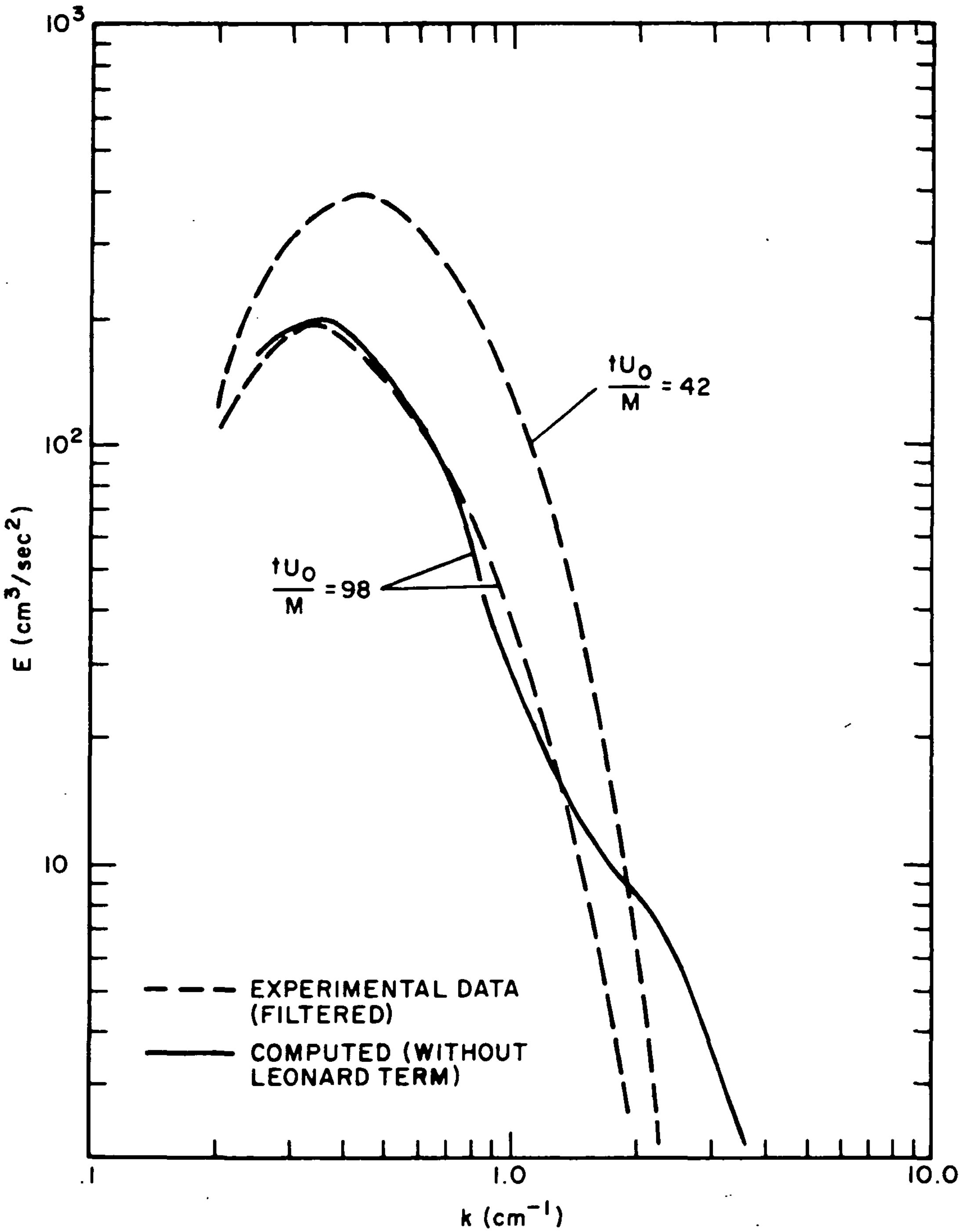


Fig. 4.7. Filtered Energy Spectra-Effect of Leonard Term 16x16x16 Mesh: Vorticity Model:

$\Delta_A = 2\Delta$; $\Delta = 1.5$ cm

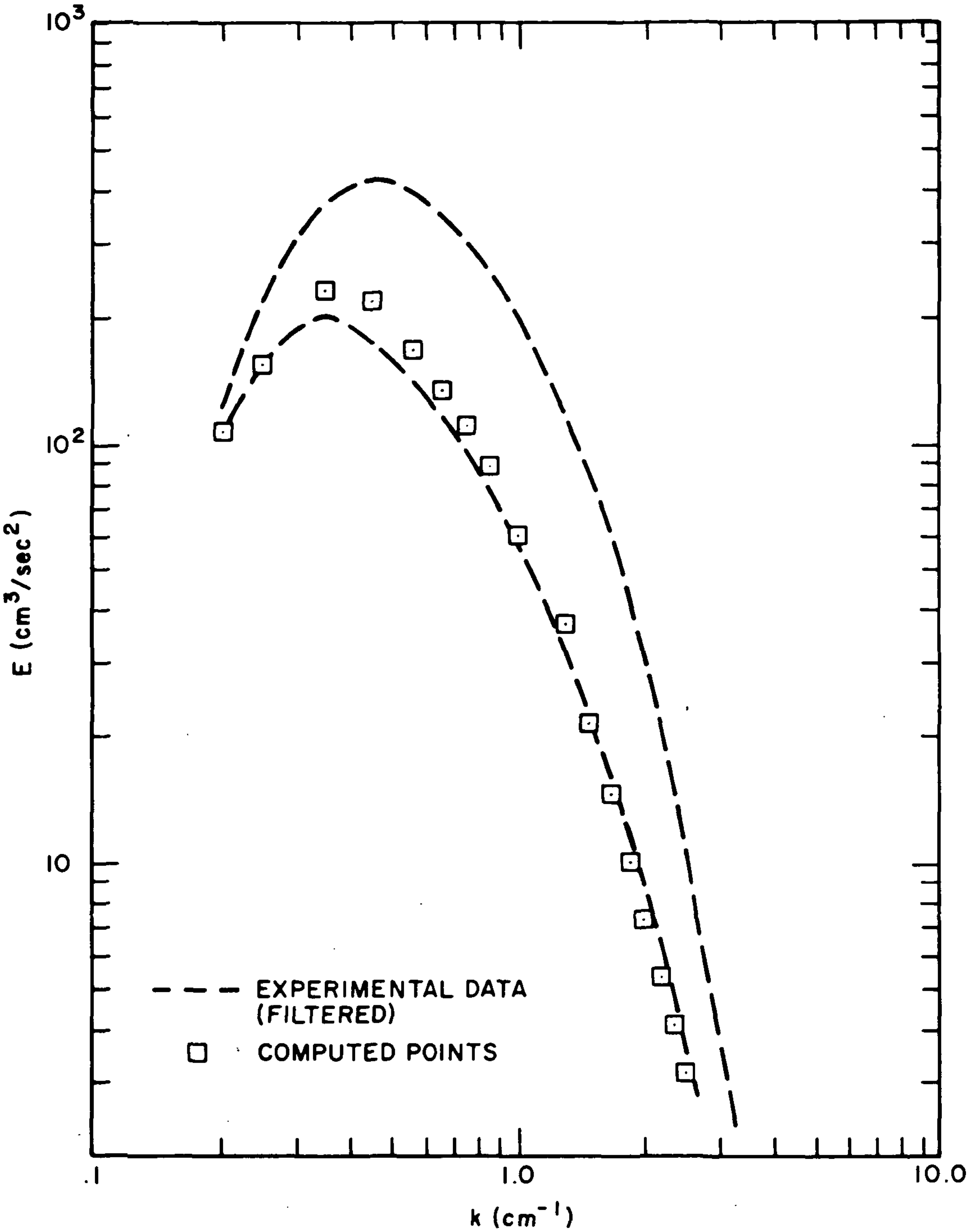


Fig. 4.8. Filtered Energy Spectra
32x32x32 Mesh: Smagorinsky Model:

$\Delta_A = 2\Delta \; ; \; \Delta = 1.0$ cm

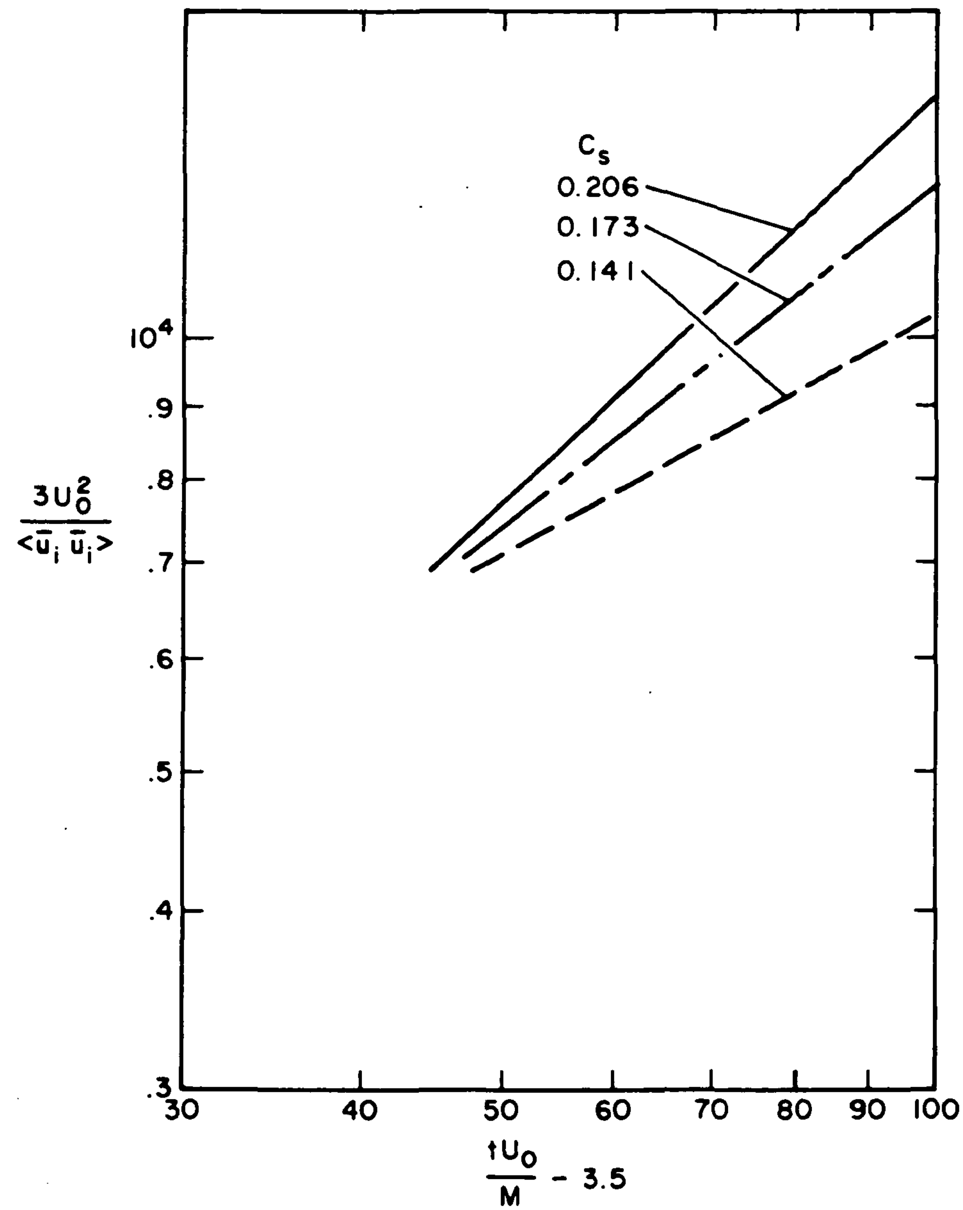


Fig. 4.9. Sensitivity of Filtered Mean Square Velocity Decay Rate to Smagorinsky Model Constant, C_s 16x16x16 Mesh: $\Delta_A = 2\Delta$; $\Delta = 1.5$ cm

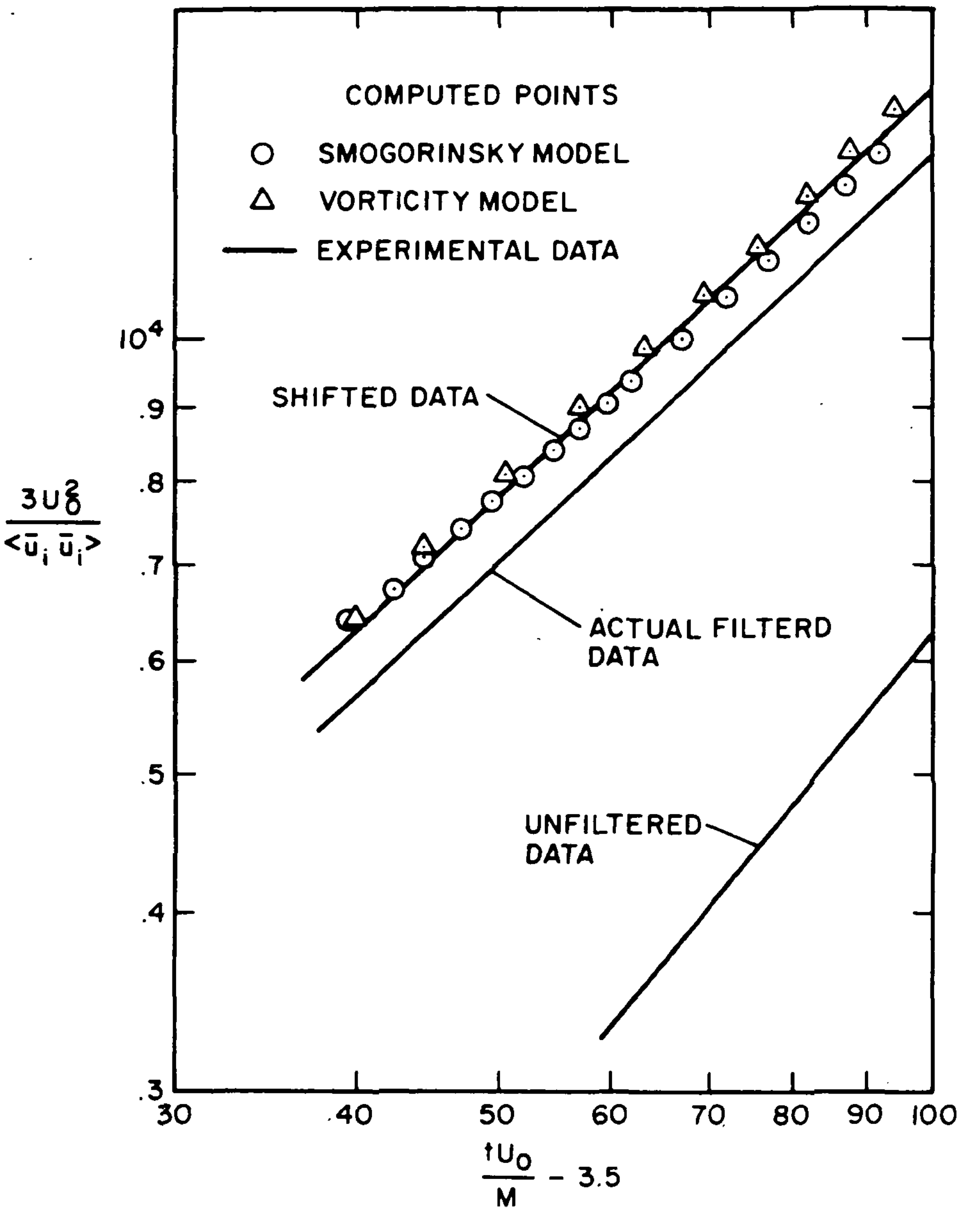


Fig. 4.10. Decay of Mean Square Filtered Velocity
16x16x16 Mesh: $\Delta_A = 2\Delta$; $\Delta = 1.5$ cm

< > : Average Over all Space

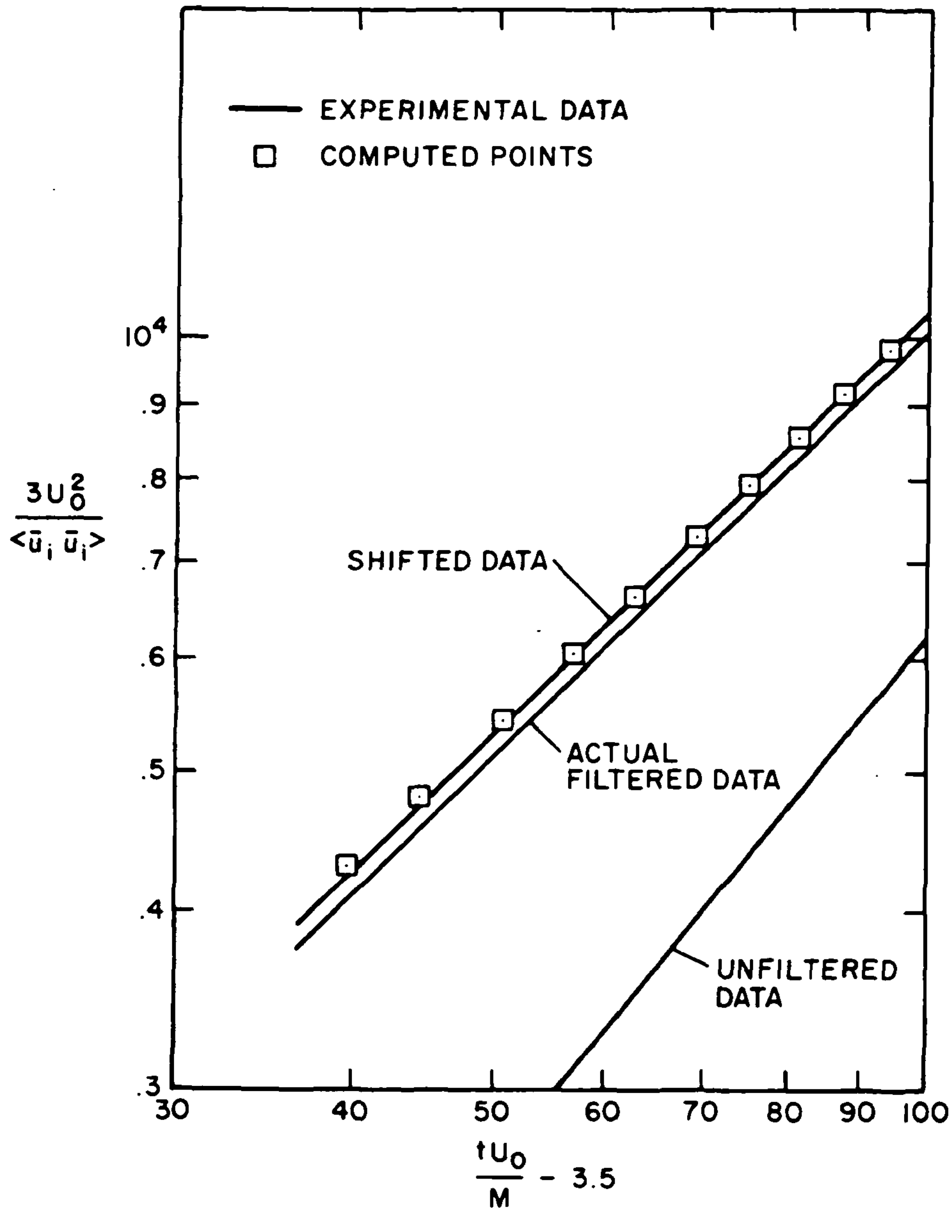


Fig. 4.11. Decay of Mean Square Filtered Velocity. 32x32x32 Mesh: Smagorinsky Model:

$\Delta_A = 2\Delta \;;\; \Delta = 1.0$ cm

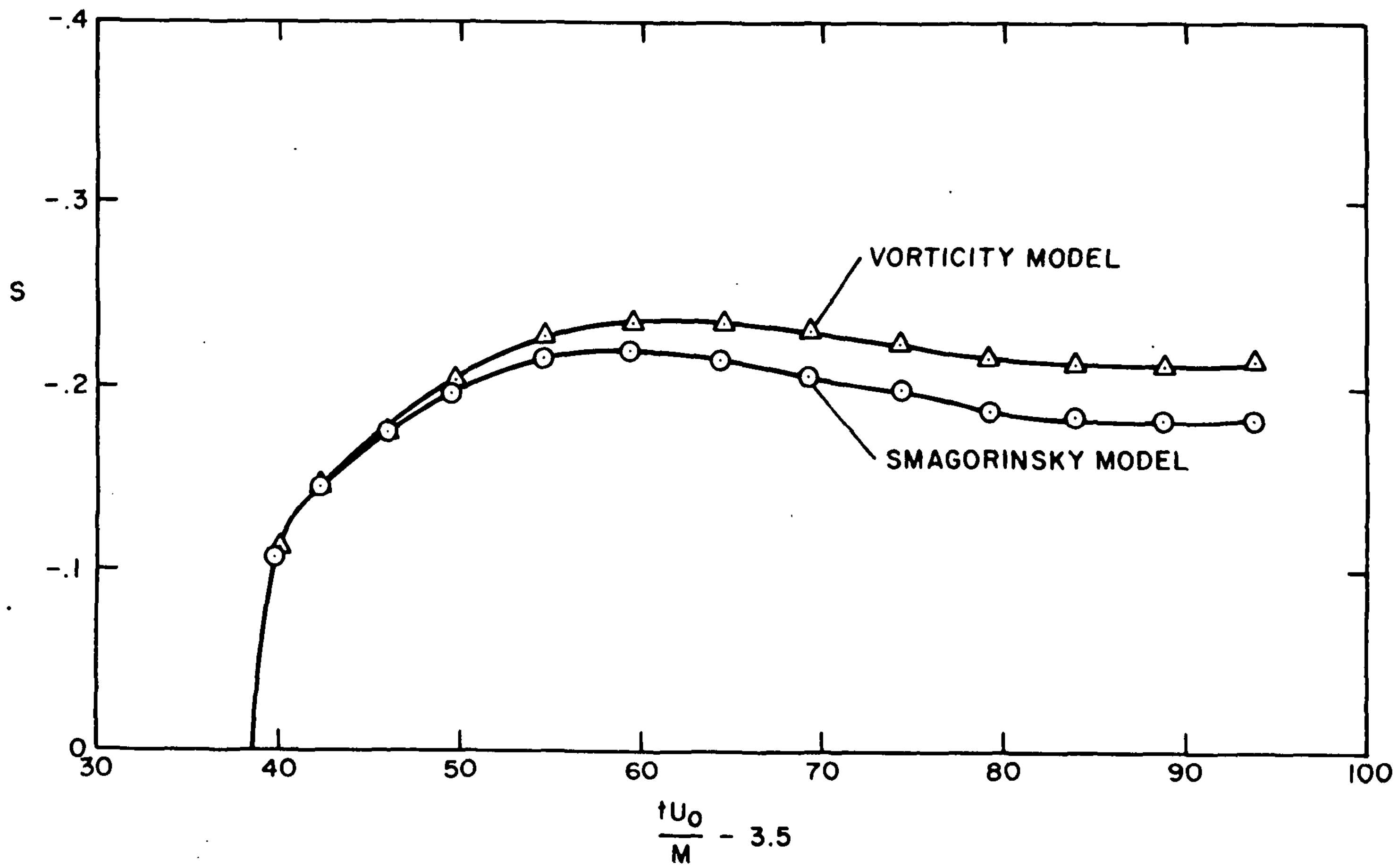


Fig. 4.12. Skewness: $S = \langle(\partial\bar{u}/\partial x)^3\rangle / \langle(\partial\bar{u}/\partial x)^2\rangle^{3/2}$

16x16x16 Mesh: $\Delta_A = 2\Delta$; $\Delta = 1.5$ cm

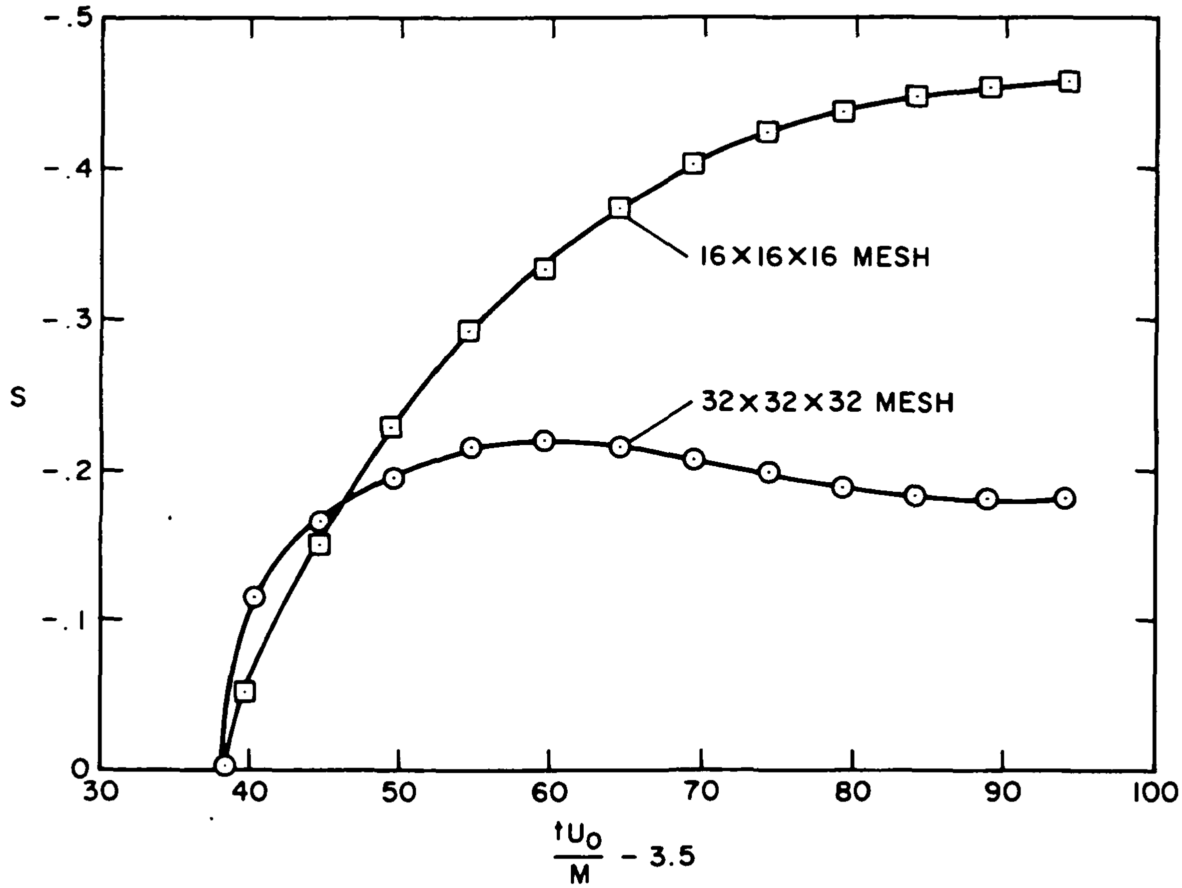


Fig. 4.13 Comparison of Skewness, S . (see Fig. 4.12 for Definition of S): Smagorinsky Model:

$\Delta_A = 2\Delta$

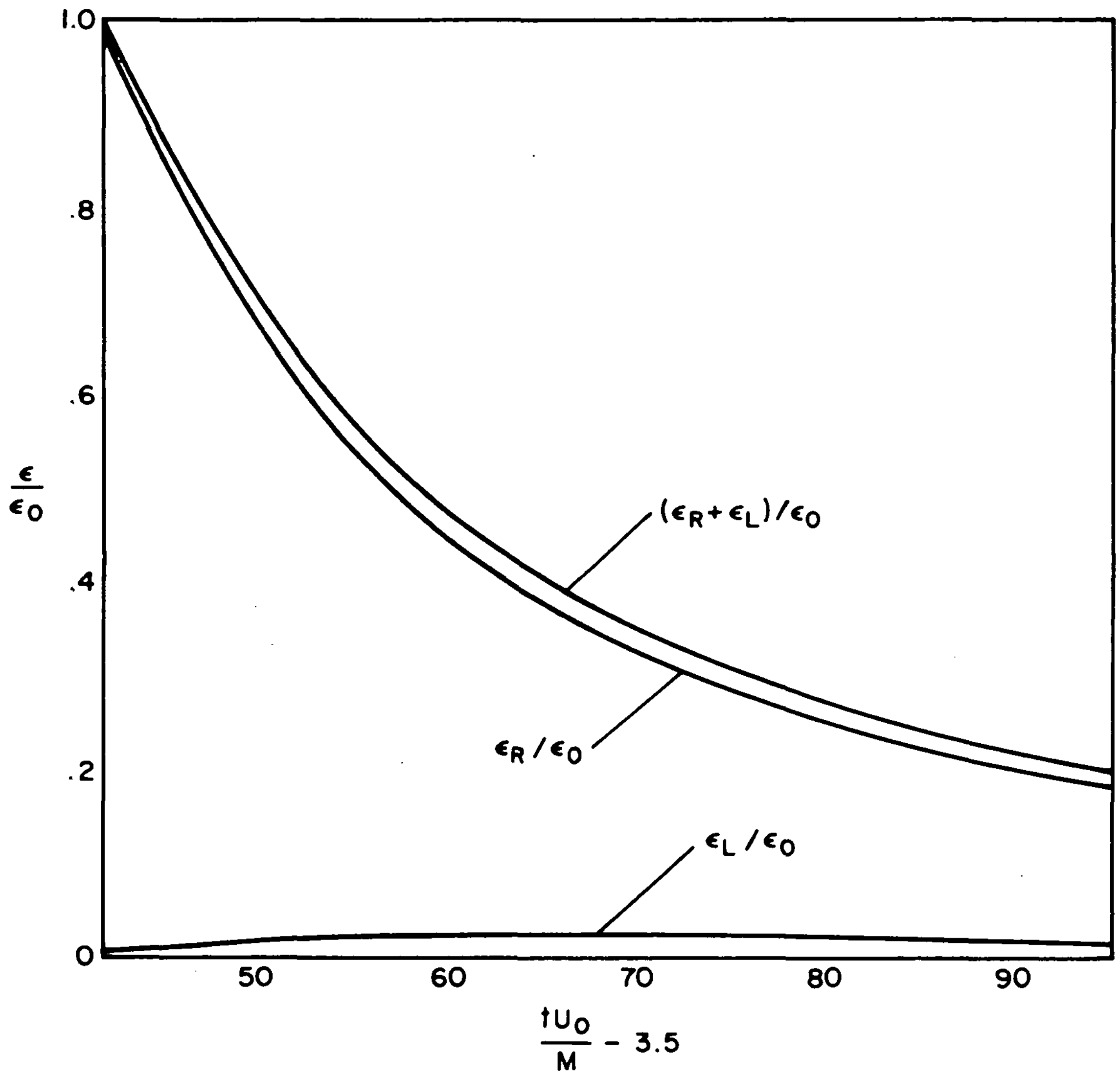


Fig. 4.14. Components of Dissipation.

16x16x16 Mesh: $\Delta_A = 2\Delta$; $\Delta = 1.5$ cm

Vorticity Model.

$\varepsilon_o = \varepsilon_R + \varepsilon_L$ at $(U_o t/M - 3.5) = 42$

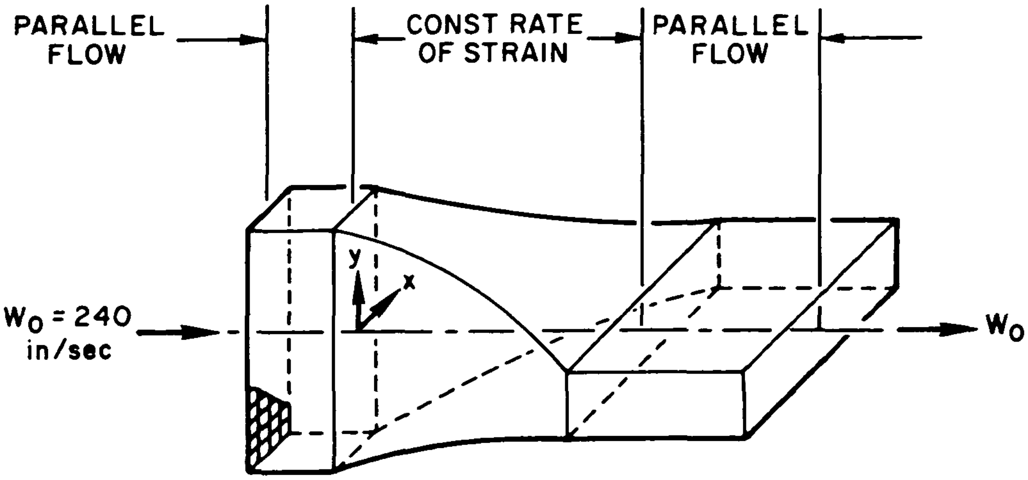


Fig. 5.1a. Schematic of Wind Tunnel Producing Constant Rate of Strain in x-y Plane

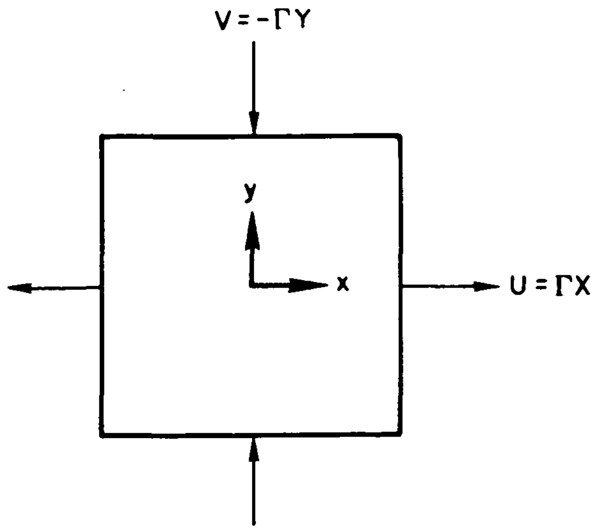


Fig. 5.1b. Equivalent Representation of the Plane Strain in a Box $\Gamma = 1.457$

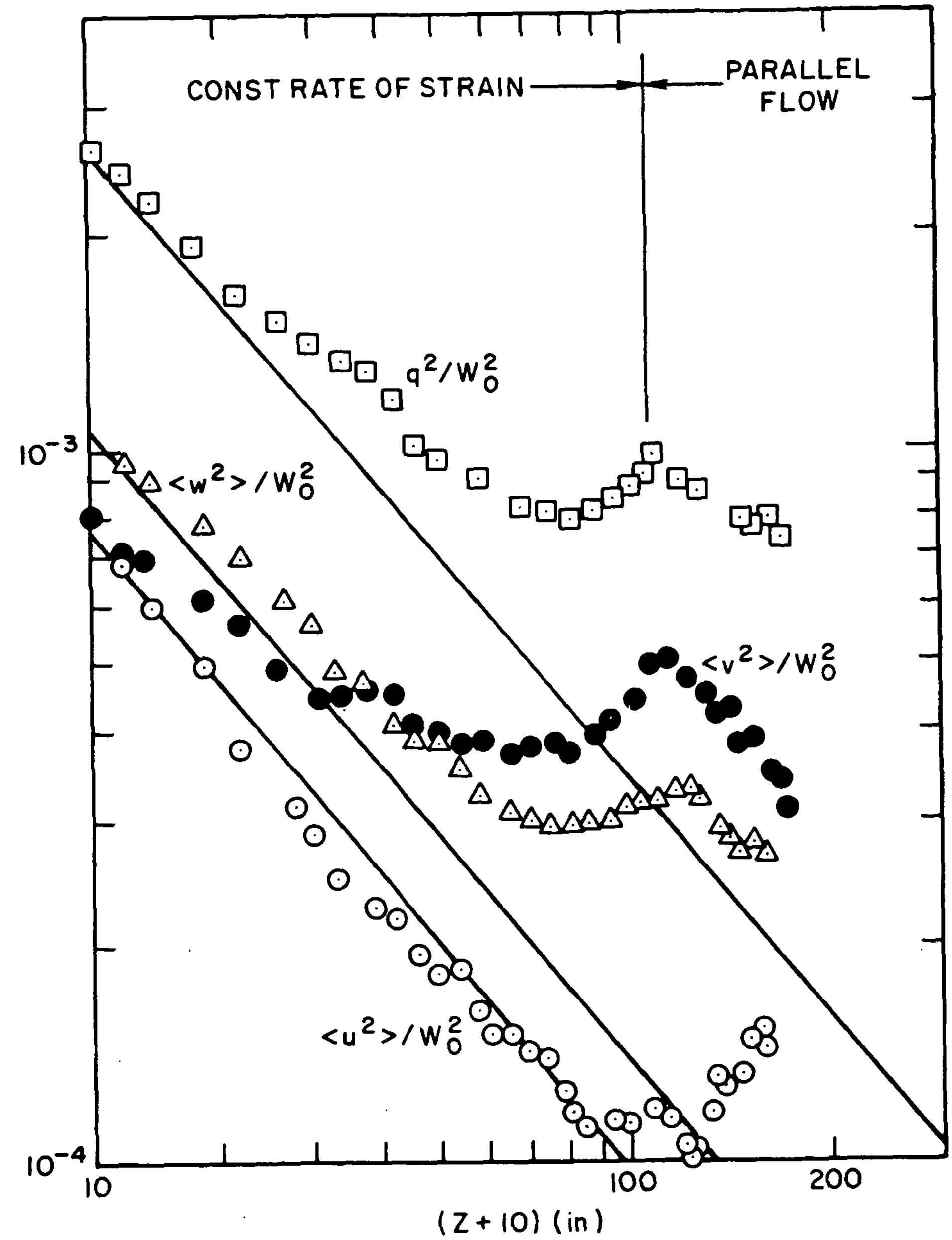


Fig. 5.2. Tucker-Reynolds Flow.
Turbulent Intensities. The solid lines are for decaying turbulence in the absence of strain. Z = Downstream distance from the beginning of strained section.

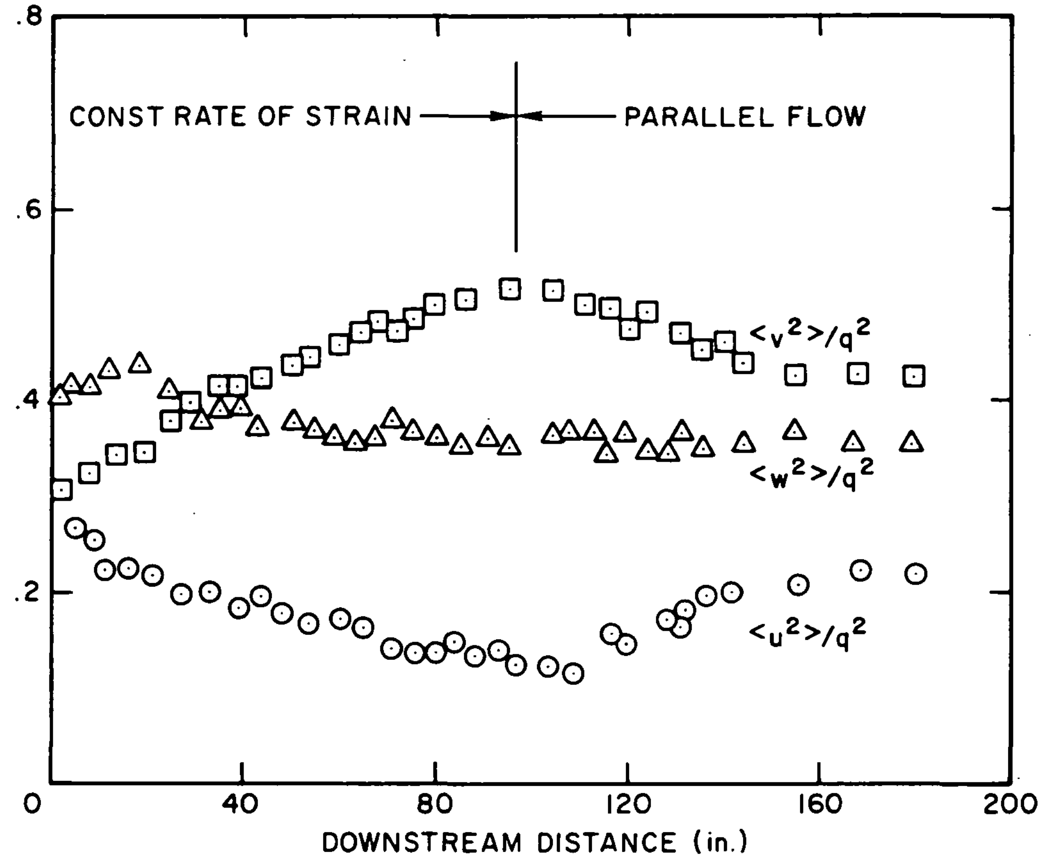


Fig. 5.3. Tucker-Reynolds Flow
Turbulent Energy Ratios

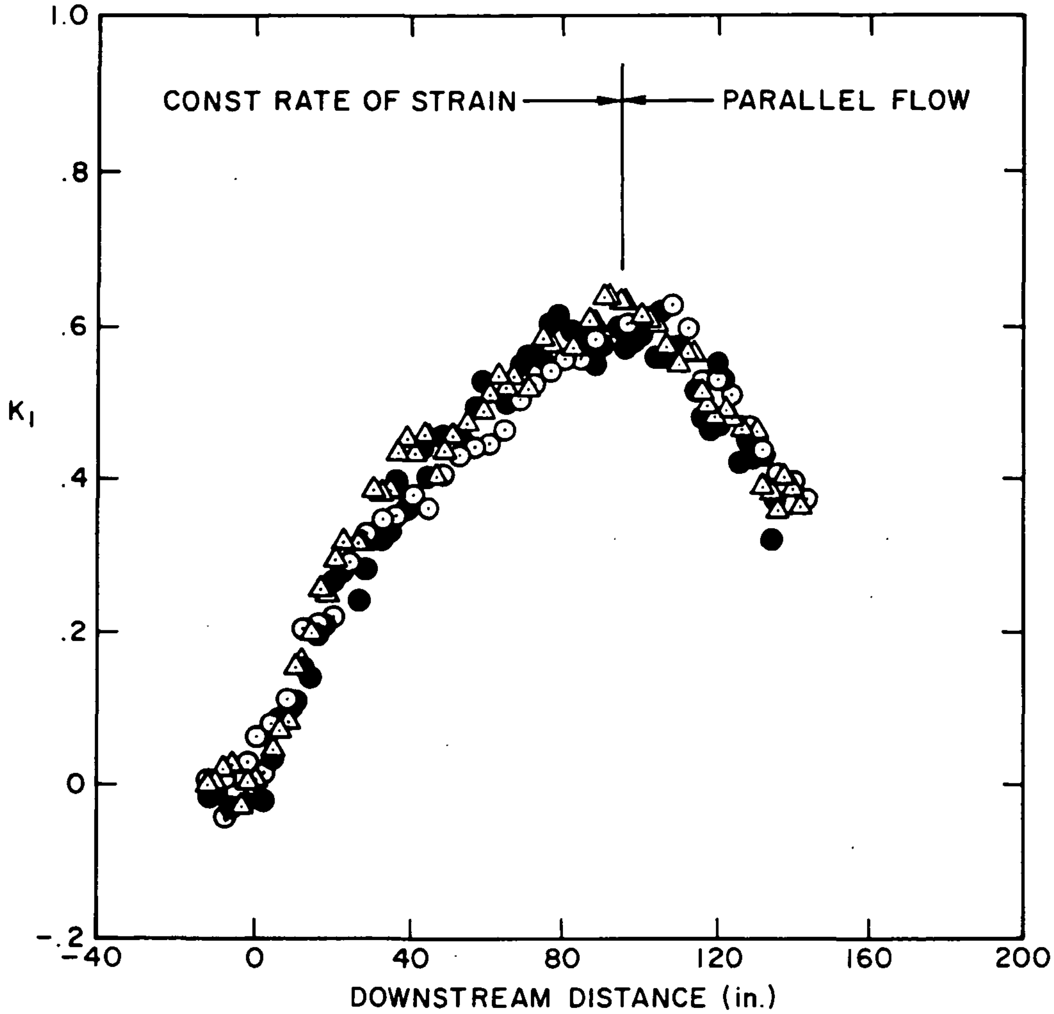


Fig. 5.4. Tucker-Reynolds Flow
Change in Structural Parameter, K_1

$$K_1 = \frac{\langle \bar{v}^2 \rangle - \langle \bar{u}^2 \rangle}{\langle \bar{v}^2 \rangle + \langle \bar{u}^2 \rangle}$$

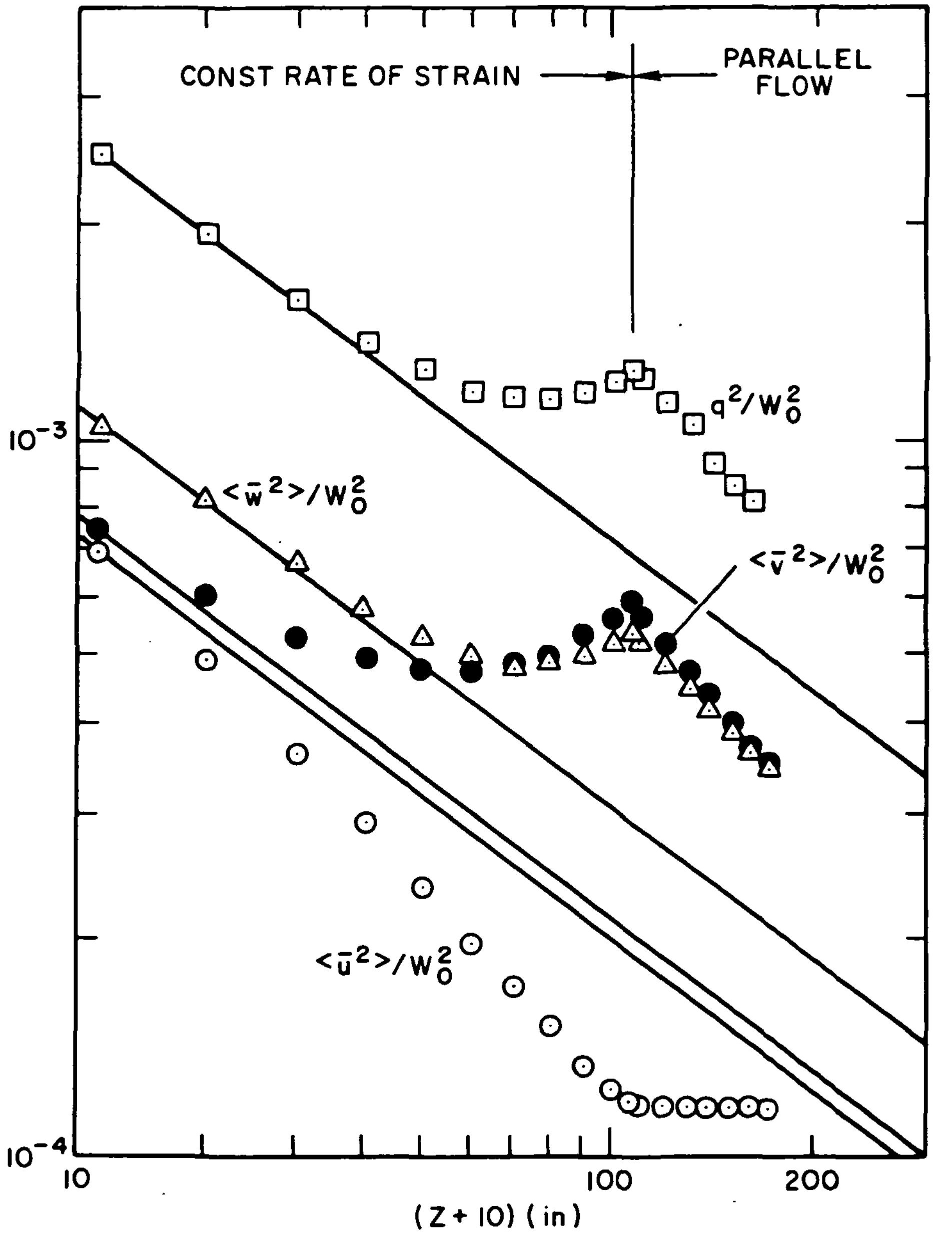


Fig. 5.5. Simulation of the Tucker-Reynolds Flow Turbulent Intensities. The solid lines are for decaying turbulence in the absence of strain. (Anisotropic Initial Condition)

$Z = W_o t$

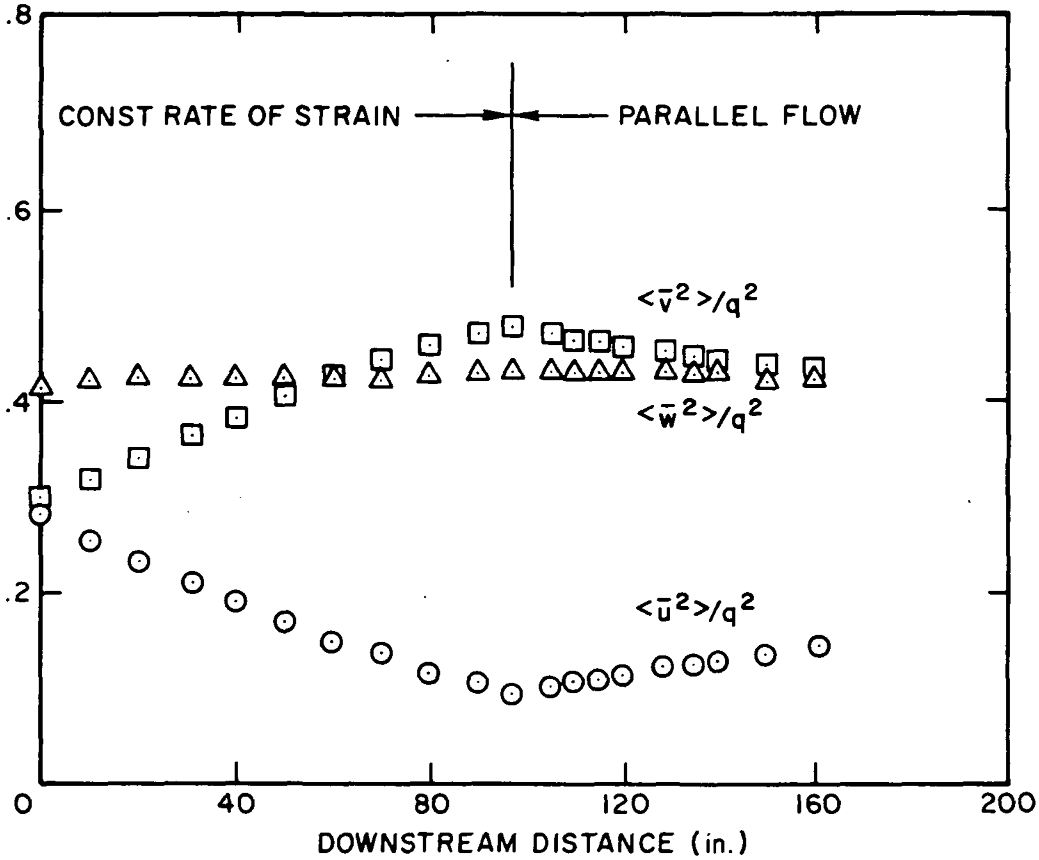


Fig. 5.6. Simulation of the Tucker-Reynolds Flow Turbulent Energy Ratios under the Plane Strain and the Return to Isotropy in Parallel Flow. (Anisotropic Initial Condition)

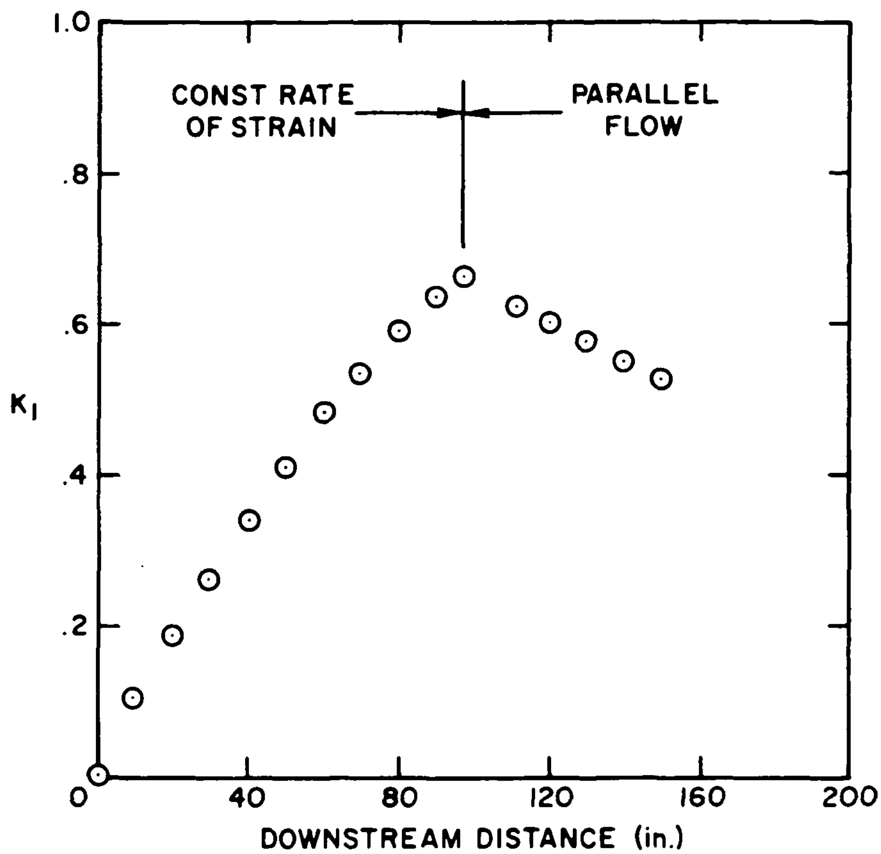


Fig. 5.7. Simulation of the Tucker-Reynolds Flow Change in Structural Parameter. (Anisotropic Initial Condition)

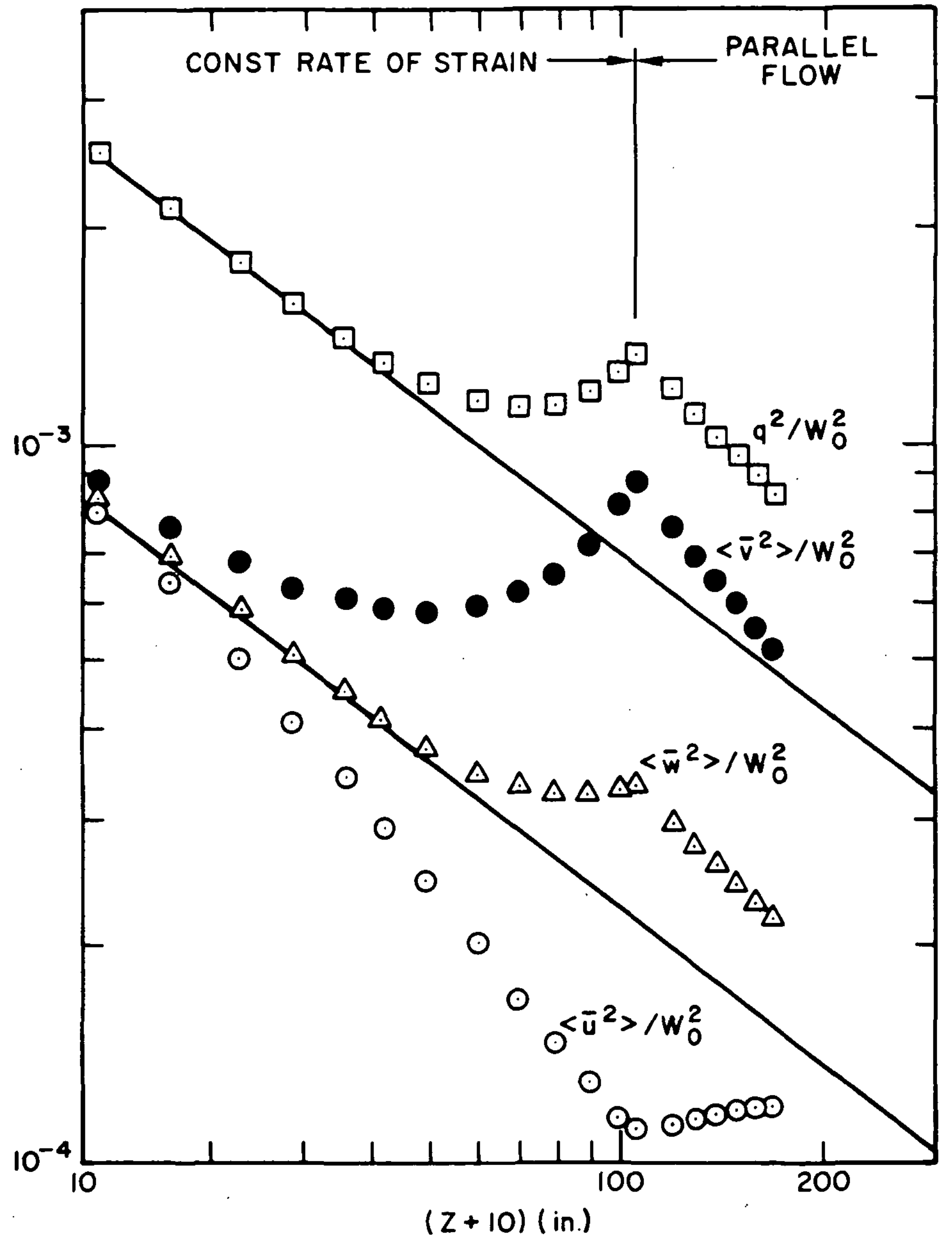


Fig. 5.8. Simulation of the Tucker-Reynolds Flow Turbulent Intensities. The solid lines are for decaying turbulence in the absence of strain. (Isotropic Initial Condition)

$Z = W_o t$

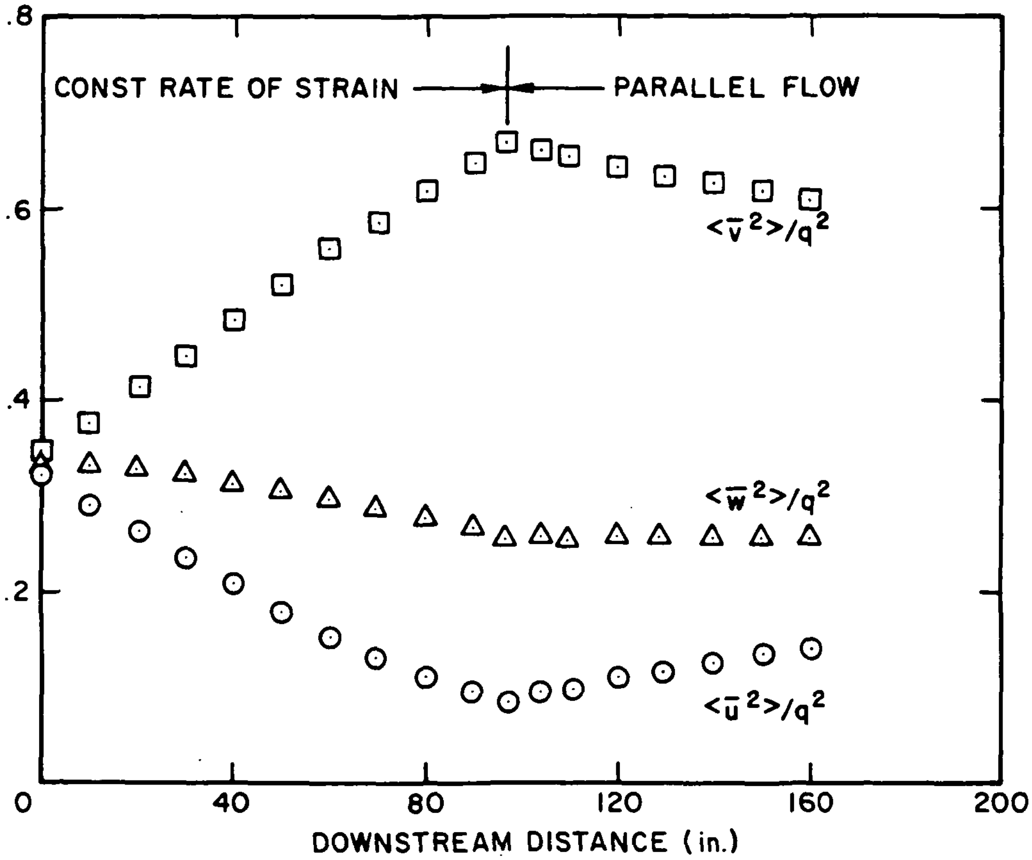


Fig. 5.9. Simulation of the Tucker-Reynolds Flow Turbulent Energy Ratios Under the Plane Strain and the Return to Isotropy in Parallel Flow. (Isotropic Initial Condition)

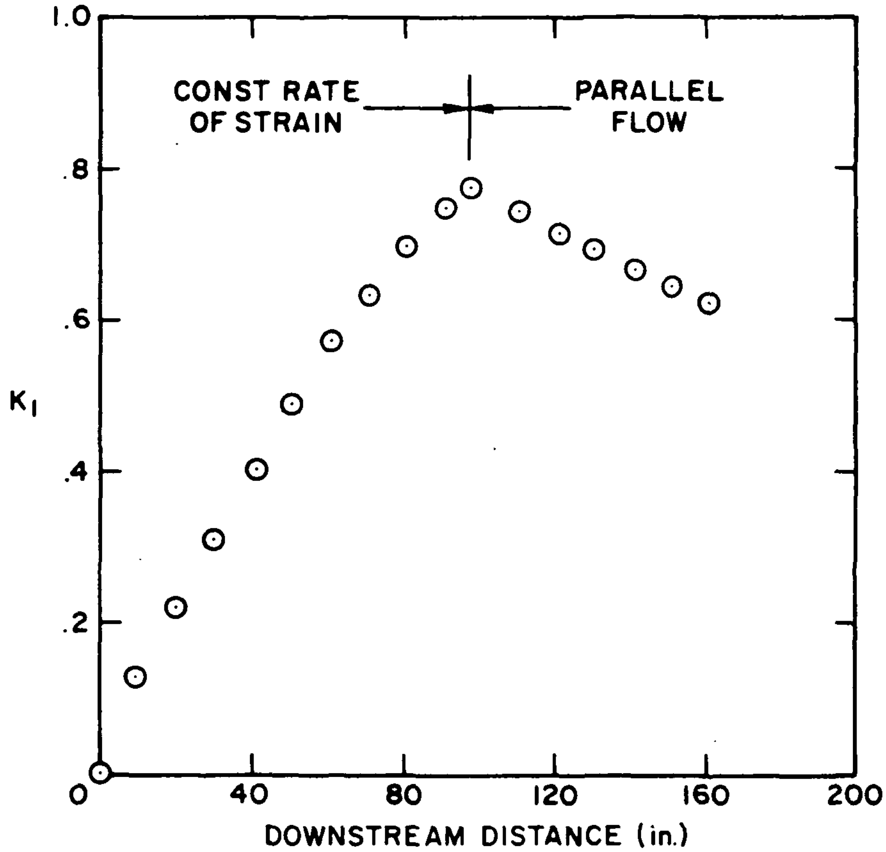


Fig. 5.10. Simulation of the Tucker-Reynolds Flow Change in Structural Parameter, K_1 (Isotropic Initial Condition)

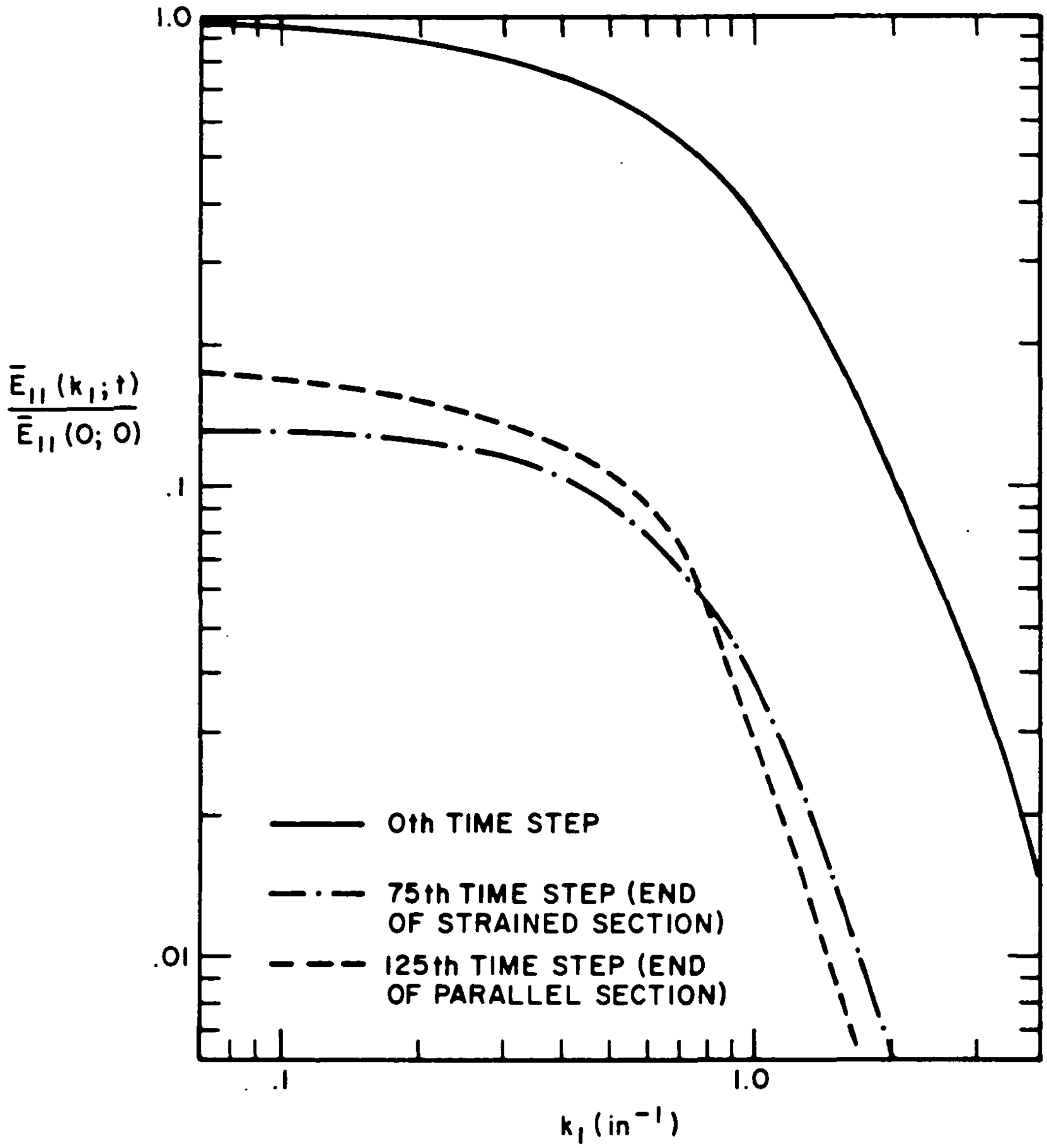


Fig. 5.11a. One-Dimensional Energy Spectra
(Isotropic Initial Condition)

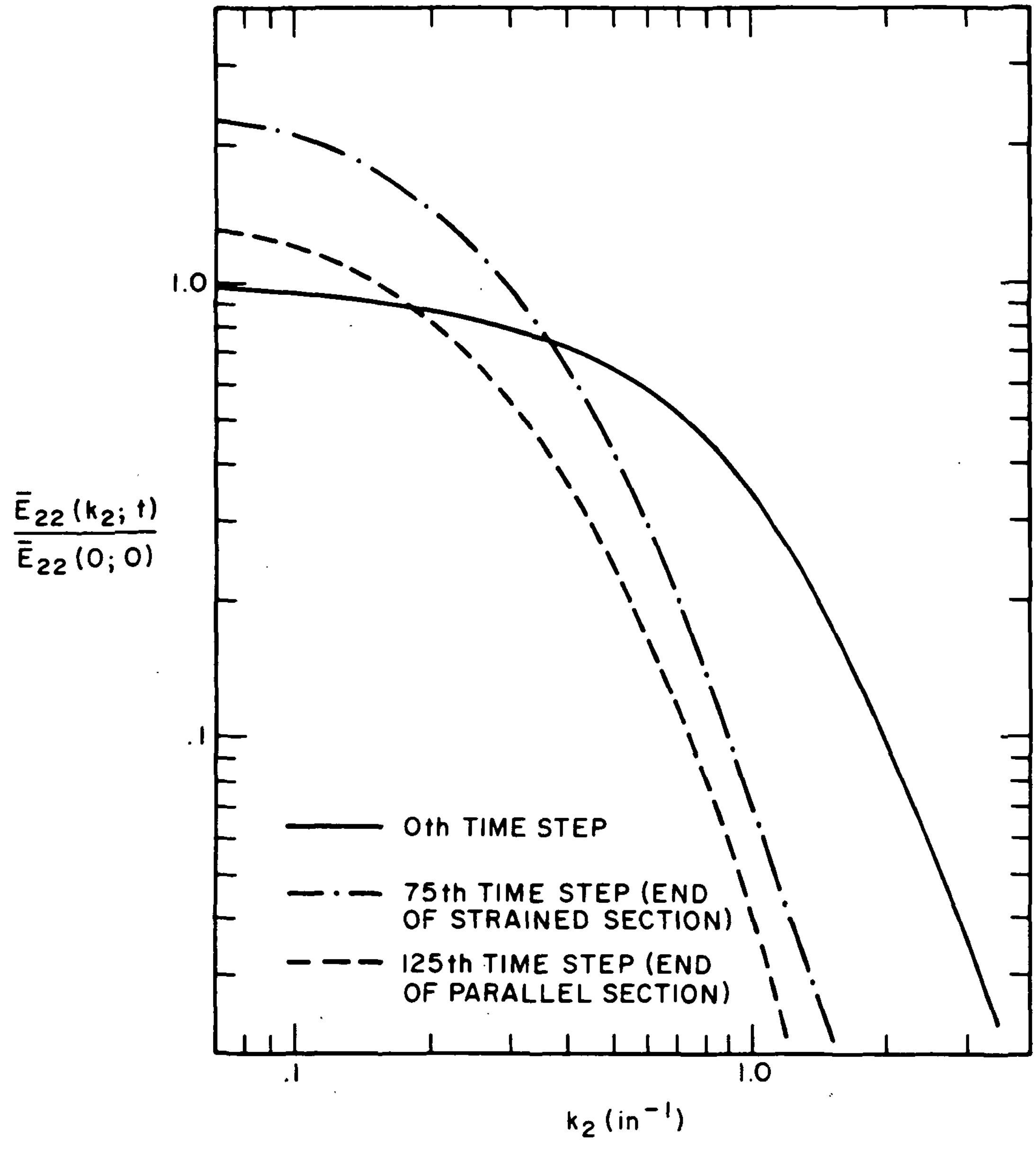


Fig. 5.11b. One-Dimensional Energy Spectra

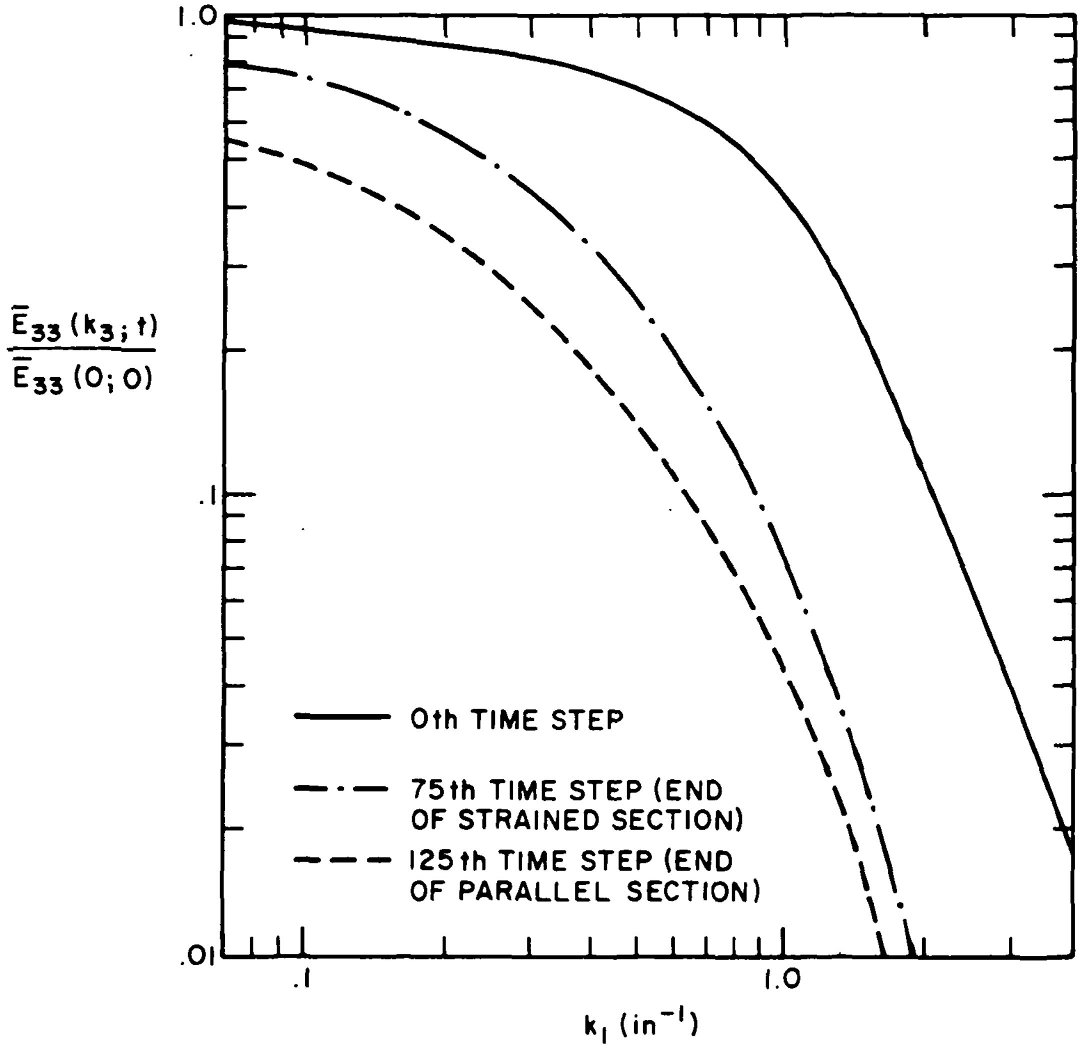


Fig. 5.11c. One-Dimensional Energy Spectra

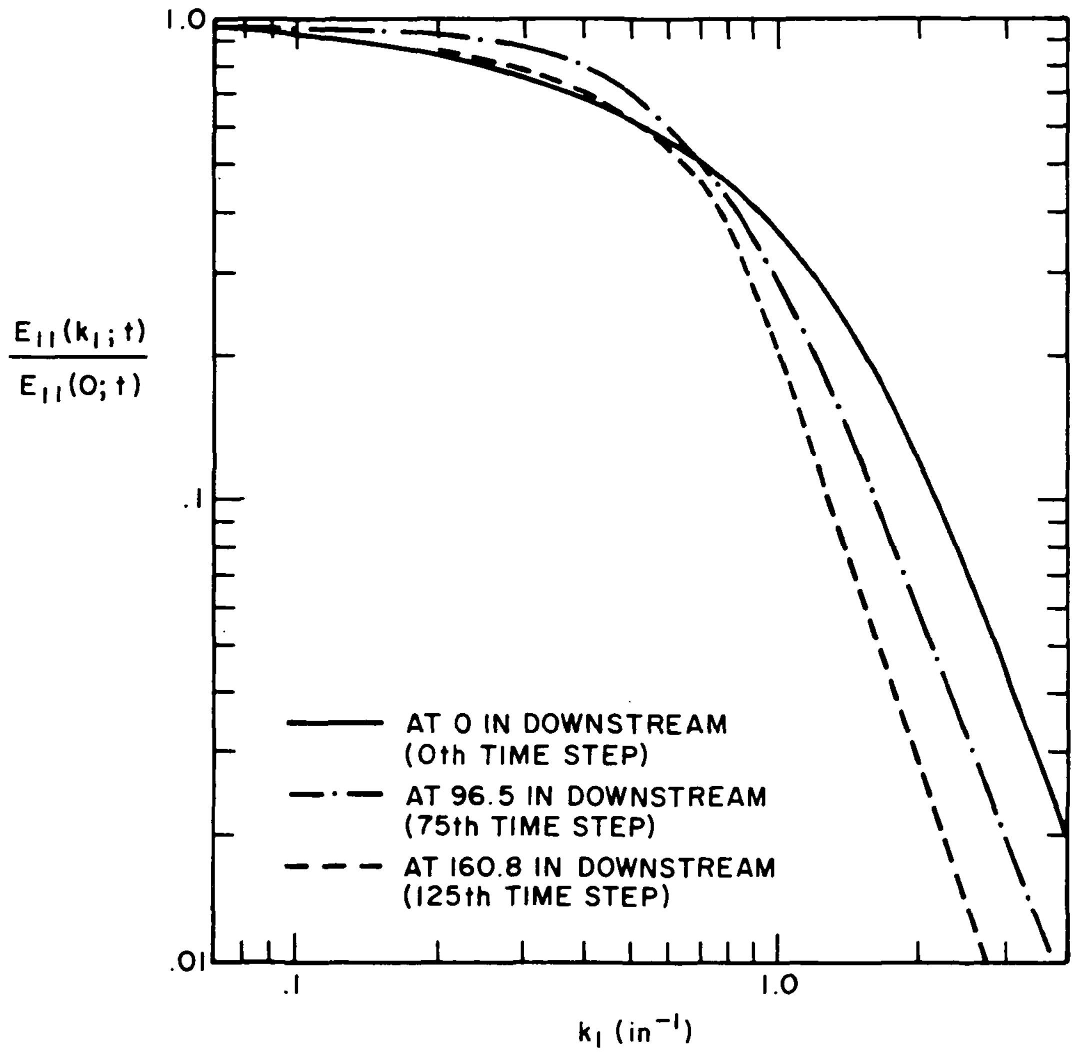


Fig. 5.12a. One-Dimensional Energy Spectra at Three Different Downstream Locations (Isotropic Initial Condition).

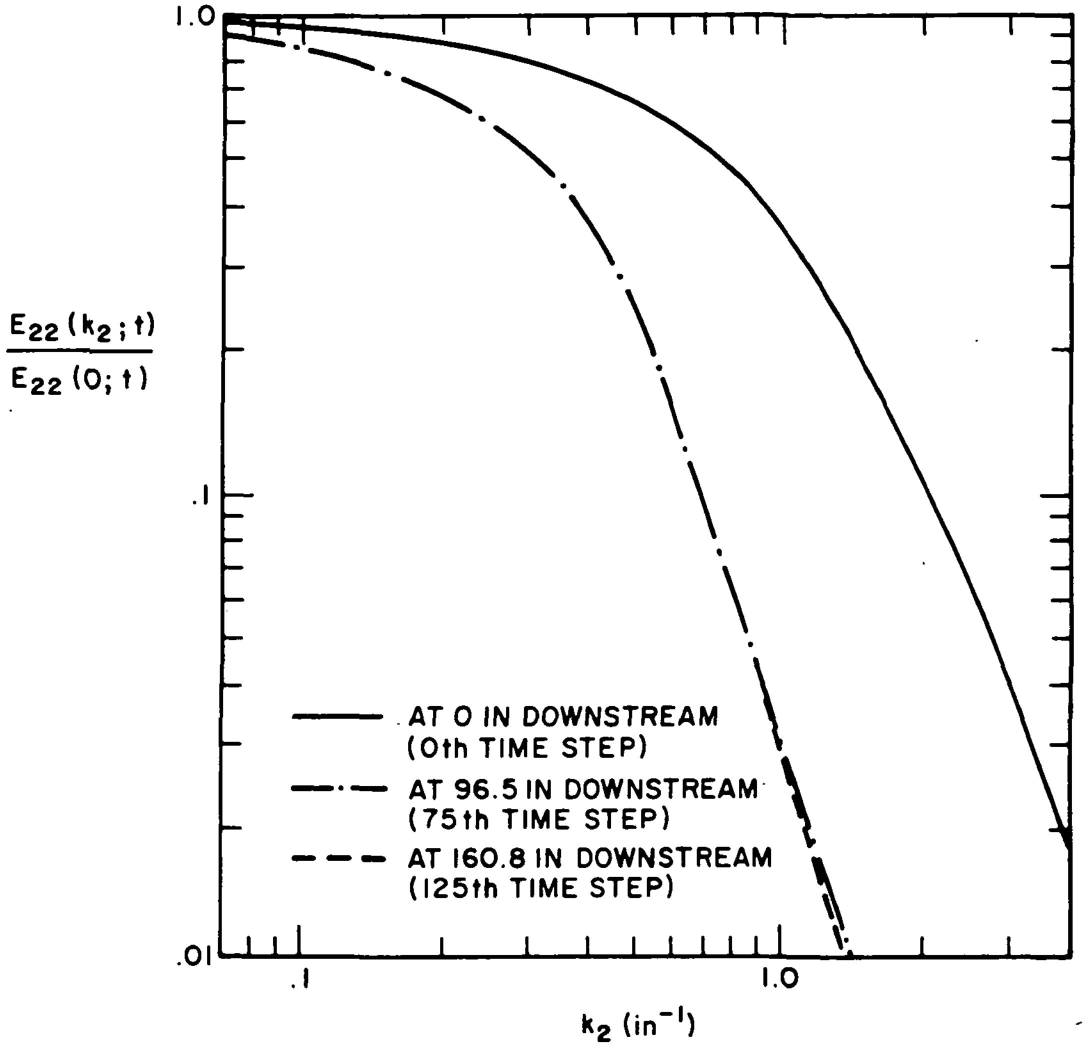


Fig. 5.12b. One-Dimensional Energy Spectra

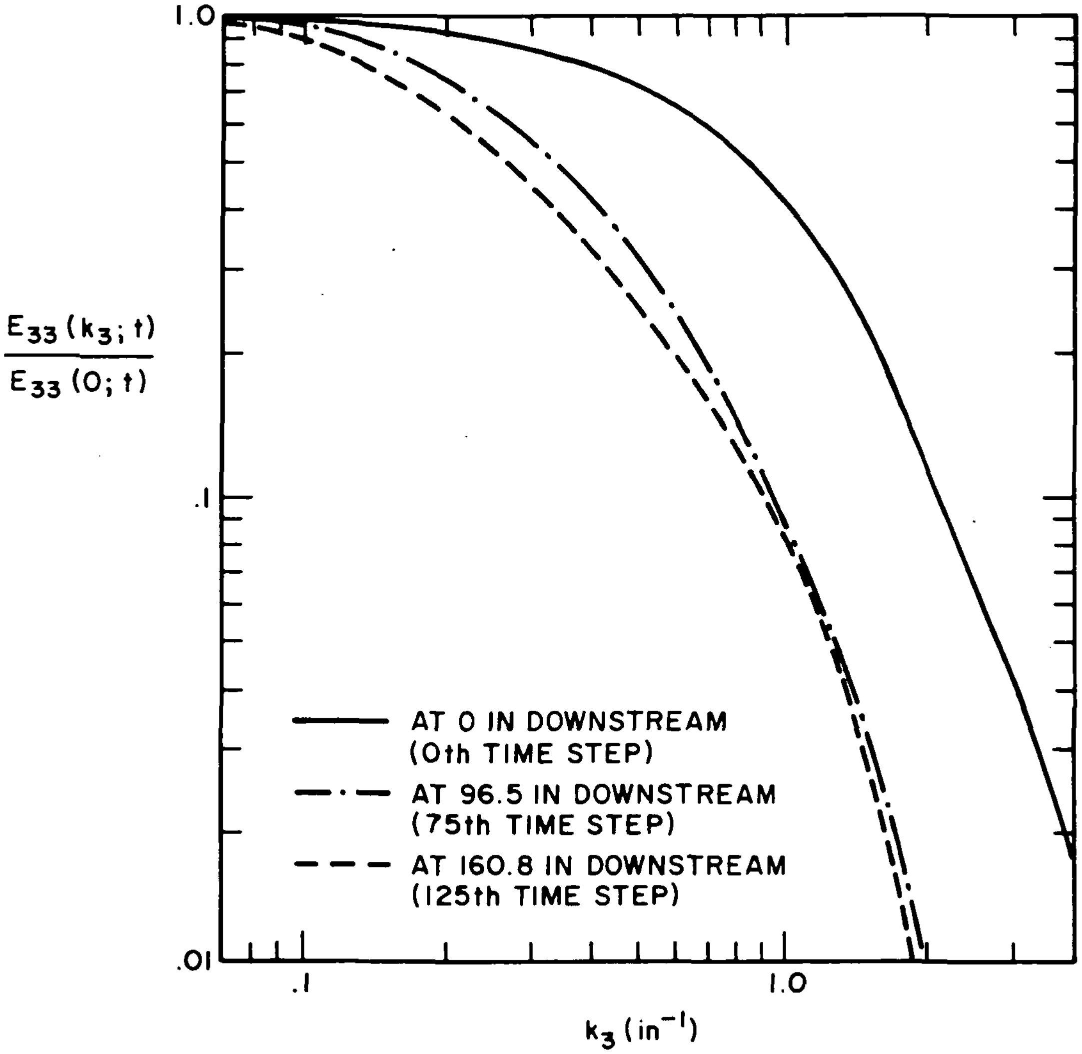


Fig. 5.12c. One-Dimensional Energy Spectra

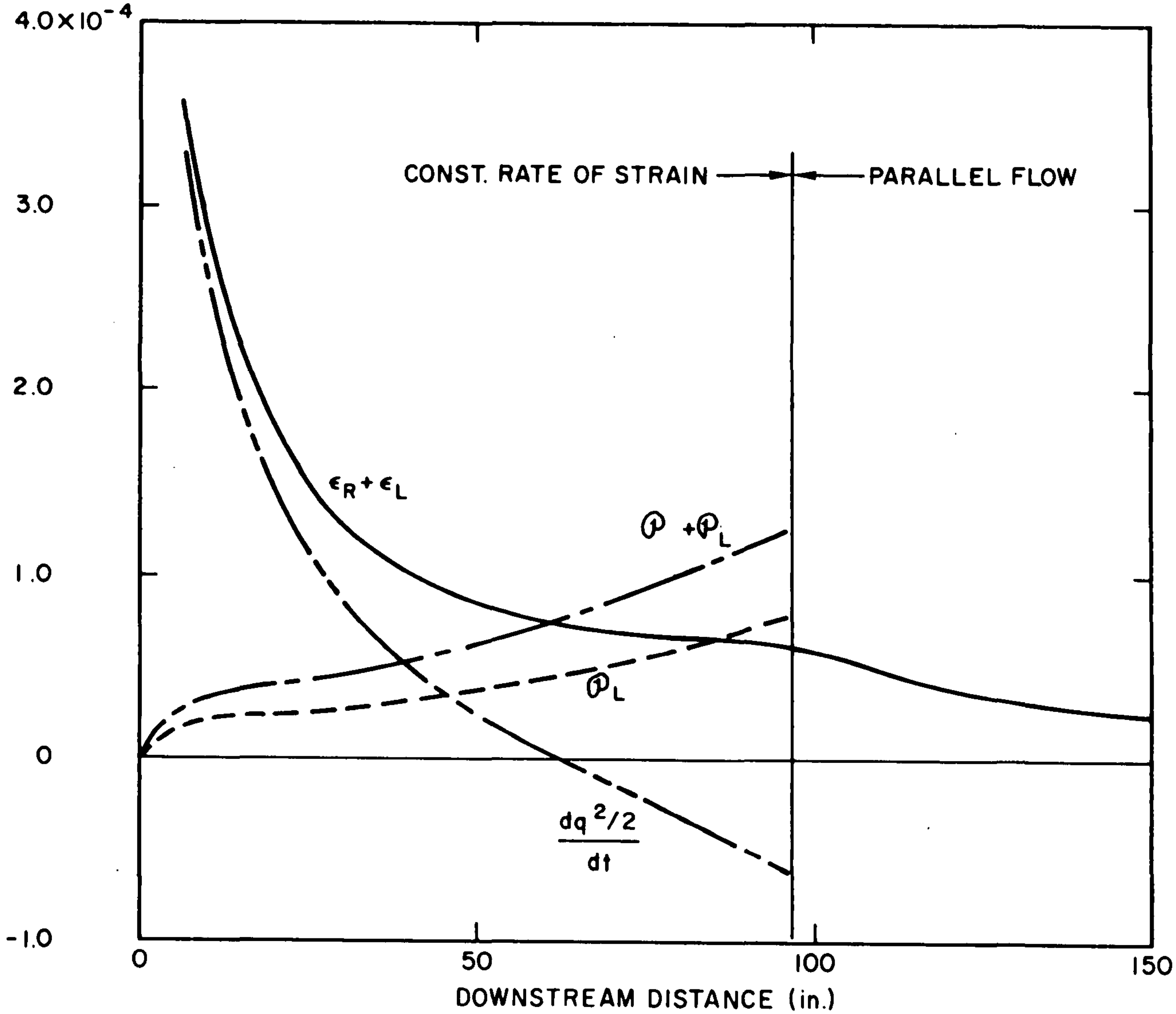


Fig. 5.13. Filtered Energy Balance
Each Term is Normalized by W_o and Computational Box Size $L = N\Delta$
(see Eqns. 5.22 and 5.23)

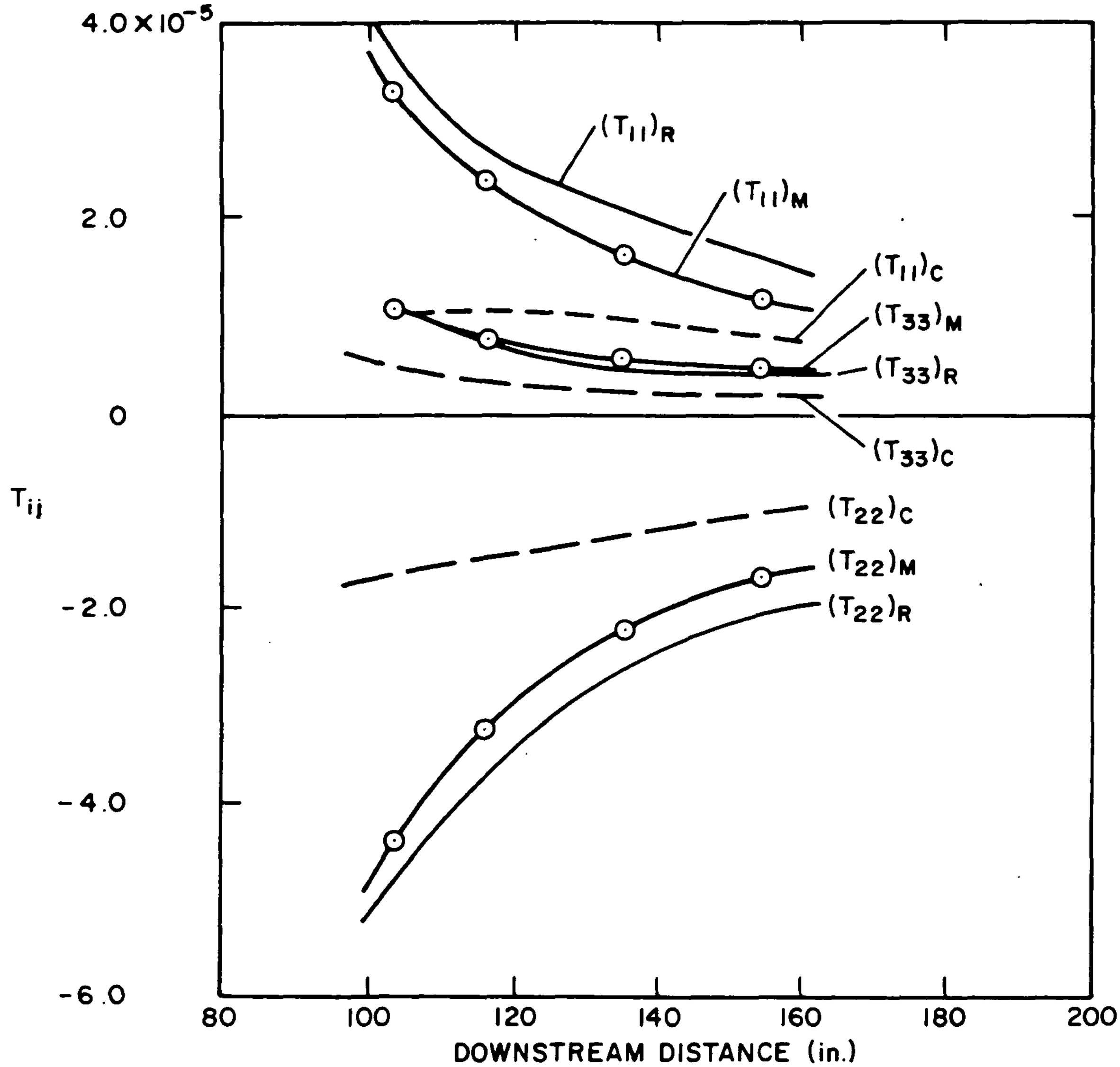


Fig. 5.14. Comparison of the Pressure-Strain Correlations Normalized by W_o and Computational Box Size, $L = N\Delta$ (see Eqns. 5.27, 5.28 and 5.29)

APPENDIX I

ON THE FOURTH ORDER CONSERVATIVE SPACE DIFFERENCING SCHEME

To explain the difference formula used here in detail, consider the following equations.

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = 0 \tag{A.1}$$

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{A.2}$$

$u_i \cdot$ (A.1)

$$u_i \frac{\partial u_i}{\partial t} + u_i \frac{\partial}{\partial x_j} u_i u_j = 0 \tag{A.3}$$

$$\frac{\partial}{\partial t}\left(\frac{u_i u_i}{2}\right) + \frac{1}{2}\frac{\partial}{\partial x_j} u_i u_i u_j + \frac{u_i u_i}{2}\frac{\partial u_j}{\partial x_j} = 0 \tag{A.4}$$

Integrating (A.4) over all space

$$\frac{d}{dt}\int_v \frac{u_i u_i}{2}\, dv = -\int_v \frac{u_i u_i}{2}\frac{\partial u_j}{\partial x_j}\, dv \tag{A.5}$$

If $\operatorname{div} \underline{u} = 0$,

$$\frac{d}{dt}\int_v \frac{u_i u_i}{2}\, dv = 0 \tag{A.6}$$

Now look at difference form of (A.3)

$$\frac{\delta}{\delta t}\frac{u_i u_i}{2} = -u_i \frac{\delta}{\delta x_j} u_i u_j \tag{A.7}$$

Summing over all mesh points

$$\sum \frac{\delta}{\delta t}\frac{u_i u_i}{2} = -\sum u_i \frac{\delta}{\delta x_j} u_i u_j \tag{A.8}$$

The RHS of (A.8) has to be zero as in (A.6). Therefore $\delta u_i u_j / \delta x_j$ must be devised such that RHS summation of (A.8) goes to zero thus conserving the total kinetic energy. The difference formula for $\delta u_i u_j / \delta x_j$ depends on the type of mesh and the number of neighboring points used.

The following is a fourth order energy conserving scheme using five points in one direction. Following usual convention

$$
\begin{aligned}
\overline{f}^{x} &= \frac{1}{\Delta}\left[f(x+\frac{\Delta}{2}) + f(x-\frac{\Delta}{2})\right] \\
f_x &= \frac{1}{\Delta}\left[f(x+\frac{\Delta}{2}) - f(x-\frac{\Delta}{2})\right]
\end{aligned}
\qquad \text{(A.9)}
$$

Note that

$$
\frac{\partial f}{\partial x} = \overline{f_x}^{x} + 0(\Delta^2) = \frac{f_{i+1} - f_{i-1}}{2\Delta} + 0(\Delta^2)
$$

$$
\frac{\partial f}{\partial x} = \frac{4}{3}\,\overline{f_x}^{x} - \frac{1}{3}\,\overline{f_{2x}}^{2x} + 0(\Delta^4) \qquad \text{(A.10)}
$$

$$
= \frac{1}{12\Delta}\,(f_{i-2} - 8f_{i-1} + 8f_{i+1} - f_{i+2}) + 0(\Delta^4)
$$

Now fourth order momentum and energy conserving form of (A.1) and (A.2) are

$$
\frac{\partial u_i}{\partial t} + \frac{4}{3}\,(\overline{u_i}^{x_j}\overline{u_j}^{x_j})_{x_j} - \frac{1}{3}\,(\overline{u_i}^{2x_j}\overline{u_j}^{2x_j})_{2x_j} = 0 \qquad \text{(A.11)}
$$

$$
\frac{4}{3}\overline{(u_i)_{x_i}}^{x_i} - \frac{1}{3}\overline{(u_i)_{2x_i}}^{2x_i} = 0 \qquad \text{(A.12)}
$$

To show that these are indeed energy conserving, work with the expanded form of (A.11).

$$\frac{\delta u_\ell}{\delta t} + \left\{ \frac{4}{3}(\bar{u}_\ell^x \bar{u}^x)_x - \frac{1}{3}(\bar{u}_\ell^{2x} \bar{u}^{2x})_{2x} \right\}$$

$$+ \left\{ \frac{4}{3}(\bar{u}_\ell^y \bar{v}^y)_y - \frac{1}{3}(\bar{u}_\ell^{2y} \bar{v}^{2y})_{2y} \right\}$$

$$+ \left\{ \frac{4}{3}(\bar{u}_\ell^z \bar{w}^z)_z - \frac{1}{3}(\bar{u}_\ell^{2z} \bar{w}^{2z})_{2z} \right\} = 0 \qquad \text{(A.13)}$$

where u_ℓ is either u, v or w . u_ℓ x (A.13) gives the energy equation. Then sum over all mesh points.

$$\sum u_\ell \frac{\delta u_\ell}{\delta t} = -\sum \left[u_\ell \left\{ \frac{4}{3}(\bar{u}_\ell^x \bar{u}^x)_x - \frac{1}{3}(\bar{u}_\ell^{2x} \bar{u}^{2x})_{2x} \right\} \right.$$

$$\left. + \left\{ v \text{ and } w - \text{component} \right\} \right]$$

$$= -\sum \left[u_\ell \frac{1}{2} \left\{ \frac{4}{3} \quad (\overline{u_\ell u})_x^x + u_\ell \bar{u}_x^x + u(\overline{u_\ell})_x^x \right\} \right.$$

$$- \frac{1}{3} \left((\overline{u_\ell u})_{2x}^{2x} + u_\ell \bar{u}_{2x}^{2x} + u(\overline{u_\ell})_{2x}^{2x} \right) \bigg\}$$

$$\left. + \ldots \right]$$

$$= -\sum \frac{u_\ell}{2} \left[\frac{4}{3} \left\{ (\overline{u_\ell u})_x^x + u(\overline{u_\ell})_x^x \right\} \right.$$

$$\left. - \frac{1}{3} \left\{ (\overline{u_\ell u})_{2x}^{2x} + u(\overline{u_\ell})_{2x}^{2x} \right\} \right]$$

$$+ \text{ v, w, - component}$$

$$-\sum \frac{u_\ell u_\ell}{2} \left\{ \left[\frac{4}{3} \bar{u}_x^x - \frac{1}{3} \bar{u}_{2x}^{2x} \right] + \left[\frac{4}{3} \bar{v}_y^y - \frac{1}{3} \bar{v}_{2y}^{2y} \right] \right.$$

$$\left. + \left[\frac{4}{3} \bar{w}_z^z - \frac{1}{3} \bar{w}_{2z}^{2z} \right] \right\} \qquad \text{(A.14)}$$

The first summation on RHS of (A.14) is zero for periodic boundary condition. Therefore (A.14) becomes

$$\sum u_\ell \frac{\delta u_\ell}{\delta t} = -\sum \frac{u_\ell u_\ell}{2} \left\{ \text{div } u \text{ in 4th order; (A.12)} \right\} \qquad \text{(A.15)}$$

So if the numerical divergence of velocity is zero, the RHS of (A.15) is zero and (A.11) is indeed energy conserving.

The accuracy of the LHS depends on the time advancing scheme and, as mentioned earlier, the Adams-Bashforth predictor method introduces very weak instability on computational mode. To show that (A.11) and (A.12) are really fourth order accurate, work with a typical term:

$$\frac{4}{3}(\bar{v}^x \bar{u}^x)_x - \frac{1}{3}(\bar{v}^{2x} \bar{u}^{2x})_{2x}$$

$$= \frac{1}{3\Delta} \left\{ (vu)_{(k+1)} - (vu)_{(k-1)} + u_{(k)} (v_{(k+1)} - v_{(k-1)}) \right.$$

$$\left. + v_{(k)} (u_{(k+1)} - u_{(k-1)}) \right\} - \frac{1}{24\Delta} \left\{ (vu)_{(k+2)} - (vu)_{(k-2)} \right.$$

$$\left. + u_{(k)} (v_{(k+2)} - v_{(k-2)}) + v_{(k)} (u_{(k+2)} - u_{(k-2)}) \right\}$$

Substitute RHS by Taylor expansion as usual. After some algebra, it can be shown that

$$\frac{4}{3}(\bar{v}^x \bar{u}^x)_x - \frac{1}{3}(\bar{v}^{2x} \bar{u}^{2x})_{2x}$$

$$= \frac{\partial}{\partial x} (vu) - \frac{\Delta^4}{60} \left\{ (vu)^v + v^v + u^v \right\} + \dots \qquad \text{(A.16)}$$

Therefore it is indeed a fourth order scheme. An extension to higher order differencing scheme can be done on the same basic idea. Overall fourth order accuracy is obtained for advective term by using second order energy conserving scheme for the Leonard term.

For convenience, fourth and second order advection term is recapitulated below.

C-2

Fourth order energy conserving scheme:

$$\frac{\partial}{\partial x_j} u_i u_j = \frac{\partial}{\partial x}(u_i u) + \frac{\partial}{\partial y}(u_i v) + \frac{\partial}{\partial z}(u_i w)$$

$$= \left\{\frac{4}{3}(\overline{u_i}^{x}\overline{u}^{x})_x - \frac{1}{3}(\overline{u_i}^{2x}\overline{u}^{2x})_{2x}\right\}$$

$$+ \left\{\frac{4}{3}(\overline{u_i}^{y}\overline{v}^{y})_y - \frac{1}{3}(\overline{u_i}^{2y}\overline{v}^{2y})_{2y}\right\}$$

$$+ \left\{\frac{4}{3}(\overline{u_i}^{z}\overline{w}^{z})_z - \frac{1}{3}(\overline{u_i}^{2z}\overline{w}^{2z})_{2z}\right\} + O(\Delta^4)$$

$$= \left\{\frac{1}{3\Delta}\left[(u_i u)_{(k+1)} - (u_i u)_{(k-1)} + u\left((u_i)_{(k+1)} - (u_i)_{(k-1)}\right)\right.\right.$$

$$\left. + (u_i)\left(u_{(k+1)} - u_{(k-1)}\right)\right] - \frac{1}{24\Delta}\left[(u_i u)_{(k+2)} - (u_i u)_{(k-2)}\right.$$

$$\left.\left. + u\left((u_i)_{(k+2)} - (u_i)_{(k-2)}\right) + u_i\left(u_{(k+2)} - u_{(k-2)}\right)\right]\right\}$$

$$+ \left\{\frac{1}{3\Delta}\left[(u_i v)_{(\ell+1)} - (u_i v)_{(\ell-1)} + v\left((u_i)_{(\ell+1)} - (u_i)_{(\ell-1)}\right)\right.\right.$$

$$\left. + u_i\left(v_{(\ell+1)} - v_{(\ell-1)}\right)\right] - \frac{1}{24\Delta}\left[(u_i v)_{(\ell+2)} - (u_i v)_{(\ell-2)}\right.$$

$$\left.\left. + v\left((u_i)_{(\ell+2)} - (u_i)_{(\ell-2)}\right) + u_i\left(v_{(\ell+2)} - v_{(\ell-2)}\right)\right]\right\}$$

$$\left\{\frac{1}{3\Delta}\left[(u_i w)_{(m+1)} - (u_i w)_{(m-1)} + w\left((u_i)_{(m+1)} - (u_i)_{(m-1)}\right)\right.\right.$$

$$\left. + u_i\left(w_{(m+1)} - w_{(m-1)}\right)\right] - \frac{1}{24\Delta}\left[(u_i w)_{(m+2)} - (u_i w)_{(m-2)}\right.$$

$$\left.\left. + w\left((u_i)_{(m+2)} - (u_i)_{(m-2)}\right) + u_i\left(w_{(m+2)} - w_{(m-2)}\right)\right]\right\} \tag{A.17}$$

$$+ O(\Delta^4)$$

Second order scheme:

$$\frac{\partial}{\partial x_j}(u_i u_j) = \frac{\partial}{\partial x} u_i u + \frac{\partial}{\partial y} u_i v + \frac{\partial}{\partial z} u_i w$$

$$= (\bar{u}_i^x \bar{u}^x)_x + (\bar{u}_i^y \bar{v}^y)_y + (\bar{u}_i^z \bar{w}^z)_z + 0(\Delta^2)$$

$$= \frac{1}{4\Delta} \Big\{ (u_i u)_{(k+1)} - (u_i u)_{(k-1)} + u \left((u_i)_{(k+1)} - (u_i)_{(k-1)} \right)$$

$$+ u_i \left(u_{(k+1)} - u_{(k-1)} \right) + (u_i v)_{(\ell+1)} - (u_i v)_{(\ell-1)}$$

$$+ v \left((u_i)_{(\ell+1)} - (u_i)_{(\ell-1)} \right) + u_i \left(v_{(\ell+1)} - v_{(\ell-1)} \right)$$

$$+ (u_i w)_{(m+1)} - (u_i w)_{(m-1)} + w \left((u_i)_{(m+1)} - (u_i)_{(m-1)} \right)$$

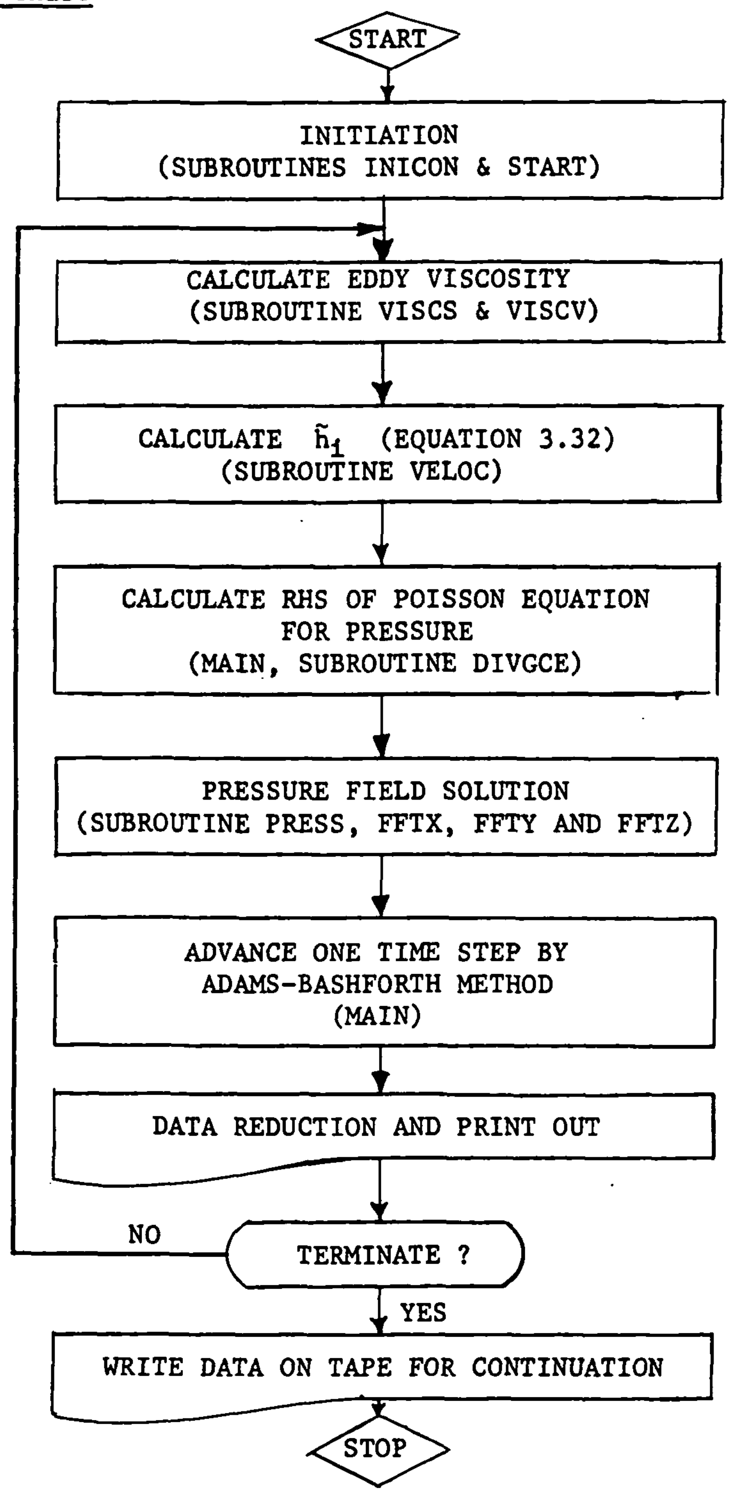
$$+ u_i \left(w_{(m+1)} - w_{(m-1)} \right) \Big\} + 0(\Delta^2) \qquad \text{(A.18)}$$

The subscript is shown whenever it is different from (k,ℓ,m), i.e. $u_{(k+1)} = u_{(k+1,\ell,m)}$

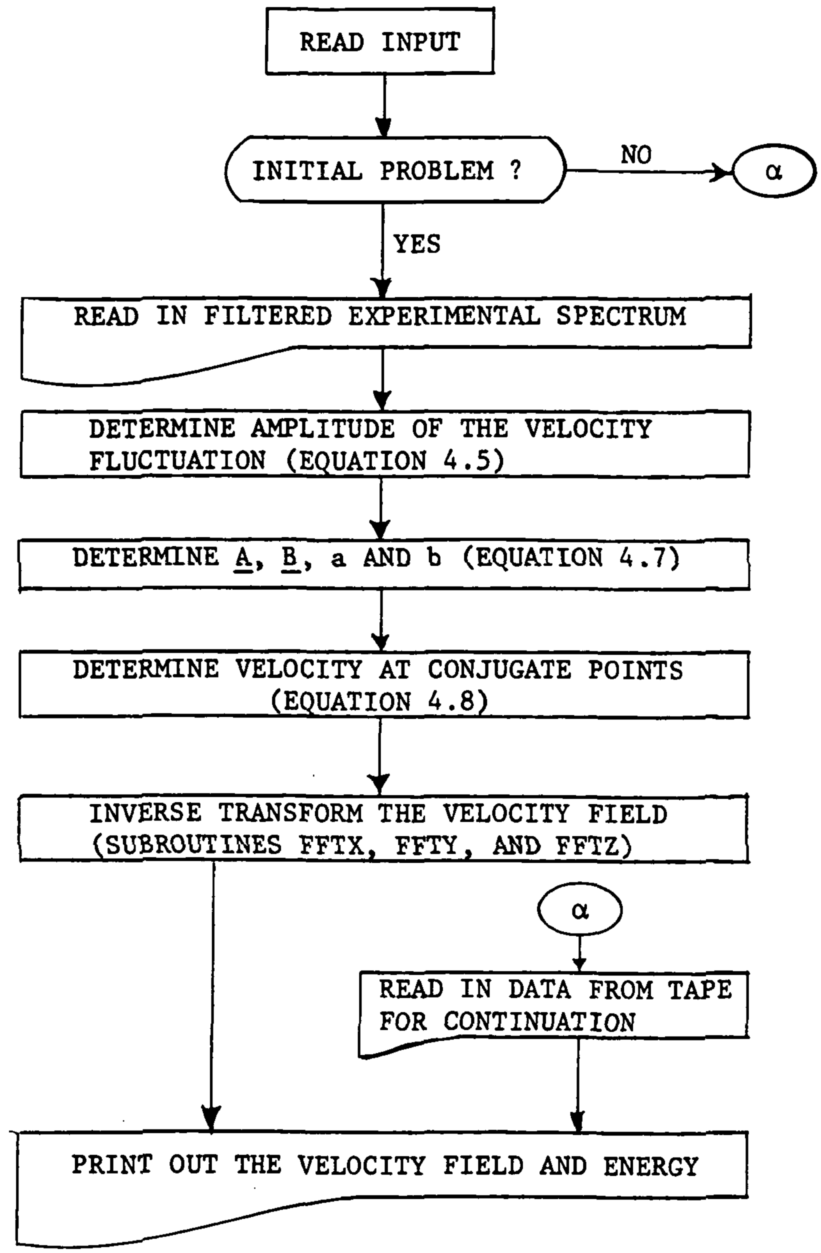
APPENDIX II

FLOW CHARTS AND PROGRAM

Overall Flow Chart



Flow Chart for INICON



```
*DECK MAIN
      PROGRAM MAIN   (INPUT,OUTPUT,TAPE9,TAPE10)
C**********************************************************************
C THIS CONTROLS THE OVERALL SEQUENCE OF COMPUTATION.                  *
C THE COMPUTATION IS PERFORMED MOSTLY IN THE SMALL CORE BY TRANSFERING *
C DATA FROM THE LARGE CORE MEMORY. AT THE END OF EACH TIME STEP THE    *
C STATISTICAL QUANTITIES ARE PRINTED OUT.  AT THE END OF THE COMPUTA-  *
C TION VELOCITIES AND THE RIGHT HAND SIDE OF MOMENTUM EQUATIONS ARE    *
C STORED FOR CONTINUATION.                                             *
C**********************************************************************
      INTEGER TIME,Z0,ZM1,ZM2,ZP1,ZP2,Z, ZADD1,ZLESS1,ZSAVE,PLANE
     1 ,TSTART,TEND
      REAL K, NLEN,NAVG
      LARGE FM(16,16,16),G(16,16,16),E(16,16,16),EI(16,16,16),
     1    D1(16,16,16),D2(16,16,16),RDUM(16,3),RD(192)
      LARGE UM(16,16,16),VM(16,16,16),WM(16,16,16) ,GU(16,16,16)
     *      ,GV(16,16,16),GW(16,16,16)
      LARGE RU(16,16,16),RV(16,16,16),RW(16,16,16),P1(16,16,16)
     .    ,GU1(16,16,16),GV1(16,16,16),GW1(16,16,16)
      DIMENSION U(16,16,5),V(16,16,5),W(16,16,5),P(16,16,5)
     1    ,DUMMY(16,16,6),SIG(16,16,3), K(16,16,3), FR(16,16),FI(16,16)
     2    ,TRR(8,3),TRI(8,3),GR(16),GI(16),NWAVE(16),NFFT(3)
      COMMON/DATA1/U,V,W
      COMMON/DATA2/P
      COMMON/DATA3/SIG,K
      COMMON/DATA4/DUMMY
      COMMON/DATA5/GR,GI,TRR,TRI
      COMMON/DATA7/FR,FI
      COMMON/DATA8/NWAVE,NFFT
      COMMON/DATA9/IMAX,JMAX,LMAX,NHALF,NAVG,NLEN,NSPEC
C     COEF1=1./(144.*DELTA**2)
C     COEF2=C*DELTA*0.5
C     COEF4=1./24.
C     COEF5=1./(4.*DELTA**3)
C     COEF6=12./DELTA
C     COEF7=1./(12.*DELTA)
C     COEF71=1./(72.*DELTA**2)
C     COEF8=2./(3.*DT)
C     COEF9=PI*DELTA*5./3.
C     ALPHA=DT/(48.*DELTA)
C     BETA=1./(48.*DELTA)
C     COF=1.0 FOR THE FIRST TIME STEP  =1.5 AFTERWARDS
C     UTMORG=U0*T/M-3.5
C     UTM=MEAN CONVECTIVE DOWNSTREAM VELOCITY
C
C-----INITIATION
C
      IWRITE=1
      UTMORG=38.5
      UOM=1000./5.08
      COF1=0.5
      CALL INICON (COEF1,COEF2,COEF3,COEF4,COEF5,COEF6,COEF7,COEF8,COEF9
     1,COEF11,COEF14,ALPHA,COF,DELTA,DT,CONST,UFR,NWRITE,GAMMA,
     2 TSTART,TEND,NMODEL)
      NFILT=1
      WOS=1./(UFR**2)
      GAMMAS=GAMMA**2
      GCOF=GAMMA*COEF7
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      G6=GAMMA/6.
      GD=GAMMA/DELTA
      GD2=0.5*GD
      COEF16=(DELTA*NAVG)**2/24.
      COEF17=(COEF11/DELTA)**2
      COEF71=COEF7/(6.*DELTA)
      BETA=COEF7/4.
      TDIV=1./4096.
C
C-----COMPUTATION STARTS
C
      DO 300 TIME=TSTART,TEND
   10 CONTINUE
C-----COMPUTE EDDY VISCOSITY "E".
      Z=1
      ZM1=1
      Z0=2
      ZP1=3
C-----FIRST TRANSFER DATA FROM LCM TO SCM FOR THE PLANES L=LMAX AND 1
      SMALLIN (U(1,1,1),UM(1,1,LMAX),256)
      SMALLIN (V(1,1,1),VM(1,1,LMAX),256)
      SMALLIN (W(1,1,1),WM(1,1,LMAX),256)
      SMALLIN (U(1,1,2),UM(1,1,1),256)
      SMALLIN (V(1,1,2),VM(1,1,1),256)
      SMALLIN (W(1,1,2),WM(1,1,1),256)
      DO 700 L=1,LMAX
      LP1=L+1
      IF (L .EQ. LMAX) LP1=1
C     TRANSFER VELOCITIES FOR THE PLANE=L+1
      SMALLIN (U(1,1,ZP1),UM(1,1,LP1),256)
      SMALLIN (V(1,1,ZP1),VM(1,1,LP1),256)
      SMALLIN (W(1,1,ZP1),WM(1,1,LP1),256)
C     THEN WE CAN COMPUTE EDDY VISCOSITY AT THE PLANE=L
      GO TO (20,25) MMODEL
   20 CALL VISCS(Z0,ZM1,ZP1,Z,COEF2)
      GO TO 30
   25 CALL VISCV(Z0,ZM1,ZP1,Z,COEF2)
   30 CONTINUE
C     THE EDDY VISCOSITY ON THE PLANE=L IS TRANSFERED TO E IN LCM
      SMALLOUT (K(1,1,Z),E(1,1,L),256)
      ZSAVE=ZM1
      ZM1=Z0
      Z0=ZP1
      ZP1=ZSAVE
  700 CONTINUE
C-----COMPUTE H(I), THEN STORE THEM IN GU, GV & GW
      ZM2=1
      ZM1=2
      Z0=3
      ZP1=4
      ZP2=5
      SMALLIN (U(1,1,1),UM(1,1,LMAX-1),512)
      SMALLIN (V(1,1,1),VM(1,1,LMAX-1),512)
      SMALLIN (W(1,1,1),WM(1,1,LMAX-1),512)
      SMALLIN (U(1,1,3),UM(1,1,1),768)
      SMALLIN (V(1,1,3),VM(1,1,1),768)
      SMALLIN (W(1,1,3),WM(1,1,1),768)
      SMALLIN (K(1,1,1),E(1,1,LMAX),256)
      SMALLIN (K(1,1,2),E(1,1,1),512)
      ZLESS1=1
      Z=2
      ZADD1=3
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 760 PLANE=1,LMAX
      CALL VELOC (Z0,ZM1,ZM2,ZP1,ZP2,Z,ZLESS1,ZADD1,BETA,COEF6,COEF7,
     1    DT,CDF,GAMMA,GAMMAS,GCOF,GD2,GD,DELTA,G6)
      SMALLOUT(DUMMY(1,1,1),GU(1,1,PLANE),256)
      SMALLOUT(DUMMY(1,1,2),GV(1,1,PLANE),256)
      SMALLOUT(DUMMY(1,1,3),GW(1,1,PLANE),256)
      SMALLOUT (SIG(1,1,1),D1(1,1,PLANE),256)
      SMALLOUT (SIG(1,1,2),D2(1,1,PLANE),256)
C     P1:PRESSURE CONTRIBUTION BY MEAN MOTION
      SMALLOUT (DUMMY(1,1,4),GU1(1,1,PLANE),256)
      SMALLOUT (DUMMY(1,1,5),GV1(1,1,PLANE),256)
      SMALLOUT (DUMMY(1,1,6),GW1(1,1,PLANE),256)
      CALL DIVGCE (Z0,ZM1,ZM2,ZP1,ZP2,COEF7)
      SMALLOUT (DUMMY(1,1,6),G(1,1,PLANE),256)
      ZSAVE=ZLESS1
      ZLESS1=Z
      Z=ZADD1
      ZADD1=ZSAVE
      L=PLANE+2
      IF (L .GT. LMAX) L=L-LMAX
      SMALLIN (K(1,1,ZADD1),E(1,1,L),256)
      ZSAVE=ZM2
      ZM2=ZM1
      ZM1=Z0
      Z0=ZP1
      ZP1=ZP2
      ZP2=ZSAVE
      L=PLANE+3
      IF (L .GT. LMAX) L=L-LMAX
      SMALLIN (U(1,1,ZP2),UM(1,1,L),256)
      SMALLIN (V(1,1,ZP2),VM(1,1,L),256)
      SMALLIN (W(1,1,ZP2),WM(1,1,L),256)
  760 CONTINUE
C-----NOW GU,GV,GW ARE COMPUTED AND DIV U IS STORED IN G.
C     COMPUTE DIV(GU), THEN GET G
      ZM2=1
      ZM1=2
      Z0=3
      ZP1=4
      ZP2=5
      SMALLIN (U(1,1,1),GU(1,1,LMAX-1),512)
      SMALLIN (V(1,1,1),GV(1,1,LMAX-1),512)
      SMALLIN (W(1,1,1),GW(1,1,LMAX-1),512)
      SMALLIN (U(1,1,3),GU(1,1,1),768)
      SMALLIN (V(1,1,3),GV(1,1,1),768)
      SMALLIN (W(1,1,3),GW(1,1,1),768)
      DO 770 PLANE=1,LMAX
      CALL DIVGCE(Z0,ZM1,ZM2,ZP1,ZP2,COEF7)
      SMALLOUT (DUMMY(1,1,6),PM(1,1,PLANE),256)
      ZSAVE=ZM2
      ZM2=ZM1
      ZM1=Z0
      Z0=ZP1
      ZP1=ZP2
      ZP2=ZSAVE
      L=PLANE+3
      IF (L .GT. LMAX) L=L-LMAX
      SMALLIN (U(1,1,ZP2),GU(1,1,L),256)
      SMALLIN (V(1,1,ZP2),GV(1,1,L),256)
      SMALLIN (W(1,1,ZP2),GW(1,1,L),256)
  770 CONTINUE
C-----D(GU) IS STORED IN PM
```

```
      DO 780 L=1,LMAX
      DO 780 J=1,JMAX
      DO 780 I=1,IMAX
      P1(I,J,L)=COEF8*G(I,J,L)+PM(I,J,L)
  780 CONTINUE
C-----NOW G IS TEMPORARILY STORED IN P1
C-----COMPUTE G1
      ZM2=1
      ZM1=2
      Z0=3
      ZP1=4
      ZP2=5
      SMALLIN (U(1,1,1),GU1(1,1,LMAX-1),512)
      SMALLIN (V(1,1,1),GV1(1,1,LMAX-1),512)
      SMALLIN (W(1,1,1),GW1(1,1,LMAX-1),512)
      SMALLIN (U(1,1,3),GU1(1,1,1),768)
      SMALLIN (V(1,1,3),GV1(1,1,1),768)
      SMALLIN (W(1,1,3),GW1(1,1,1),768)
      DO 1770 PLANE=1,LMAX
      CALL DIVGCE(Z0,ZM1,ZM2,ZP1,ZP2,COEF7)
      SMALLOUT (DUMMY(1,1,6),G(1,1,PLANE),256)
      ZSAVE=ZM2
      ZM2=ZM1
      ZM1=Z0
      Z0=ZP1
      ZP1=ZP2
      ZP2=ZSAVE
      L=PLANE+3
      IF (L .GT. LMAX) L=L-LMAX
      SMALLIN (U(1,1,ZP2),GU1(1,1,L),256)
      SMALLIN (V(1,1,ZP2),GV1(1,1,L),256)
      SMALLIN (W(1,1,ZP2),GW1(1,1,L),256)
 1770 CONTINUE
C-----COMPUTE P1
      CALL PRESS (COEF3,COEF11,COEF71)
C-----NOW STORE P1 TO P1 FROM PM AND SHIFT G FROM P1 TO G1
      DO 1880 L=1,LMAX
      SMALLIN (DUMMY(1,1,1),P1(1,1,L),256)
      SMALLIN (DUMMY(1,1,2),PM(1,1,L),256)
      SMALLOUT (DUMMY(1,1,1),G(1,1,L),256)
      SMALLOUT (DUMMY(1,1,2),P1(1,1,L),256)
 1880 CONTINUE
C
      PRINT 903
      DO 400 L=1,16
      PRINT 904, L
      PRINT 902, (E(I,10,L),I=1,16)
  400 CONTINUE
      PRINT 917
      DO 401 J=1,16
      PRINT 918,J
      PRINT 902, (E(I,J,10),I=1,16)
  401 CONTINUE
C
      CALL PRESS(COEF3,COEF11,COEF71)
C-----STORE U,V,W,P AT FIVE STEP INTERVAL
      IF (NSTORE .EQ. 5) GO TO 110
      NSTORE=NSTORE+1
      GO TO M20
  110 NTIME =TIME-1
      WRITE (9) NTIME,UM,VM,WM,PM,P1
      END FILE 9
```

```
      NSTORE=1
  120 CONTINUE
C-----RHS=GU(I)-DP/DX(I)
C-----STORE RHS(N) TO RDUM, THEN ADVANCE 1 STEP
      DO 800 L=1,LMAX
      LM2=L-2
      LM1=L-1
      LP1=L+1
      LP2=L+2
      GO TO (781,782,785,785,785,785,785,785,785,785,785,785,785,785,
     -    783,784) L
  781 LM2=LMAX-1
      LM1=LMAX
      GO TO 785
  782 LM2=LMAX
      GO TO 785
  783 LP2=1
      GO TO 785
  784 LP1=1
      LP2=2
  785 CONTINUE
      DO 800 J=1,JMAX
      JM2=J-2
      JM1=J-1
      JP1=J+1
      JP2=J+2
      GO TO (786,787,790,790,790,790,790,790,790,790,790,790,790,790,
     - 788,789) J
  786 JM2=JMAX-1
      JM1=JMAX
      GO TO 790
  787 JM2=JMAX
      GO TO 790
  788 JP2=1
      GO TO 790
  789 JP1=1
      JP2=2
  790 CONTINUE
      DO 800 I=1,IMAX
      IM2=I-2
      IM1=I-1
      IP1=I+1
      IP2=I+2
      GO TO (791,792,795,795,795,795,795,795,795,795,795,795,795,795,
     -    793,794) I
  791 IM2=IMAX-1
      IM1=IMAX
      GO TO 795
  792 IM2=IMAX
      GO TO 795
  793 IP2=1
      GO TO 795
  794 IP1=1
      IP2=2
  795 CONTINUE
      RDUM(I,1)=GU(I,J,L)-COEF7*(PM(IM2,J,L)-8.*(PM(IM1,J,L)-PM(IP1,J,L)
     - )-PM(IP2,J,L))
      RDUM(I,2)=GV(I,J,L)-COEF7*(PM(I,JM2,L)-8.*(PM(I,JM1,L)-PM(I,JP1,L)
     - )-PM(I,JP2,L))
      RDUM(I,3)=GW(I,J,L)-COEF7*(PM(I,J,LM2)-8.*(PM(I,J,LM1)-PM(I,J,LP1)
     - )-PM(I,J,LP2))
      UM(I,J,L)=UM(I,J,L)+DT*(COF*RDUM(I,1)-0.5*RU(I,J,L))
```

```
      VM(I,J,L)=VM(I,J,L)+DT*(COF*RDUM(I,2)-0.5*RV(I,J,L))
      WM(I,J,L)=WM(I,J,L)+DT*(COF*RDUM(I,3)-0.5*RW(I,J,L))
      RU(I,J,L)=RDUM(I,1)
      RV(I,J,L)=RDUM(I,2)
      RW(I,J,L)=RDUM(I,3)
  800 CONTINUE
C-----TURBULENT ENERGY
      TKU=0.
      TKV=0.
      TKW=0.
      DO 230 L=1,LMAX
      RDUM(L,1)=0.
      RDUM(L,2)=0.
      RDUM(L,3)=0.
      DO 230 J=1,JMAX
      DO 230 I=1,IMAX
      TKU=TKU+UM(I,J,L)**2
      TKV=TKV+VM(I,J,L)**2
      TKW=TKW+WM(I,J,L)**2
      RDUM(L,1)=RDUM(L,1)+UM(I,J,L)
      RDUM(L,2)=RDUM(L,2)+VM(I,J,L)
      RDUM(L,3)=RDUM(L,3)+WM(I,J,L)
  230 CONTINUE
      TKU=TKU*TDIV
      TKV=TKV*TDIV
      TKW=TKW*TDIV
      USUM=0.
      VSUM=0.
      WSUM=0.
      DO 240 L=1,LMAX
      USUM=USUM+RDUM(L,1)
      VSUM=VSUM+RDUM(L,2)
      WSUM=WSUM+RDUM(L,3)
  240 CONTINUE
C
      COF=1.5
      PRINT 910
      PRINT 908
      TKSUM=TKU+TKV+TKW
      PRINT 909,TIME ,TKU,TKV,TKW,TKSUM
      RTIME=TIME*DT
      DIST=UFR*RTIME
      SRATIO=EXP(0.0185416*DIST)
      PRINT 915,RTIME,DIST,SRATIO
      TKDIV=1./TKSUM
      TKU1=TKU*TKDIV
      TKV1=TKV*TKDIV
      TKW1=TKW*TKDIV
      TK1=(TKV-TKU)/(TKV+TKU)
      PRINT 906,TKU1,TKV1,TKW1,TK1
      UTM=UOM*TIME*DT +UTMORG
      UOU=1.0E+06/TKU
      UOV=1.0E+06/TKV
      UOW=1.0E+06/TKW
      UOA=3.0E+06/TKSUM
      PRINT 911,UTM,UOU,UOV,UOW,UOA
      UW=TKU*WOS
      VW=TKV*WOS
      WW=TKW*WOS
      QW=TKSUM*WOS
      PRINT 916,UW,VW,WW,QW
C-----DISSIPATION TERMS; LEONARD, SGS, TOTAL.
```

```
      DLEN=0.
      DSGS=0.
      DO 241 L=1,LMAX
      DO 241 J=1,JMAX
      DO 241 I=1,IMAX
      DLEN=DLEN+D1(I,J,L)
      DSGS=DSGS+D2(I,J,L)
  241 CONTINUE
      DTOT=DLEN+DSGS
      PRINT 913, DLEN,DSGS,DTOT
C-----SKEWNESS CHECK
      SK3=0.
      SK2=0.
      DO 260 I=1,IMAX
      IM2=I-2
      IM1=I-1
      IP1=I+1
      IP2=I+2
      IF (I .GT. 2) GO TO 253
      GO TO (251,252) I
  251 IM1=LMAX
      IM2=LMAX-1
      GO TO 256
  252 IM2=LMAX
      GO TO 256
  253 I1=IMAX-1
      IF (I .LT. I1) GO TO 256
      I2=IMAX-I+1
      GO TO (255,254) I2
  254 IP2=1
      GO TO 256
  255 IP1=1
      IP2=2
  256 CONTINUE
      DO 260 J=1,JMAX
      DO 260 L=1,LMAX
      DUDX=UM(IM2,J,L)-8.*(UM(IM1,J,L)-UM(IP1,J,L))-UM(IP2,J,L)
      SK3=SK3+DUDX**3
      SK2=SK2+DUDX**2
  260 CONTINUE
      SK3=SK3*TDIV
      SK2=(SK2*TDIV)**1.5
      SK=SK3/SK2
      PRINT 912,SK
      GO TO (270,280) NFILT
  270 NFILT=2
      GO TO 281
  280 NFILT=1
  281 CONTINUE
      PRINT 908
      PRINT 905
      DO 410 L=1,NHALF
      PRINT 904,L
      PRINT 902, (UM(I,10,L), I=1,NHALF)
      PRINT 902, (VM(I,10,L), I=1,NHALF)
      PRINT 902, (WM(I,10,L), I=1,NHALF)
      PRINT 902, (RDU4(L,I),I=1,3)
  410 CONTINUE
      PRINT 902, USUM,VSUM,WSUM
  300 CONTINUE
      WRITE (9)  TIME,UM,VM,WM,PM,P1,RU,RV,RW
      END FILE 9
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
  320 CONTINUE
  902 FORMAT (X, 6(E14.6,X))
  903 FORMAT (///,X,*EDDY VISCOSITY AT J=10*)
  904 FORMAT (X,*PLANE = *,I3)
  905 FORMAT (///,X,*VELOCITY AT J=10 ; U= *)
  906 FORMAT (X, 1H*,5X,9HU**2/SUM= , E14.7, 5X, 9HV**2/SUM= , E14.7,
     1 5X, 9HW**2/SUM= , E14.7, 5X, *K1=(V2-U2)/(V2+U2)=*, E14.7,9X,1H*)
  908 FORMAT (X, 130H*****************************************************
     -*****************************************************************
     -**************     )
  909 FORMAT (X,1H*, 5X, *TIME STEP=*, I4,5X,*U2 =*,E14.7, 5X, *V2 =*,
     -  E14.7, 5X, *W2 =*, E14.7,  *  SUM=*, E14.7, 19X, 1H*)
  910 FORMAT (1H1)
  911 FORMAT (X, 1H*,5X,*U01/W=*,E12.5,3X,*U02/U2=*,E12.5,3X,*U02/V2=*,
     -  E12.5,3X,*U02/W2=*,E12.5,3X, *U02/UAVG2=*, E12.5,14X,1H*)
  912 FORMAT (X,1H*,5X,*SKEWNESS=-(DU/DX) =*,E12.5,91X,1H*)
  913 FORMAT (X, 1H*,5X,*DISSIPATION; LEONARD=*,E12.5,3X, *SGS=*,E12.5,
     1     3X, *TOTAL=*, E12.5, 49X, 1H*)
  915 FORMAT (X, 1H*, 5X, *REAL TIME=*, F8.4, 5X, *DISTANCE=*, F9.4,
     1     * IN*, 5X, *STRAIN RATIO=*, F7.3, 53X, 1H*)
  916 FORMAT (X, 1H*, 5X, 12HU**2/U0**2 = ,E14.7, 4X, 12HV**2/U0**2 = ,
     1    E14.7, 4X, 12HW**2/U0**2 = ,E14.7, 4X, 12HQ**2/U0**2 =,E14.7,
     2    6X, 1H*)
  917 FORMAT (///,X,*EDDY VISCOSITY AT PLANE=10*)
  918 FORMAT (X, *J=*, I3)
      STOP
      END



*DECK INICON
      SUBROUTINE INICON (COEF1,COEF2,COEF3,COEF4,COEF5,COEF6,COEF7,
     -COEF8,COEF9,COEF11,COEF14,ALPHA,COF,DELTA,DT,CONST,UTM,NWRITE,
     2  GAMMA,TSTART,TEND,NMODEL)
C*********************************************************************
C THIS SUBROUTINE INITIATES THE PROGRAM. FOR STARTING PROBLEM, THE INI-*
C TIAL FIELD IS GENERATED.  FOR CONTINUATION PROBLEM, THE DATA STORED  *
C ON TAPE AT THE END OF THE PREVIOUS RUN ARE READ IN.                  *
C*********************************************************************
      INTEGER PLANE,TSTART,TEND
      REAL NDIV,N12,NX,NSUR,K,NLEN,NAVG
      LARGE PH(16,16,16),G(16,16,16),F(16,16,16),EI(16,16,16),
     1     D1(16,16,16),D2(16,16,16),EN(60),EU1(60),ED1(60),ED2(60)
      LARGE UR(16,16,16),VR(16,16,16),WR(16,16,16) ,UI(16,16,16)
     -       ,VI(16,16,16),WI(16,16,16)
      LARGE RU(16,16,16),RV(16,16,16),RW(16,16,16),P1(16,16,16)
     .     ,GU1(16,16,16),GV1(16,16,16),GW1(16,16,16)
      DIMENSION HH(16,16,16),H(16,16,16),FR(16,16),FI(16,16),TRR(8,3),
     -  TRI(8,3),HDUM(16,2),GR(16),GI(16),NWAVE(16),NFFT(3)
      COMMON/DATA5/GR,GI,TRR,TRI
      COMMON/DATA7/FR,FI
      COMMON/DATA8/NWAVE,NFFT
      COMMON/DATA9/IMAX,JMAX,LMAX,NHALF,NAVG,NLEN,NSPEC
C-----NSTART=1  STARTING FROM TIME STEP=0
C-----NSTART=2 CONTINUED FROM PREVIOUS RUN
C-----IMAX=MAXIMUM MESH NUMBER IN X-DIRECTION
C-----JMAX=MAXIMUM MESH NUMBER IN Y-DIRECTION
C-----LMAX=MAXIMUM MESH NUMBER IN Z-DIRECTION
C-----TSTART=STARTING TIME STEP
C-----TEND=ENDING TIME STEP
```

```
C-----DELTA= MESH SIZE
C-----DT=TIME STEP
C-----C=MODEL CONSTANT
C-----NAVG=DELTA(AVERAGING)/DELTA(MESH)
C-----ANISO= R IN EQUATION (5.16B)
C-----GAMMA=STRAIN RATE
C-----NMODEL=1 FOR SMAGORINSKY MODEL
C-----NMODEL=2 FOR VORTICITY MODEL
      READ 703,NSTART ,IMAX,JMAX,LMAX,TSTART,TEND,NMODEL
      READ 4, DELTA,DT,C, NAVG,ANISO,UTM,GAMMA
      FLEN=NAVG**2
      NMESH=16
      NHALF=NMESH/2
      NHP1=NHALF+1
      NM1=NMESH-1
      RISO=3./(3.+ANISO)
      TEMP=ANISO/3.
      RANISO=SQRT(TEMP)
      CC=1.
      TFAC=IMAX*JMAX*LMAX
      FAC=SQRT(TFAC)
      COEF1=1./(144.*DELTA**2)
      COEF2=C*DELTA*0.5
      COEF4=1./24.
      COEF5=1./(4.*DELTA**3)
      COEF6=12./DELTA
      COEF7=1./(12.*DELTA)
      COEF72=COEF7*2.
      COEF8=2./(3.*DT)
      COEF9=0.31415926535898
      COEF10=3.1415926535898/NHALF
      COEF11=COEF10
      COEF12=3.1415926535898*2.
      ALPHA=DT/(48.*DELTA)
      CONST=COEF10/ DELTA
      CONSTS=CONST**2
      COEF14=COEF12*FAC*CONSTS
      COEF15=COEF12*FAC
      PI1=COEF10
      PI2=PI1*2.
      CALL START (COEF3,COEF11,DELTA)
      GO TO (1,1000,5000), NSTART
    1 COF=1.0
      MCONT=1
      DO 2 M=1,25
      Y9=RGEN(X9)
    2 CONTINUE
C-----ENERGY SPECTRUM DATA
C----- 0.1 INTERVAL UP TO 1.0   THEN .5 INTERVALL UP TO 6.0
C-----EN IS THE ENERGY SPECTRUM FOR THE ISOTROPIC PART.  EN1 IS FOR THE
C ANISOTROPIC PART.
      DO 3 M=1,24,8
      M7=M+7
      READ 4, (EN(MM),MM=M,M7)
    3 CONTINUE
      DO 503 M=1,24,8
      M7=M+7
      READ 4, (EN1(MM),MM=M,M7)
  503 CONTINUE
    4 FORMAT (8E10.4)
      DO 5 L=1,LMAX
      DO 5 J=1,JMAX
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 5 I=1,IMAX
      UR(I,J,L)=0.
      VR(I,J,L)=0.
      WR(I,J,L)=0.
      UI(I,J,L)=0.
      VI(I,J,L)=0.
      WI(I,J,L)=0.
      RU(I,J,L)=0.
      RV(I,J,L)=0.
      RW(I,J,L)=0.
    5 CONTINUE
      DO 40 L=1,NHALF
      LL=L
      N3=L-1
      N3S=N3**2
      DO 30 J=1,NM1
      JINDEX=J/8
      JJ=J+NHP1-JINDEX*JMAX
      N2=J-NHALF
      N2S=N2**2
      DO 20 I=1,NM1
      IINDEX=I/8
      II=I+NHP1-IINDEX*IMAX
      N1=I-NHALF
      N1S=N1**2
      NSQR=N1S+N2S+N3S
      IF (NSQR .LT. 0.1) GO TO 20
      WAVN=SQRT(NSQR)
      NDIV=1./WAVN
      N12=N1S+N2S
      IF (N12 .LT. 0.1) NCONT=2
      IF (ABS(N1) .EQ. NHALF .AND. ABS(N2) .EQ. NHALF) NCONT=2
C-----GET FOURIER AMPLITUDE OF THE INITIAL FIELD AD DESCRIBED IN SEC 4.4
    7 X=CONST*WAVN
      NREG=X+1.
      GO TO (310,315,315,315,315,315,315) NREG
  310 M=X/0.1
      YM=X-0.1*M
      M1=M+1
      ED=EN(M1)-EN(M)
      ENERGY=EN(M)+ED*YM*10.
      EDA=EN1(M1)-EN1(M)
      EANISO=EN1(M)+EDA*YM*10.
      GO TO 320
  315 M=(X-1.)*2.
      YM=X-1.-0.5*M
      M=M+10
      M1=M+1
      ED=EN(M1)-EN(M)
      ENERGY=EN(M)+ED*YM*2.
      EDA=EN1(M1)-EN1(M)
      EANISO=EN1(M)+EDA*YM*2.
  320 QS=ENERGY*RISO/(COEF1S*X**2)
      QN=SQRT(QS)
      QSA=EANISO*RISO/(COEF1S*X**2)
      QNA=SQRT(QSA)
C-----CHANGE WAVE NUMBER VECTOR TO SATISFY NUMERICSL DIV FREE
C-----R1,R2,AND R3 ARE THE MODIFIED WAVE NUMBER
      GO TO (330,340) NCONT
  330 CONTINUE
      ARG1=PI1*N1
      ARG2=PI2*N1
```

```
      R1=(8.*SIN(ARG1)-SIN(ARG2))*COEF72
      ARG1=PI1*N2
      ARG2=PI2*N2
      R2=(8.*SIN(ARG1)-SIN(ARG2))*COEF72
      ARG1=PI1*N3
      ARG2=PI2*N3
      R3=(8.*SIN(ARG1)-SIN(ARG2))*COEF72
      R1S=R1**2
      R2S=R2**2
      R3S=R3**2
      R12S=R1S+R2S
      RSQ=R12S+R3S
      GO TO (335,340) NCONT
  335 CONTINUE
      R12=SQRT(R12S)
      R12DIV=1./R12
      RDIV=1./SQRT(RSQ)
C ----GET A & B  VECTOR
C     FIRST CHOOSE RANDOM PHI
  340 CONTINUE
      YY=RGEN(XX)
      PHI=YY*COEF12
      CPHI=COS(PHI)
      SPHI=SIN(PHI)
      GO TO ( 8,11) NCONT
    8 CONTINUE
      A1=(-R2*CPHI+R1*R3*RDIV*SPHI)*R12DIV
      A2=(R1*CPHI+R2*R3*RDIV*SPHI)*R12DIV
      A3=-R12*RDIV*SPHI
C     CALL RANDOM PHI
      Y2=RGEN(X2)
      PHI=Y2*COEF12
      CPHI=COS(PHI)
      SPHI=SIN(PHI)
      B1=(-R2*CPHI+R1*R3*RDIV*SPHI)*R12DIV
      B2=(R1*CPHI+R2*R3*RDIV*SPHI)*R12DIV
      B3=-R12*RDIV*SPHI
      GO TO 12
   11 CONTINUE
      INDEX=(YY+0.25)*4
      PHI=0.7853982*(2*INDEX-1)
      A1=SIN(PHI)
      A2=COS(PHI)
      A3=0.
      Y1=RGEN(X1)
      INDEX=(Y1+0.25)*4
      PHI=0.7853982*(2*INDEX-1)
      B1=SIN(PHI)
      B2=COS(PHI)
      B3=0.
      NCONT=1
   12 CONTINUE
C     DETERMINE A AND B IN EQUATION (4.6)
C     RANDOM THETA
      Y3=RGEN(X3)
      THETA=Y3*COEF12
      CA=COS(THETA)
      CB=SIN(THETA)
      UR(II,JJ,LL)=GN*CA*A1
      VR(II,JJ,LL)=GN*CA*A2
      WR(II,JJ,LL)=GN*CA*A3
      UI(II,JJ,LL)=GN*CB*B1
```

```
      VI(II,JJ,LL)=ON*CB*B2
      WI(II,JJ,LL)=ON*CB*B3
      IF (N3 .NE. 0) GO TO 20
      WSIGN=ABS(A3)/A3
      VSIGN=ABS(B3)/B3
      WRAN=ONA*CA*RANISO
      WIAN=ONA*CB*RANISO
      WR(II,JJ,LL)=WR(II,JJ,LL)+WRAN*WSIGN
      VI(II,JJ,LL)=VI(II,JJ,LL)+WIAN*VSIGN
   20 CONTINUE
   30 CONTINUE
   40 CONTINUE
C     NOW THE UPPER HALF OF THE K-SPACE HAS BEEN DETERMINED
C     GET THE TRANSFORMED VELOCITY AT THE CONJUGATE POINTS
C-----CONJUGATE FORM
C     N3=1 TO 7, N1 & N2=-7 TO 7
C     N3=L-1      -N3=LM
C     N2=J-1      -N2=JM
      DO 41 L=2,8
      LM=L+18-2*L
      DO 41 J=1,16
      M=(J+15)/17
      JM=J+(18-2*J)*M
      DO 41 I=1,16
      M=(I+15)/17
      IM=I+(18-2*I)*M
      UR(IM,JM,LM)= UR(I,J,L)
      VR(IM,JM,LM)= VR(I,J,L)
      WR(IM,JM,LM)= WR(I,J,L)
      UI(IM,JM,LM)=-UI(I,J,L)
      VI(IM,JM,LM)=-VI(I,J,L)
      WI(IM,JM,LM)=-WI(I,J,L)
   41 CONTINUE
C     N3=0, N1=1 TO 7, N2=-7 TO 7
      DO 42 I=2,8
      IM=I+18-2*I
      DO 42 J=1,16
      M=(J+15)/17
      JM=J+(18-2*J)*M
      IF (J .EQ. 9) GO TO 42
      UR(IM,JM,1 )= UR(I,J,1)
      VR(IM,JM,1 )= VR(I,J,1)
      WR(IM,JM,1 )= WR(I,J,1)
      UI(IM,JM,1 )=-UI(I,J,1)
      VI(IM,JM,1 )=-VI(I,J,1)
      WI(IM,JM,1 )=-WI(I,J,1)
   42 CONTINUE
C     N1=N3=0
      DO 43 J=2,8
      JM=J+18-2*J
      UR(1,JM,1)= UR(1,J,1)
      VR(1,JM,1)= VR(1,J,1)
      WR(1,JM,1)= WR(1,J,1)
      UI(1,JM,1)=-UI(1,J,1)
      VI(1,JM,1)=-VI(1,J,1)
      WI(1,JM,1)=-WI(1,J,1)
   43 CONTINUE
C     INVERSE TRANSFORM
C     X AND Y TRANSFORM
      SIGN=-1.
      DO 50 L=1,LMAX
      SMALLIN (FR(1,1),UR(1,1,L),256)
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
      SMALLIN (FI(1,1),UI(1,1,L),256)
      CALL FFTY(SIGN)
      CALL FFTY(SIGN,CC)
      SMALLOUT(FR(1,1),UR(1,1,L),256)
      SMALLOUT(FI(1,1),UI(1,1,L),256)
      SMALLIN (FR(1,1),VR(1,1,L),256)
      SMALLIN (FI(1,1),VI(1,1,L),256)
      CALL FFTX(SIGN)
      CALL FFTY(SIGN,CC)
      SMALLOUT(FR(1,1),VR(1,1,L),256)
      SMALLOUT(FI(1,1),VI(1,1,L),256)
      SMALLIN (FR(1,1),WR(1,1,L),256)
      SMALLIN (FI(1,1),WI(1,1,L),256)
      CALL FFTX(SIGN)
      CALL FFTY(SIGN,CC)
      SMALLOUT(FR(1,1),WR(1,1,L),256)
      SMALLOUT(FI(1,1),WI(1,1,L),256)
   50 CONTINUE
C     Z TRANSFORM
      SMALLIN (HM(1,1,1),UR(1,1,1),4096)
      SMALLIN (H(1,1,1),UI(1,1,1),4096)
      CALL FFTZ(SIGN,HM,H)
      SMALLOUT (HM(1,1,1),UR(1,1,1),4096)
      SMALLOUT (H(1,1,1),UI(1,1,1),4096)
      SMALLIN (HM(1,1,1),VR(1,1,1),4096)
      SMALLIN (H(1,1,1),VI(1,1,1),4096)
      CALL FFTZ(SIGN,HM,H)
      SMALLOUT (HM(1,1,1),VR(1,1,1),4096)
      SMALLOUT (H(1,1,1),VI(1,1,1),4096)
      SMALLIN (HM(1,1,1),WR(1,1,1),4096)
      SMALLIN (H(1,1,1),WI(1,1,1),4096)
      CALL FFTZ(SIGN,HM,H)
      SMALLOUT (HM(1,1,1),WR(1,1,1),4096)
      SMALLOUT (H(1,1,1),WI(1,1,1),4096)
C-----THE INITIAL FIELD HAS BEEN GENERATED.  THE FOLLOWING IS TO PRINT
C     OUT  INFORMATION ON THE GENERATED FIELD
C     VELOCITIES ARE STORED IN UR, VR AND WR
C-----TURBULENT ENERGY CHECK
      TKU=0.
      TKV=0.
      TKW=0.
      TKI=0.
      DO 95 L=1, LMAX
      DO 95 J=1, JMAX
      DO 95 I=1, IMAX
      TKU=TKU+UR(I,J,L)**2
      TKV=TKV+VR(I,J,L)**2
      TKW=TKW+WR(I,J,L)**2
   95 CONTINUE
      TDIV=1./4096.
      TKU=TKU*TDIV
      TKV=TKV*TDIV
      TKW=TKW*TDIV
      TKSUM=TKU+TKV+TKW
      TKUR=TKU/TKSUM
      TKVR=TKV/TKSUM
      TKWR=TKW/TKSUM
      PRINT 706
      PRINT 700,DT,DELTA,C,NAVG
      PRINT 702, TKU,TKV,TKW, TKSUM
      PRINT 702,TKUR,TKVR,TKWR
      PRINT 706
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
      NTIME=0
  115 CONTINUE
      PRINT 601
      UTOT=0.
      VTOT=0.
      WTOT=0.
      DO 120 L=1,LMAX
      PRINT 710,L
      USUM=0.
      VSUM=0.
      WSUM=0.
      DO 116 J=1,JMAX
      DO 116 I=1,IMAX
      USUM=USUM+UR(I,J,L)
      VSUM=VSUM+VR(I,J,L)
      WSUM=WSUM+WR(I,J,L)
  116 CONTINUE
      PRINT 702, (UR(I,10,L),I=1,8)
      PRINT 702, (VR(I,10,L),I=1,8)
      PRINT 702, (WR(I,10,L),I=1,8)
      PRINT 702, USUM,VSUM,WSUM
      UTOT=UTOT+USUM
      VTOT=VTOT+VSUM
      WTOT=WTOT+WSUM
  120 CONTINUE
      PRINT 702,UTOT,VTOT,WTOT
      GO TO 130
 1000 COF=1.5
      PRINT 707
      PRINT 706
      PRINT 711,DT,DELTA,C
      READ (10) NTIME,UR,VR,WR,PM,PU,PV,PW
      NTIME=NTIME-1
      PRINT 705, NTIME
      PRINT 706
  601 FORMAT (X,*UM,VM,WM*)
      PRINT 601
      DO 125 L=1,LMAX
      PRINT 710,L
      PRINT 702, (UR(I,10,L),I=1,IMAX)
      PRINT 702, (VR(I,10,L),I=1,IMAX)
      PRINT 702, (WR(I,10,L),I=1,IMAX)
  125 CONTINUE
  130 CONTINUE
  700 FORMAT (X  ,*INITIAL CONDITION,  DT=*,E10.4,*  DELTA=*,E10.4,
     1 *   C=*,F7.4,3X,*AVERAGING GRID=*,F4.1, *  DELTA*)
  701 FORMAT (X,*L=*,I3)
  702 FORMAT (X,8(E14.7,X))
  703 FORMAT (10I3)
  704 FORMAT (4E20.14)
  705 FORMAT (X,  *CONTINUED AT TIME STEP=*,I4,/,/)
  706 FORMAT (X, 130H***********************************************
     -*************************************************************
     -*************      )
  707 FORMAT (1H1)
  710 FORMAT (X,*PLANE=*,I3)
  711 FORMAT (X,* INITIAL CONDITION*,/,X,*DT=*, E10.4,*  DELTA=*, E10.4
     - ,*  G=*,F7.4, *  UC=*, E10.4,/)
      GO TO 6000
 5000 CONTINUE
      COF=1.0
 6000 CONTINUE
```

```
      RETURN
      END




*DECK PRESS
      SUBROUTINE PRESS(COEF3,COEF11,COEF71)
C***********************************************************************
C     SOLVE POISSON EQUATION BY FOURIER TRANSFORM METHOD.              *
C***********************************************************************
      LARGE PM(16,16,16),G(16,16,16),E(16,16,16),EI(16,16,16),
     1    D1(16,16,16),D2(16,16,16),FN(60),EM1(60),ED1(60),PDUM(16,3),
     2    PD(12)
      LARGE UM(16,16,16),VM(16,16,16),WM(16,16,16) ,UI(16,16,16)
     *      ,VI(16,16,16),WI(16,16,16)
      LARGE RU(16,16,16),RV(16,16,16),RW(16,16,16),P1(16,16,16)
     *   ,GU1(16,16,16),GV1(16,16,16),GW1(16,16,16)
      DIMENSION HM(16,16,16),H(16,16,16)
      DIMENSION FR(16,16),FI(16,16),TRR(8,3),TRI(8,3),GR(16),GI(16)
     *   ,NWAVE(16),NFFT(3)
      COMMON/DATA5/GR,GI,TRR,TRI
      COMMON/DATA7/FR,FI
      COMMON/DATA8/NWAVE,NFFT
      COMMON/DATA9/IMAX,JMAX,LMAX,NHALF,NAVG,NLEN,NSPEC
C     FORWARD TRANSFORM
C     FORWARD TRANSFORM IN EACH PLANE. AFTER TRANSFORM STORE FR & FI TO G & PM.
      SIGN=+1.
      DO 20 L=1,LMAX
       SMALLIN (FR(1,1),G(1,1,L),256)
      CALL FFTY(SIGN)
      CALL FFTX(SIGN,COEF3)
      SMALLOUT(FR(1,1),PM(1,1,L),256)
      SMALLOUT (FI(1,1),G(1,1,L),256)
   20 CONTINUE
      SMALLIN (HM(1,1,1),PM(1,1,1),4096)
      SMALLIN (H(1,1,1),G(1,1,1),4096)
      CALL FFTZ (SIGN,HM,H)
      SMALLOUT(HM(1,1,1),PM(1,1,1),4096)
      SMALLOUT(H(1,1,1),G(1,1,1),4096)
C-----GET TRANSFORMED PR AND PI STORED IN G AND PM
      NP1=NHALF+1
      DO 210 L=1,LMAX
      MM=L/9
      M=MM*16+1
      ARG1=COEF11*(L-M)
      ARG2=ARG1*2.
      ARG3=ARG1*3.
      ARG4=ARG1*4.
      WAVEL=COS(ARG4)+16.*(COS(ARG1)-COS(ARG3))+64.*COS(ARG2)-65.
      DO 209 J=1,JMAX
      MM=J/9
      M=MM*16+1
      ARG1=COEF11*(J-M)
      ARG2=ARG1*2.
      ARG3=ARG1*3.
      ARG4=ARG1*4.
      WAVEJ=COS(ARG4)+16.*(COS(ARG1)-COS(ARG3))+64.*COS(ARG2)-65.
      DO 208 I=1,IMAX
      MM=I/9
      M=MM*16+1
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
      ARG1=COEF11*(I-M)
      ARG2=ARG1*2.
      ARG3=ARG1*3.
      ARG4=ARG1*4.
      WAVEI=COS(ARG4)+16.*(COS(ARG1)-COS(ARG3))+64.*COS(ARG2)-65.
      WV=(WAVEI+WAVEJ+WAVEL)*COEF71
      ITEST=I+J+L
      IF (ABS(WV) .LT. 0.0001) GO TO 205
      WAVE=1./WV
      G(I,J,L)=G(I,J,L)*WAVE
      PM(I,J,L)=PM(I,J,L)*WAVE
      GO TO 208
  205 G(I,J,L)=0.
      PM(I,J,L)=0.
  208 CONTINUE
  209 CONTINUE
  210 CONTINUE
C-----EXTRAPOLATE FOR LAST CORNER POINT
      I=NP1
      J=NP1
      L=NP1
      NM1=NHALF
      NM2=NM1-1
      G(I,J,L)=2.*(G(NM1,J,L)+G(I,NM1,L)+G(I,J,NM1))
     1    -(G(NM2,J,L)+G(I,NM2,L)+G(I,J,NM2))
      G(I,J,L)=G(I,J,L)/3
      PM(I,J,L)=2.*(PM(NM1,J,L)+PM(I,NM1,L)+PM(I,J,NM1))
     1    -(PM(NM2,J,L)+PM(I,NM2,L)+PM(I,J,NM2))
      PM(I,J,L)=PM(I,J,L)/3
C     INVERSE TRANSFORM
      SIGN=-1.
      DO 300 L=1,LMAX
      SMALLIN (FR(1,1),PM(1,1,L),256)
      SMALLIN (FI(1,1),G(1,1,L),256)
      CALL FFTY(SIGN,COEF3)
      CALL FFTX(SIGN)
       SMALLOUT(FR(1,1),PM(1,1,L),256)
      SMALLOUT (FI(1,1),G(1,1,L),256)
  300 CONTINUE
      SMALLIN (HM(1,1,1),PM(1,1,1),4096)
      SMALLIN (H(1,1,1),G(1,1,1),4096)
      CALL FFTZ (SIGN,HM,H)
      SMALLOUT(HM(1,1,1),PM(1,1,1),4096)
      SMALLOUT(H(1,1,1),G(1,1,1),4096)
      PRINT 904
      DO 910 L=1,8
          PRINT 903,L
      PRINT 901, (PM(I,10,L),I=1,8)
  910     CONTINUE
  901     FORMAT (X,8(E14.7,X))
  903     FORMAT (X, *PLANE=*, I3)
  904 FORMAT (1H0,*PRESSURE AT J=10*)
      RETURN
      END



*DECK VELOC
      SUBROUTINE VELOC (Z0,ZM1,ZM2,ZP1,ZP2,Z,ZLESS1,ZADD1,BETA,COEF6,
     1    COEF7,DT,COF,GAMMA,GAMMAS,GCOF,GO2,GD,DELTA,G6)
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
C**************************************************************************
C THIS SUBROUTINE COMPUTES H(I) IN EQUATION (3.32).                       *
C**************************************************************************
      INTEGER Z0,ZM1,ZM2,ZP1,ZP2,Z,ZLESS1,ZADD1
      REAL K, NLEN,NAVG
      DIMENSION U(16,16,5),V(16,16,5),W(16,16,5),SIG(16,16,3),K(16,16,3)
     *    ,P(16,16,5) ,DUMMY(16,16,6)
      COMMON/DATA1/U,V,W
      COMMON/DATA2/P
      COMMON/DATA3/SIG,K
      COMMON/DATA4/DUMMY
      COMMON/DATA9/IMAX,JMAX,LMAX,NHALF,NAVG,NLEN,NSPEC
      RDIV=1./BETA
      JTEST=JMAX-1
      ITEST=IMAX-1
      DO 210  J=1,JMAX
      JJ=J
      JM1=J-1
      JM2=J-2
      JP1=J+1
      JP2=J+2
      IF (J .GT. 2) GO TO 30
      GO TO (10,20),J
   10 JM2=JMAX-1
      JM1=JMAX
      GO TO 60
   20 JM2=JMAX
      GO TO 60
   30 IF (JJ .LT. JTEST) GO TO 60
      J1=JJ-JMAX+2
      GO TO (40,50),J1
   40 JP2=1
      GO TO 60
   50 JP1=1
      JP2=2
   60 CONTINUE
      Y=(J-7.5)*DELTA
      YM=Y-DELTA
      YP=Y+DELTA
      DO 200 I=1,IMAX
      IM2=I-2
      IM1=I-1
      IP1=I+1
      IP2=I+2
      IF (I .GT.2) GO TO 90
      GO TO (70,80), I
   70 IM2=IMAX-1
      IM1=IMAX
      GO TO 120
   80 IM2=IMAX
      GO TO 120
   90 IF (I .LT. ITEST) GO TO 120
      I1=I-IMAX+2
      GO TO (100,110), I1
  100 IP2=1
      GO TO 120
  110 IP1=1
      IP2=2
  120 CONTINUE
      X=(I-7.5)*DELTA
      XM=X-DELTA
      XP=X+DELTA
```

```
C
C      U COMPONENT
C
C-----ADVECTION TERM---U COMPONENT
      S1=U(IP1,J,Z0)*U(IP1,J,Z0)-U(IM1,J,Z0)*U(IM1,J,Z0)
      S1=S1+U(I,J,Z0)*(U(IP1,J,Z0)-U(IM1,J,Z0))
      S1=S1+U(I,J,Z0)*(U(IP1,J,Z0)-U(IM1,J,Z0))
      S1=S1+U(I,JP1,Z0)*V(I,JP1,Z0)-U(I,JM1,Z0)*V(I,JM1,Z0) +V(I,J,Z0)*
     *(U(I,JP1,Z0)-U(I,JM1,Z0))+U(I,J,Z0)*(V(I,JP1,Z0)-V(I,JM1,Z0))
      S1=S1+U(I,J,ZP1)*W(I,J,ZP1)-U(I,J,ZM1)*W(I,J,ZM1)+W(I,J,Z0)*(
     .U(I,J,ZP1)-U(I,J,ZM1))+U(I,J,Z0)*(W(I,J,ZP1)-W(I,J,ZM1))
      S2=U(IP2,J,Z0)*U(IP2,J,Z0)-U(IM2,J,Z0)*U(IM2,J,Z0)+U(I,J,Z0)*(
     .U(IP2,J,Z0)-U(IM2,J,Z0))+U(I,J,Z0)*(U(IP2,J,Z0)-U(IM2,J,Z0))
      S2=S2+U(I,JP2,Z0)*V(I,JP2,Z0)-U(I,JM2,Z0)*V(I,JM2,Z0)+V(I,J,Z0)*(
     .U(I,JP2,Z0)-U(I,JM2,Z0))+U(I,J,Z0)*(V(I,JP2,Z0)-V(I,JM2,Z0))
      S2=S2+U(I,J,ZP2)*W(I,J,ZP2)-U(I,J,ZM2)*W(I,J,ZM2)+W(I,J,Z0)*(
     .U(I,J,ZP2)-U(I,J,ZM2))+U(I,J,Z0)*(W(I,J,ZP2)-W(I,J.ZM2))
      ADVEC =-16.*S1+2.*S2
      S1=U(IM2,J,Z0)-8.*(U(IM1,J,Z0)-U(IP1,J,Z0))-U(IP2,J,Z0)
      S2=U(I,JM2,Z0)-8.*(U(I,JM1,Z0)-U(I,JP1,Z0))-U(I,JP2,Z0)
      S3 =   GCOF*(Y*S2-X*S1)-GAMMA*U(I,J,Z0)
      ADVEC=ADVEC+RDIV*S3
C-----RESOLVABLE SCALE LOSS TERM (LEONARD TERM )---U COMPONENT
      S31=U(IP2,J,Z0)*U(IP2,J,Z0)+U(IP1,J,Z0)*(U(IP2,J,Z0)-U(I,J,Z0))
      S31=S31+U(IP1,J,Z0)*(U(IP2,J,Z0)-U(I,J,Z0))
      S31=S31+U(IP1,JP1,Z0)*V(IP1,JP1,Z0)-U(IP1,JM1,Z0)*V(IP1,JM1,Z0)
      S31=S31+V(IP1,J,Z0)*(U(IP1,JP1,Z0)-U(IP1,JM1,Z0))
      S31=S31     +U(IP1,J,Z0)*(V(IP1,JP1,Z0)-V(IP1,JM1,Z0))
      S31=S31+U(IP1,J,ZP1)*W(IP1,J,ZP1)-U(IP1,J,ZM1)*W(IP1,J,ZM1)
      S31=S31+W(IP1,J,Z0)*(U(IP1,J,ZP1)-U(IP1,J,ZM1))
      S31=S31+U(IP1,J,Z0)*(W(IP1,J,ZP1)-W(IP1,J,ZM1))
      S32=-U(IM2,J,Z0)*U(IM2,J,Z0)+U(IM1,J,Z0)*(U(I,J,Z0)-U(IM2,J,Z0))
      S32=S32+U(IM1,J,Z0)*(U(I,J,Z0)-U(IM2,J,Z0))
      S32=S32+U(IM1,JP1,Z0)*V(IM1,JP1,Z0)-U(IM1,JM1,Z0)*V(IM1,JM1,Z0)
      S32=S32+V(IM1,J,Z0)*(U(IM1,JP1,Z0)-U(IM1,JM1,Z0))
      S32=S32          +U(IM1,J,Z0)*(V(IM1,JP1,Z0)-V(IM1,JM1,Z0))
      S32=S32+U(IM1,J,ZP1)*W(IM1,J,ZP1)-U(IM1,J,ZM1)*W(IM1,J,ZM1)
      S32=S32+W(IM1,J,Z0)*(U(IM1,J,ZP1)-U(IM1,J,ZM1))
      S32=S32+U(IM1,J,Z0)*(W(IM1,J,ZP1)-W(IM1,J,ZM1))
      S41=U(IP1,JP1,Z0)*U(IP1,JP1,Z0)-U(IM1,JP1,Z0)*U(IM1,JP1,Z0)
      S41=S41+U(I,JP1,Z0)*(U(IP1,JP1,Z0)-U(IM1,JP1,Z0))
      S41=S41+U(I,JP1,Z0)*(U(IP1,JP1,Z0)-U(IM1,JP1,Z0))
      S41=S41          +U(I,JP2,Z0)*V(I,JP2,Z0)+V(I,JP1,Z0)*(U(I,JP2,Z0)-
     .U(I,J,Z0))+U(I,JP1,Z0)*(V(I,JP2,Z0)-V(I,J,Z0))
      S41=S41+U(I,JP1,ZP1)*W(I,JP1,ZP1)-U(I,JP1,ZM1)*W(I,JP1,ZM1)
      S41=S41+W(I,JP1,Z0)*(U(I,JP1,ZP1)-U(I,JP1,ZM1))
      S41=S41+U(I,JP1,Z0)*(W(I,JP1,ZP1)-W(I,JP1,ZM1))
      S42=U(IP1,JM1,Z0)*U(IP1,JM1,Z0)-U(IM1,JM1,Z0)*U(IM1,JM1,Z0)
      S42=S42+U(I,JM1,Z0)*(U(IP1,JM1,Z0)-U(IM1,JM1,Z0))
      S42=S42+U(I,JM1,Z0)*(U(IP1,JM1,Z0)-U(IM1,JM1,Z0))
      S42=S42-U(I,JM2,Z0)*V(I,JM2,Z0)+V(I,JM1,Z0)*(U(I,J,Z0)-U(I,JM2,Z0)
     1)
      S42=S42+U(I,JM1,Z0)*(V(I,J,Z0)-V(I,JM2,Z0))
      S42=S42+U(I,JM1,ZP1)*W(I,JM1,ZP1)-U(I,JM1,ZM1)*W(I,JM1,ZM1)
      S42=S42+W(I,JM1,Z0)*(U(I,JM1,ZP1)-U(I,JM1,ZM1))
      S42=S42+U(I,JM1,Z0)*(W(I,JM1,ZP1)-W(I,JM1,ZM1))
      S51=U(IP1,J,ZP1)*U(IP1,J,ZP1)-U(IM1,J,ZP1)*U(IM1,J,ZP1)+U(I,J,ZP1)
     .*(U(IP1,J,ZP1)-U(IM1,J,ZP1))
      S51=S51                          +U(I,J,ZP1)*(U(IP1,J,ZP1)-U(IM1,J,ZP1)
     .)+U(I,JP1,ZP1)*V(I,JP1,ZP1)-U(I,JM1,ZP1)*V(I,JM1,ZP1)
      S51=S51+V(I,J,ZP1)*(U(I,JP1,ZP1)-U(I,JM1,ZP1))+U(I,J,ZP1)*(V(I,JP1
     .,ZP1)-V(I,JM1,ZP1))
```

```
      S51=S51            +U(I,J,ZP2)*W(I,J,ZP2)+W(I,J,ZP1)*(U(I,J,ZP2)-
     .U(I,J,Z0))+U(I,J,ZP1)*(W(I,J,ZP2)-W(I,J,Z0))
      S52=U(IP1,J,ZM1)*U(IP1,J,ZM1)-U(IM1,J,ZM1)*U(IM1,J,ZM1)+U(I,J,ZM1)
     .*(U(IP1,J,ZM1)-U(IM1,J,ZM1))
      S52=S52                  +U(I,J,ZM1)*(U(IP1,J,ZM1)-U(IM1,J,ZM1)
     .)+U(I,JP1,ZM1)*V(I,JP1,ZM1)-U(I,JM1,ZM1)*V(I,JM1,ZM1)
      S52=S52+V(I,J,ZM1)*(U(I,JP1,ZM1)-U(I,JM1,ZM1))+U(I,J,ZM1)*(V(I,
     .JP1,ZM1)-V(I,JM1,ZM1))
      S52=S52              -U(I,J,ZM2)*W(I,J,ZM2)+W(I,J,ZM1)*(U(I,J,Z0)
     .-U(I,J,ZM2))+U(I,J,ZM1)*(W(I,J,Z0)-W(I,J,ZM2))
      S6=U(IP1,J,Z0)*U(IP1,J,Z0)-U(IM1,J,Z0)*U(IM1,J,Z0)
      S6=S6+U(I,J,Z0)*(U(IP1,J,Z0)-U(IM1,J,Z0))
      S6=S6                +U(I,J,Z0)*(U(IP1,J,Z0)-U(IM1,J,Z0))
      S6=S6+U(I,JP1,Z0)*V(I,JP1,Z0)-U(I,JM1,Z0)*V(I,JM1,Z0)
      S6=S6+V(I,J,Z0)*(U(I,JP1,Z0)-U(I,JM1,Z0))
      S6=S6                 +U(I,J,Z0)*(V(I,JP1,Z0)-V(I,JM1,Z0))
      S6=S6+U(I,J,ZP1)*W(I,J,ZP1)-U(I,J,ZM1)*W(I,J,ZM1)+W(I,J,Z0)*(U(I,J
     .,ZP1)-U(I,J,ZM1))+U(I,J,Z0)*(W(I,J,ZP1)-W(I,J,ZM1))
      RESOLV=-0.5*(S31+S32+S41+S42+S51+S52 -6.*S6)
      RESOLV=RESOLV*HLFM
      S1= -XP*(U(IP2,J,Z0)-U(I,J,Z0))+Y*(U(IP1,JP1,Z0)-U(IP1,JM1,Z0))
      S2=-XM*(U(I,J,Z0)-U(IM2,J,Z0))+Y*(U(IM1,JP1,Z0)-U(IM1,JM1,Z0))
      S3=-X*(U(IP1,JP1,Z0)-U(IM1,JP1,Z0))+YP*(U(I,JP2,Z0)-U(I,J,Z0))
      S4=-X*(U(IP1,JM1,Z0)-U(IM1,JM1,Z0))+YM*(U(I,J,Z0)-U(I,JM2,Z0))
      S5=-X*(U(IP1,J,ZP1)-U(IM1,J,ZP1))+Y*(U(I,JP1,ZP1)-U(I,JM1,ZP1))
      S6=-X*(U(IP1,J,ZM1)-U(IM1,J,ZM1))+Y*(U(I,JP1,ZM1)-U(I,JM1,ZM1))
      S7=GCOF*(S1+S2+S3+S4+S5+S6)
      S8=-G6*(U(IP1,J,Z0)+U(I,JP1,Z0)+U(I,J,ZP1)+U(IM1,J,Z0)+U(I,JM1,Z0)
     1   +U(I,J,ZM1))
      S9=GP2*( (U(IP1,J,Z0)-U(IM1,J,Z0))*X-(U(I,JP1,Z0)-U(I,JM1,Z0))*Y)
     1   +GAMMA*U(I,J,Z0)
      RESADD=(S7+S8+S9)*RDIV
C-----SUB-GRID-SCALE MODEL
      S71= K(IP1,J,Z )*(U(IP2,J,Z0)-   U(I,J,Z0))*2.
      S71=S71 -K(IM1,J,Z )*2.*(U(I,J,Z0) -U(IM2,J,Z0))
      S8=COEF6*S71
      S91=            U(I,JP2,Z0)-U(I,J,Z0)+V(IP1,JP1,Z0) -V(IM1,JP1,Z0)
      S9=K(I,JP1,Z)*S91
      S71=            U(I,J,Z0)-U(I,JM2,Z0)+V(IP1,JM1,Z0)-V(IM1,JM1,Z0)
      S71=K(I,JM1,Z)*S71
      S9=S9-S71
      S71=             U(I,J,ZP2)-U(I,J,Z0)+W(IP1,J,ZP1)-W(IM1,J,ZP1)
      S71=K(I,J,ZADD1)*S71
      S=               U(I,J,Z0)-U(I,J,ZM2)+W(IP1,J,ZM1)-W(IM1,J,ZM1)
      S=K(I,J,ZLESS1)*S
      S9=COEF6*(S9+S71-S)
      SGS =  (GD*(K(IP1,J,Z)-K(IM1,J,Z)))*RDIV+S8+S9
      DUMMY(I,J,1)=BETA*(ADVEC+RESOLV+SGS+RESADD)
      DUMMY(I,J,4)=ADVECS+RESADD+SGSADD
      SIG(I,J,1)=-RESOLV*U(I,J,Z0)
      SIG(I,J,2)=-SGS*U(I,J,Z0)
C
C     V COMPONENT
C
C-----ADVECTION TERM---V COMPONENT
      S1=V(IP1,J,Z0)*U(IP1,J,Z0)-V(IM1,J,Z0)*U(IM1,J,Z0)
      S1=S1+U(I,J,Z0)*(V(IP1,J,Z0)-V(IM1,J,Z0))
      S1=S1+V(I,J,Z0)*(U(IP1,J,Z0)-U(IM1,J,Z0))
      S1=S1+V(I,JP1,Z0)*V(I,JP1,Z0)-V(I,JM1,Z0)*V(I,JM1,Z0) +V(I,J,Z0)*
     *(V(I,JP1,Z0)-V(I,JM1,Z0))+V(I,J,Z0)*(V(I,JP1,Z0)-V(I,JM1,Z0))
      S1=S1+V(I,J,ZP1)*W(I,J,ZP1)-V(I,J,ZM1)*W(I,J,ZM1)+W(I,J,Z0)*(
     .V(I,J,ZP1)-V(I,J,ZM1))+V(I,J,Z0)*(W(I,J,ZP1)-W(I,J,ZM1))
```

```
      S2=V(IP2,J,Z0)*U(IP2,J,Z0)-V(IM2,J,Z0)*U(IM2,J,Z0)+U(I,J,Z0)*(
     .V(IP2,J,Z0)-V(IM2,J,Z0))+V(I,J,Z0)*(U(IP2,J,Z0)-U(IM2,J,Z0))
      S2=S2+V(I,JP2,Z0)*V(I,JP2,Z0)-V(I,JM2,Z0)*V(I,JM2,Z0)+V(I,J,Z0)*(
     .V(I,JP2,Z0)-V(I,JM2,Z0))+V(I,J,Z0)*(V(I,JP2,Z0)-V(I,JM2,Z0))
      S2=S2+V(I,J,ZP2)*W(I,J,ZP2)-V(I,J,ZM2)*W(I,J,ZM2)+W(I,J,Z0)*(
     .V(I,J,ZP2)-V(I,J,ZM2))+V(I,J,Z0)*(W(I,J,ZP2)-W(I,J,ZM2))
      ADVEC =-16.*S1+2.*S2
      S1=V(IM2,J,Z0)-8.*(V(IM1,J,Z0)-V(IP1,J,Z0))-V(IP2,J,Z0)
      S2=V(I,JM2,Z0)-8.*(V(I,JM1,Z0)-V(I,JP1,Z0))-V(I,JP2,Z0)
      ADVEC=(GCOF*(Y*S2-X*S1)+GAMMA*V(I,J,Z0))+ADIV+ADVEC
C------RESOLVABLE SCALE LOSS TERM (LEONARD TERM )---V COMPONENT
      S31=V(IP2,J,Z0)*U(IP2,J,Z0)+U(IP1,J,Z0)*(V(IP2,J,Z0)-V(I,J,Z0))
      S31=S31+V(IP1,J,Z0)*(U(IP2,J,Z0)-U(I,J,Z0))
      S31=S31+V(IP1,JP1,Z0)*V(IP1,JP1,Z0)-V(IP1,JM1,Z0)*V(IP1,JM1,Z0)
      S31=S31+V(IP1,J,Z0)*(V(IP1,JP1,Z0)-V(IP1,JM1,Z0))
      S31=S31     +V(IP1,J,Z0)*(V(IP1,JP1,Z0)-V(IP1,JM1,Z0))
      S31=S31+V(IP1,J,ZP1)*W(IP1,J,ZP1)-V(IP1,J,ZM1)*W(IP1,J,ZM1)
      S31=S31+W(IP1,J,Z0)*(V(IP1,J,ZP1)-V(IP1,J,ZM1))
      S31=S31+V(IP1,J,Z0)*(W(IP1,J,ZP1)-W(IP1,J,ZM1))
      S32=-V(IM2,J,Z0)*U(IM2,J,Z0)+U(IM1,J,Z0)*(V(I,J,Z0)-V(IM2,J,Z0))
      S32=S32+V(IM1,J,Z0)*(U(I,J,Z0)-U(IM2,J,Z0))
      S32=S32+V(IM1,JP1,Z0)*V(IM1,JP1,Z0)-V(IM1,JM1,Z0)*V(IM1,JM1,Z0)
      S32=S32+V(IM1,J,Z0)*(V(IM1,JP1,Z0)-V(IM1,JM1,Z0))
      S32=S32          +V(IM1,J,Z0)*(V(IM1,JP1,Z0)-V(IM1,JM1,Z0))
      S32=S32+V(IM1,J,ZP1)*W(IM1,J,ZP1)-V(IM1,J,ZM1)*W(IM1,J,ZM1)
      S32=S32+W(IM1,J,Z0)*(V(IM1,J,ZP1)-V(IM1,J,ZM1))
      S32=S32+V(IM1,J,Z0)*(W(IM1,J,ZP1)-W(IM1,J,ZM1))
      S41=V(IP1,JP1,Z0)*U(IP1,JP1,Z0)-V(IM1,JP1,Z0)*U(IM1,JP1,Z0)
      S41=S41+U(I,JP1,Z0)*(V(IP1,JP1,Z0)-V(IM1,JP1,Z0))
      S41=S41+V(I,JP1,Z0)*(U(IP1,JP1,Z0)-U(IM1,JP1,Z0))
      S41=S41          +V(I,JP2,Z0)*V(I,JP2,Z0)+V(I,JP1,Z0)*(V(I,JP2,Z0)-
     .V(I,J,Z0))+V(I,JP1,Z0)*(V(I,JP2,Z0)-V(I,J,Z0))
      S41=S41+V(I,JP1,ZP1)*W(I,JP1,ZP1)-V(I,JP1,ZM1)*W(I,JP1,ZM1)
      S41=S41+W(I,JP1,Z0)*(V(I,JP1,ZP1)-V(I,JP1,ZM1))
      S41=S41+V(I,JP1,Z0)*(W(I,JP1,ZP1)-W(I,JP1,ZM1))
      S42=V(IP1,JM1,Z0)*U(IP1,JM1,Z0)-V(IM1,JM1,Z0)*U(IM1,JM1,Z0)
      S42=S42+U(I,JM1,Z0)*(V(IP1,JM1,Z0)-V(IM1,JM1,Z0))
      S42=S42+V(I,JM1,Z0)*(U(IP1,JM1,Z0)-U(IM1,JM1,Z0))
      S42=S42-V(I,JM2,Z0)*V(I,JM2,Z0)+V(I,JM1,Z0)*(V(I,J,Z0)-V(I,JM2,Z0)
     1)
      S42=S42+V(I,JM1,Z0)*(V(I,J,Z0)-V(I,JM2,Z0))
      S42=S42+V(I,JM1,ZP1)*W(I,JM1,ZP1)-V(I,JM1,ZM1)*W(I,JM1,ZM1)
      S42=S42+W(I,JM1,Z0)*(V(I,JM1,ZP1)-V(I,JM1,ZM1))
      S42=S42+V(I,JM1,Z0)*(W(I,JM1,ZP1)-W(I,JM1,ZM1))
      S51=V(IP1,J,ZP1)*U(IP1,J,ZP1)-V(IM1,J,ZP1)*U(IM1,J,ZP1)+U(I,J,ZP1)
     .*(V(IP1,J,ZP1)-V(IM1,J,ZP1))
      S51=S51                         +V(I,J,ZP1)*(U(IP1,J,ZP1)-U(IM1,J,ZP1)
     .)+V(I,JP1,ZP1)*V(I,JP1,ZP1)-V(I,JM1,ZP1)*V(I,JM1,ZP1)
      S51=S51+V(I,J,ZP1)*(V(I,JP1,ZP1)-V(I,JM1,ZP1))+V(I,J,ZP1)*(V(I,JP1
     .,ZP1)-V(I,JM1,ZP1))
      S51=S51                  +V(I,J,ZP2)*W(I,J,ZP2)+W(I,J,ZP1)*(V(I,J,ZP2)-
     .V(I,J,Z0))+V(I,J,ZP1)*(W(I,J,ZP2)-W(I,J,Z0))
      S52=V(IP1,J,ZM1)*U(IP1,J,ZM1)-V(IM1,J,ZM1)*U(IM1,J,ZM1)+U(I,J,ZM1)
     .*(V(IP1,J,ZM1)-V(IM1,J,ZM1))
      S52=S52                         +V(I,J,ZM1)*(U(IP1,J,ZM1)-U(IM1,J,ZM1)
     .)+V(I,JP1,ZM1)*V(I,JP1,ZM1)-V(I,JM1,ZM1)*V(I,JM1,ZM1)
      S52=S52+V(I,J,ZM1)*(V(I,JP1,ZM1)-V(I,JM1,ZM1))+V(I,J,ZM1)*(V(I,
     .JP1,ZM1)-V(I,JM1,ZM1))
      S52=S52               -V(I,J,ZM2)*W(I,J,ZM2)+W(I,J,ZM1)*(V(I,J,Z0)
     .-V(I,J,ZM2))+V(I,J,ZM1)*(W(I,J,Z0)-W(I,J,ZM2))
      S6=V(IP1,J,Z0)*U(IP1,J,Z0)-V(IM1,J,Z0)*U(IM1,J,Z0)
      S6=S6+U(I,J,Z0)*(V(IP1,J,Z0)-V(IM1,J,Z0))
```

```
      S6=S6                        +V(I,J,Z0)*(U(IP1,J,Z0)-U(IM1,J,Z0))
      S6=S6+V(I,JP1,Z0)*V(I,JP1,Z0)-V(I,JM1,Z0)*V(I,JM1,Z0)
      S6=S6+V(I,J,Z0)*(V(I,JP1,Z0)-V(I,JM1,Z0))
      S6=S6                          +V(I,J,Z0)*(V(I,JP1,Z0)-V(I,JM1,Z0))
      S6=S6+V(I,J,ZP1)*W(I,J,ZP1)-V(I,J,ZM1)*W(I,J,ZM1)+W(I,J,Z0)*(V(I,J
     *,ZP1)-V(I,J,ZM1))+V(I,J,Z0)*(W(I,J,ZP1)-W(I,J,ZM1))
      RESOLV=-0.5*(S31+S32+S41+S42+S51+S52 -6.*S6)
      RESOLV=RESOLV*ALEN
      S1= -XP*(V(IP2,J,Z0)-V(I,J,Z0))+Y*(V(IP1,JP1,Z0)-V(IP1,JM1,Z0))
      S2=-XM*(V(I,J,Z0)-V(IM2,J,Z0))+Y*(V(IM1,JP1,Z0)-V(IM1,JM1,Z0))
      S3=-X*(V(IP1,JP1,Z0)-V(IM1,JP1,Z0))+YP*(V(I,JP2,Z0)-V(I,J,Z0))
      S4=-X*(V(IP1,JM1,Z0)-V(IM1,JM1,Z0))+YM*(V(I,J,Z0)-V(I,JM2,Z0))
      S5=-X*(V(IP1,J,ZP1)-V(IM1,J,ZP1))+Y*(V(I,JP1,ZP1)-V(I,JM1,ZP1))
      S6=-X*(V(IP1,J,ZM1)-V(IM1,J,ZM1))+Y*(V(I,JP1,ZM1)-V(I,JM1,ZM1))
      S7=GCOF*(S1+S2+S3+S4+S5+S6)
      S8= G6*(V(IP1,J,Z0)+V(I,JP1,Z0)+V(I,J,ZP1)+V(IM1,J,Z0)+V(I,JM1,Z0)
     1    +V(I,J,ZM1))
      S9=GOP*( (V(IP1,J,Z0)-V(IM1,J,Z0))*X-(V(I,JP1,Z0)-V(I,JM1,Z0))*Y)
     1    -GAMMA*V(I,J,Z0)
      RESADD=(S7+S8+S9)*RDIV
C-----SUB-GRID-SCALE MODEL
      SS=                V(IP2,J,Z0)-V(I,J,Z0)+U(IP1,JP1,Z0)-U(IP1,JM1,Z0)
      S=K(IP1,J,Z)*SS
      SS1=                 V(I,J,Z0)-V(IM2,J,Z0)+U(IM1,JP1,Z0)-U(IM1,JM1,Z0)
      S=S-K(IM1,J,Z)*SS1
      S8=COEF6*S
      S9=K(I,JP1,Z)*2.*(V(I,JP2,Z0)-V(I,J,Z0))
      S9=S9-K(I,JM1,Z)*2.*(V(I,J,Z0)-V(I,JM2,Z0))
      S9=S9+K(I,J,ZADD1)*(V(I,J,ZP2)-V(I,J,Z0))
      S9=S9+K(I,J,ZADD1)*(W(I,JP1,ZP1)-W(I,JM1,ZP1))
      S=                      V(I,J,Z0)-V(I,J,ZM2)+W(I,JP1,ZM1)-W(I,JM1,ZM1)
      S=K(I,J,ZLESS1)*S
      S9=COEF6*(S9-S)
      SGS=-GC*   (K(I,JP1,Z)-K(I,JM1,Z))*RDIV+S8+S9
      DUMMY(I,J,2)=BETA*(ADVEC+RESOLV+SGS+RESADD)
      DUMMY(I,J,5)=ADVECS+RESADD+SGSADD
      SIG(I,J,1)=SIG(I,J,1)-RESOLV*V(I,J,Z0)
      SIG(I,J,2)=SIG(I,J,2)-SGS*V(I,J,Z0)
C
C       W COMPONENT
C
C-----ADVECTION TERM----W COMPONENT
      S1=W(IP1,J,Z0)*U(IP1,J,Z0)-W(IM1,J,Z0)*U(IM1,J,Z0)
      S1=S1+U(I,J,Z0)*(W(IP1,J,Z0)-W(IM1,J,Z0))
      S1=S1+W(I,J,Z0)*(U(IP1,J,Z0)-U(IM1,J,Z0))
      S1=S1+W(I,JP1,Z0)*V(I,JP1,Z0)-W(I,JM1,Z0)*V(I,JM1,Z0) +V(I,J,Z0)*
     *(W(I,JP1,Z0)-W(I,JM1,Z0))+W(I,J,Z0)*(V(I,JP1,Z0)-V(I,JM1,Z0))
      S1=S1+W(I,J,ZP1)*W(I,J,ZP1)-W(I,J,ZM1)*W(I,J,ZM1)+W(I,J,Z0)*(
     * W(I,J,ZP1)-W(I,J,ZM1))+W(I,J,Z0)*(W(I,J,ZP1)-W(I,J,ZM1))
      S2=W(IP2,J,Z0)*U(IP2,J,Z0)-W(IM2,J,Z0)*U(IM2,J,Z0)+U(I,J,Z0)*(
     *W(IP2,J,Z0)-W(IM2,J,Z0))+W(I,J,Z0)*(U(IP2,J,Z0)-U(IM2,J,Z0))
      S2=S2+W(I,JP2,Z0)*V(I,JP2,Z0)-W(I,JM2,Z0)*V(I,JM2,Z0)+V(I,J,Z0)*(
     *W(I,JP2,Z0)-W(I,JM2,Z0))+W(I,J,Z0)*(V(I,JP2,Z0)-V(I,JM2,Z0))
      S2=S2+W(I,J,ZP2)*W(I,J,ZP2)-W(I,J,ZM2)*W(I,J,ZM2)+W(I,J,Z0)*(
     *W(I,J,ZP2)-W(I,J,ZM2))+W(I,J,Z0)*(W(I,J,ZP2)-W(I,J,ZM2))
      S3 =    -16.*S1+2.*S2
      S1=W(IM2,J,Z0)-8.*(W(IM1,J,Z0)-W(IP1,J,Z0))-W(IP2,J,Z0)
      S2=W(I,JM2,Z0)-8.*(W(I,JM1,Z0)-W(I,JP1,Z0))-W(I,JP2,Z0)
      ADVEC =GCOF*(Y*S2-X*S1)*RDIV+S3
C-----RESOLVABLE SCALE LOSS TERM (LEONARD TERM )----W COMPONENT
      S31=W(IP2,J,Z0)*U(IP2,J,Z0)+U(IP1,J,Z0)*(W(IP2,J,Z0)-W(I,J,Z0))
      S31=S31+W(IP1,J,Z0)*(U(IP2,J,Z0)-U(I,J,Z0))
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
S31=S31+W(IP1,JP1,Z0)*V(IP1,JP1,Z0)-W(IP1,JM1,Z0)*V(IP1,JM1,Z0)
S31=S31+V(IP1,J,Z0)*(W(IP1,JP1,Z0)-W(IP1,JM1,Z0))
S31=S31      +W(IP1,J,Z0)*(V(IP1,JP1,Z0)-V(IP1,JM1,Z0))
S31=S31+W(IP1,J,ZP1)*W(IP1,J,ZP1)-W(IP1,J,ZM1)*W(IP1,J,ZM1)
S31=S31+W(IP1,J,Z0)*(W(IP1,J,ZP1)-W(IP1,J,ZM1))
S31=S31+W(IP1,J,Z0)*(W(IP1,J,ZP1)-W(IP1,J,ZM1))
S32=-W(IM2,J,Z0)*U(IM2,J,Z0)+U(IM1,J,Z0)*(W(I,J,Z0)-W(IM2,J,Z0))
S32=S32+W(IM1,J,Z0)*(U(I,J,Z0)-U(IM2,J,Z0))
S32=S32+W(IM1,JP1,Z0)*V(IM1,JP1,Z0)-W(IM1,JM1,Z0)*V(IM1,JM1,Z0)
S32=S32+V(IM1,J,Z0)*(W(IM1,JP1,Z0)-W(IM1,JM1,Z0))
S32=S32      +W(IM1,J,Z0)*(V(IM1,JP1,Z0)-V(IM1,JM1,Z0))
S32=S32+W(IM1,J,ZP1)*W(IM1,J,ZP1)-W(IM1,J,ZM1)*W(IM1,J,ZM1)
S32=S32+W(IM1,J,Z0)*(W(IM1,J,ZP1)-W(IM1,J,ZM1))
S32=S32+W(IM1,J,Z0)*(W(IM1,J,ZP1)-W(IM1,J,ZM1))
S41=W(IP1,JP1,Z0)*U(IP1,JP1,Z0)-W(IM1,JP1,Z0)*U(IM1,JP1,Z0)
S41=S41+U(I,JP1,Z0)*(W(IP1,JP1,Z0)-W(IM1,JP1,Z0))
S41=S41+W(I,JP1,Z0)*(U(IP1,JP1,Z0)-U(IM1,JP1,Z0))
S41=S41      +W(I,JP2,Z0)*V(I,JP2,Z0)+V(I,JP1,Z0)*(W(I,JP2,Z0)-
.W(I,J,Z0))+W(I,JP1,Z0)*(V(I,JP2,Z0)-V(I,J,Z0))
S41=S41+W(I,JP1,ZP1)*W(I,JP1,ZP1)-W(I,JP1,ZM1)*W(I,JP1,ZM1)
S41=S41+W(I,JP1,Z0)*(W(I,JP1,ZP1)-W(I,JP1,ZM1))
S41=S41+W(I,JP1,Z0)*(W(I,JP1,ZP1)-W(I,JP1,ZM1))
S42=W(IP1,JM1,Z0)*U(IP1,JM1,Z0)-W(IM1,JM1,Z0)*U(IM1,JM1,Z0)
S42=S42+U(I,JM1,Z0)*(W(IP1,JM1,Z0)-W(IM1,JM1,Z0))
S42=S42+W(I,JM1,Z0)*(U(IP1,JM1,Z0)-U(IM1,JM1,Z0))
S42=S42-W(I,JM2,Z0)*V(I,JM2,Z0)+V(I,JM1,Z0)*(W(I,J,Z0)-W(I,JM2,Z0)
1)
S42=S42+W(I,JM1,Z0)*(V(I,J,Z0)-V(I,JM2,Z0))
S42=S42+W(I,JM1,ZP1)*W(I,JM1,ZP1)-W(I,JM1,ZM1)*W(I,JM1,ZM1)
S42=S42+W(I,JM1,Z0)*(W(I,JM1,ZP1)-W(I,JM1,ZM1))
S42=S42+W(I,JM1,Z0)*(W(I,JM1,ZP1)-W(I,JM1,ZM1))
S51=W(IP1,J,ZP1)*U(IP1,J,ZP1)-W(IM1,J,ZP1)*U(IM1,J,ZP1)+U(I,J,ZP1)
.*(W(IP1,J,ZP1)-W(IM1,J,ZP1))
S51=S51                         +W(I,J,ZP1)*(U(IP1,J,ZP1)-U(IM1,J,ZP1)
.)+W(I,JP1,ZP1)*V(I,JP1,ZP1)-W(I,JM1,ZP1)*V(I,JM1,ZP1)
S51=S51+V(I,J,ZP1)*(W(I,JP1,ZP1)-W(I,JM1,ZP1))+W(I,J,ZP1)*(V(I,JP1
.,ZP1)-V(I,JM1,ZP1))
S51=S51                  +W(I,J,ZP2)*W(I,J,ZP2)+W(I,J,ZP1)*(W(I,J,ZP2)-
.W(I,J,Z0))+W(I,J,ZP1)*(W(I,J,ZP2)-W(I,J,Z0))
S52=W(IP1,J,ZM1)*U(IP1,J,ZM1)-W(IM1,J,ZM1)*U(IM1,J,ZM1)+U(I,J,ZM1)
.*(W(IP1,J,ZM1)-W(IM1,J,ZM1))
S52=S52                         +W(I,J,ZM1)*(U(IP1,J,ZM1)-U(IM1,J,ZM1)
.)+W(I,JP1,ZM1)*V(I,JP1,ZM1)-W(I,JM1,ZM1)*V(I,JM1,ZM1)
S52=S52+V(I,J,ZM1)*(W(I,JP1,ZM1)-W(I,JM1,ZM1))+W(I,J,ZM1)*(V(I,
.JP1,ZM1)-V(I,JM1,ZM1))
S52=S52                   -W(I,J,ZM2)*W(I,J,ZM2)+W(I,J,ZM1)*(W(I,J,Z0)
.-W(I,J,ZM2))+W(I,J,ZM1)*(W(I,J,Z0)-W(I,J,ZM2))
S6=W(IP1,J,Z0)*U(IP1,J,Z0)-W(IM1,J,Z0)*U(IM1,J,Z0)
S6=S6+U(I,J,Z0)*(W(IP1,J,Z0)-W(IM1,J,Z0))
S6=S6                    +W(I,J,Z0)*(U(IP1,J,Z0)-U(IM1,J,Z0))
S6=S6+W(I,JP1,Z0)*V(I,JP1,Z0)-W(I,JM1,Z0)*V(I,JM1,Z0)
S6=S6+V(I,J,Z0)*(W(I,JP1,Z0)-W(I,JM1,Z0))
S6=S6                     +W(I,J,Z0)*(V(I,JP1,Z0)-V(I,JM1,Z0))
S6=S6+W(I,J,ZP1)*W(I,J,ZP1)-W(I,J,ZM1)*W(I,J,ZM1)+W(I,J,Z0)*(W(I,J
.,ZP1)-W(I,J,ZM1))+W(I,J,Z0)*(W(I,J,ZP1)-W(I,J,ZM1))
RESOLV=-0.5*(S31+S32+S41+S42+S51+S52 -6.*S6)
RESOLV=RESOLV*RLEN
S1= XP*(W(IP2,J,Z0)-W(I,J,Z0))+Y*(W(IP1,JP1,Z0)-W(IP1,JM1,Z0))
S2=-XM*(W(I,J,Z0)-W(IM2,J,Z0))+Y*(W(IM1,JP1,Z0)-W(IM1,JM1,Z0))
S3=-X*(W(IP1,JP1,Z0)-W(IM1,JP1,Z0))+YP*(W(I,JP2,Z0)-W(I,J,Z0))
S4=-X*(W(IP1,JM1,Z0)-W(IM1,JM1,Z0))+YM*(W(I,J,Z0)-W(I,JM2,Z0))
S5=-X*(W(IP1,J,ZP1)-W(IM1,J,ZP1))+Y*(W(I,JP1,ZP1)-W(I,JM1,ZP1))
```

```
      S6=-X*(W(IP1,J,ZM1)-W(IM1,J,ZM1))+Y*(W(I,JP1,ZM1)-W(I,JM1,ZM1))
      S7=GCOF*(S1+S2+S3+S4+S5+S6)
      S9=GD2*( (W(IP1,J,Z0)-W(IM1,J,Z0))*X-(W(I,JP1,Z0)-W(I,JM1,Z0))*Y)
      RESADD=(S7+S9)*RDIV
C-----SUB-GRID-SCALE MODEL
      SS=             W(IP2,J,Z0) -W(I,J,Z0)+U(IP1,J,ZP1)-U(IP1,J,ZM1)
      S=K(IP1,J,Z )*SS
      SS1=             W(I,J,Z0)-W(IM2,J,Z0)+U(IM1,J,ZP1)-U(IM1,J,ZM1)
      S=S-K(IM1,J,Z )*SS1
      S8=COEF6*S
      S=              W(I,JP2,Z0)-W(I,J,Z0)+V(I,JP1,ZP1)-V(I,JP1,ZM1)
      S9=K(I,JP1,Z )*S
      SS=              W(I,J,Z0)-W(I,JM2,Z0)+V(I,JM1,ZP1)-V(I,JM1,ZM1)
      S9=S9-K(I,JM1,Z)*SS
      S9=S9 +K(I,J,ZADD1)*2.*(W(I,J,ZP2)-W(I,J,Z0))
      S9=COEF6*(S9-K(I,J,ZLESS1)*2.*(W(I,J,Z0) -W(I,J,ZM2)))
      SGS=S8+S9
      DUMMY(I,J,3)=BETA*(ADVEC+RESOLV+SGS+RESADD)
      DUMMY(I,J,6)=ADVECS+RESADD
      SIG(I,J,1)=SIG(I,J,1)-RESOLV*W(I,J,Z0)
      SIG(I,J,2)=SIG(I,J,2)-SGS*W(I,J,Z0)
  200 CONTINUE
  210 CONTINUE
      RETURN
      END




*DECK DIVGCE
      SUBROUTINE DIVGCE (Z0,ZM1,ZM2,ZP1,ZP2,COEF7)
C*********************************************************************
C THIS SUBROUTINE CALCULATES THE DIVERGENCE OF (U,V,W) FOR THE PLANE Z0*
C THEN THE VALUE IS STORED IN DUMMY(I,J,6).                           *
C*********************************************************************
      INTEGER Z0,ZM1,ZM2,ZP1,ZP2
      DIMENSION U(16,16,5),V(16,16,5),W(16,16,5),DUMMY(16,16,6)
      COMMON/DATA1/U,V,W
      COMMON/DATA4/DUMMY
      COMMON/DATA9/IMAX,JMAX,LMAX,NHALF,NAVG,NLEN,NSPEC
      JTEST=JMAX-1
      ITEST=IMAX-1
      DO 210  J=1,JMAX
      JJ=J
      JM2=J-2
      JM1=J-1
      JP1=J+1
      JP2=J+2
      IF (J .GT. 2) GO TO 30
      GO TO (10,20),J
   10 JM2=JMAX-1
      JM1=JMAX
      GO TO 60
   20 JM2=JMAX
      GO TO 60
   30 IF (JJ .LT. JTEST) GO TO 60
      J1=JJ-JMAX+2
      GO TO (40,50),J1
   40 JP2=1
      GO TO 60
   50 JP1=1
```

```
      JP2=2
   60 CONTINUE
      DO 200 I=1,IMAX
      IM2=I-2
      IM1=I-1
      IP1=I+1
      IP2=I+2
      IF (I .GT. 2) GO TO 90
      GO TO (70, 80),I
   70 IM2=IMAX-1
      IM1=IMAX
      GO TO 120
   80 IM2=IMAX
      GO TO 120
   90 IF (I .LT. ITEST) GO TO 120
      I1=I-IMAX+2
      GO TO (100,110), I1
  100 IP2=1
      GO TO 120
  110 IP1=1
      IP2=2
  120 CONTINUE
      DIV=U(IM2,J,Z0)-U(IP2,J,Z0)+V(I,JM2,Z0)-V(I,JP2,Z0)
      DIV=DIV+W(I,J,ZM2)-W(I,J,ZP2)
      S=U(IP1,J,Z0)-U(IM1,J,Z0)+V(I,JP1,Z0)-V(I,JM1,Z0)
      S=S+W(I,J,ZP1)-W(I,J,ZM1)
      DIV=DIV+8.*S
      DUMMY(I,J,6)=DIV*COEF7
  200 CONTINUE
  210 CONTINUE
      RETURN
      END


*DECK START
      SUBROUTINE START(COEF3,COEF11,DELTA)
C*************************************************************************
C THIS SUBROUTINE INITIATES THE CONSTANTS FOR FFT ROUTINES.              *
C*************************************************************************
      DIMENSION TRR(8,3),TRI(8,3),GR(16),GI(16),NWAVE(16),NFFT(3)
      COMMON/DATA5/GR,GI,TRR,TRI
      COMMON/DATA8/NWAVE,NFFT
      COMMON/DATA9/IMAX,JMAX,LMAX,MHALF,NAVG,NLEN,NSPEC
      DATA NWAVE/1,9,5,13,3,11,7,15,2,10,6,14,4,12,8,16/
      COEF3=1./16.**3
      COEF11=3.14159265358979/8.
      NFFT(1)=8
      NFFT(2)=4
      NFFT(3)=2
      DO 30 J=1,3
       TRR(1,J)=1.
       TRI(1,J)=0.
   30  CONTINUE
      DO 40 I=2,8
      B=FLOAT(I)-1.
      B=B*COEF11
      TRR(I,1)=COS(B)
      TRI(I,1)=-SIN(B)
   40 CONTINUE
```

```
      DO 50 I=2,4
      F=FLOAT(I)-1.
      B=2.*F*COEF11
      TRR(I,2)=COS(B)
      TRI(I,2)=-SIN(B)
   50 CONTINUE
      B=4.*COEF11
      TRR(2,3)=COS(B)
      TRI(2,3)=-SIN(B)
      RETURN
      END
```

```
*DECK FFTX
      SUBROUTINE FFTX(SIGN)
C***********************************************************************
C FAST FOURIER TRANSFORM IN X-DIRECTION                                 *
C***********************************************************************
      DIMENSION FR(16,16),FI(16,16),TRR(8,3),TRI(8,3),GR(16),GI(16)
     .    ,NWAVE(16),NFFT(3)
      COMMON/DATA5/GR,GI,TRR,TRI
      COMMON/DATA7/FR,FI
      COMMON/DATA8/NWAVE,NFFT
      COMMON/DATA9/IMAX,JMAX,LMAX,NHALF,NAVG,NLEN,NSPEC
      IF (SIGN .LT. 0.) GO TO 3
      DO 2 J=1,JMAX
      DO 1 I=1,IMAX
      FI(I,J)=0.
    1 CONTINUE
    2 CONTINUE
    3 CONTINUE
      DO 100 J=1,JMAX
      JP=J
      DO 20 MM=1,3
      IEND=0
      INCR=NFFT(MM)
      IP=0
    5 CONTINUE
      ISTART=1+IP
      IEND=ISTART+INCR-1
      M=1
      DO 10 I=ISTART,IEND
      IP=I+INCR
      GDUM1=FR(I,J)+FR(IP,JP)
      GDUM2=FI(I,J)+FI(IP,JP)
      GDUM3=FR(I,J)-FR(IP,JP)
      GDUM4=FI(I,J)-FI(IP,JP)
      GDUM5=GDUM3*TRR(M,MM)-GDUM4*TRI(M,MM)*SIGN
      GDUM6=GDUM3*TRI(M,MM)*SIGN+GDUM4*TRR(M,MM)
      FR(I,J)=GDUM1
      FI(I,J)=GDUM2
      FR(IP,JP)=GDUM5
      FI(IP,JP)=GDUM6
      M=M+1
   10 CONTINUE
      IF (IP .LT. IMAX) GO TO 5
   20 CONTINUE
C     FIFTH 1,2,3,  ETC.
   60 DO 70 I=1,IMAX,2
```

```
       IP=I+1
      GDUM1=FR(I,J)+FR(IP,JP)
      GDUM2=FI(I,J)+FI(IP,JP)
      GDUM3=FR(I,J)-FR(IP,JP)
      GDUM4=FI(I,J)-FI(IP,JP)
      GR(I)=GDUM1
      GI(I)=GDUM2
      GR(IP)=GDUM3
      GI(IP)=GDUM4
   70  CONTINUE
C     LET TRANSFORMED VALUE ORDERED
      DO 80 I=1,IMAX
       IM=NWAVE(I)
       FR(IM,J)=GR(I)
       FI(IM,J)=GI(I)
   80  CONTINUE
  100 CONTINUE
      RETURN
      END



*DECK FFTY
      SUBROUTINE FFTY(SIGN,COEF3)
C*****************************************************************************
C FAST FOURIER TRANSFORM IN Y-DIRECTION                                       *
C*****************************************************************************
      DIMENSION FR(16,16),FI(16,16),TRR(8,3),TRI(8,3),GR(16),GI(16)
     *  ,NWAVE(16),NFFT(3)
      COMMON/DATA5/GR,GI,TRR,TRI
      COMMON/DATA7/FR,FI
      COMMON/DATA8/NWAVE,NFFT
      COMMON/DATA9/IMAX,JMAX,LMAX,NHALF,NAVG,NLEN,NSPEC
C     Y-TRANSFORM
      DO 200 I=1,IMAX
      IP=I
      DO 120 MM=1,3
      IEND=0
      INCR=NFFT(MM)
      JP=0
  105 CONTINUE
      ISTART=1+JP
      IEND=ISTART+INCR-1
      M=1
      DO 110 J=ISTART,IEND
      JP=J+INCR
      GDUM1=FR(I,J)+FR(IP,JP)
      GDUM2=FI(I,J)+FI(IP,JP)
      GDUM3=FR(I,J)-FR(IP,JP)
      GDUM4=FI(I,J)-FI(IP,JP)
      GDUM5=GDUM3*TRR(M,MM)-GDUM4*TRI(M,MM)*SIGN
      GDUM6=GDUM3*TRI(M,MM)*SIGN+GDUM4*TRR(M,MM)
      FR(I,J)=GDUM1
      FI(I,J)=GDUM2
      FR(IP,JP)=GDUM5
      FI(IP,JP)=GDUM6
      M=M+1
  110 CONTINUE
      IF (JP .LT. JMAX) GO TO 105
  120 CONTINUE
```

```
      DO 170 J=1,JMAX,2
      JP=J+1
      GDUM1=FR(I,J)+FR(IP,JP)
      GDUM2=FI(I,J)+FI(IP,JP)
      GDUM3=FR(I,J)-FR(IP,JP)
      GDUM4=FI(I,J)-FI(IP,JP)
      IF (SIGN .LT. 0.) GO TO 160
      GR(J)=GDUM1
      GI(J)=GDUM2
      GR(JP)=GDUM3
      GI(JP)=GDUM4
      GO TO 170
  160 CONTINUE
      GR(J)=GDUM1*COEF3
      GI(J)=GDUM2*COEF3
      GR(JP)=GDUM3*COEF3
      GI(JP)=GDUM4*COEF3
  170 CONTINUE
C     GET ORDERED SET
      DO 180 J=1,JMAX
      JM=NWAVE(J)
      FR(I,JM)=GR(J)
      FI(I,JM)=GI(J)
  180 CONTINUE
  200 CONTINUE
      RETURN
      END
```

```
*DECK FFTZ
      SUBROUTINE FFTZ(SIGN,HM,H)
C************************************************************************
C FAST FOURIER TRANSFORM IN Z-DIRECTION                                 *
C************************************************************************
      DIMENSION HM(16,16,16), H(16,16,16),HDUM(16,2),TRR(8,3),TRI(8,3),
     -GR(16),GI(16),NWAVE(16),NFFT(3)
      COMMON/DATA5/GR,GI,TRR,TRI
      COMMON/DATA8/NWAVE,NFFT
      COMMON/DATA9/IMAX,JMAX,LMAX,NHALF,NAVG,NLEN,NSPEC
      DO 200 I=1,IMAX
      DO 200 J=1,JMAX
      DO 120 MM=1,3
      IEND=0
      INCR=NFFT(MM)
      LP=0
  105 CONTINUE
      ISTART=1+LP
      IEND=ISTART+INCR-1
      M=1
      DO 110 L=ISTART,IEND
      LP=L+INCR
      DUM1=HM(I,J,L)+HM(I,J,LP)
      DUM2=H(I,J,L)+H(I,J,LP)
      DUM3=HM(I,J,L)-HM(I,J,LP)
      DUM4=H(I,J,L)-H(I,J,LP)
      DUM5=DUM3*TRR(M,MM)-DUM4*TRI(M,MM)*SIGN
      DUM6=DUM3*TRI(M,MM)*SIGN+DUM4*TRR(M,MM)
      HM(I,J,L)=DUM1
      H(I,J,L)=DUM2
```

```
      HM(I,J,LP)=DUM5
      H(I,J,LP)=DUM6
      M=M+1
  110 CONTINUE
      IF (LP .LT. LMAX) GO TO 105
  120 CONTINUE
      DO 170 L=1,LMAX,2
      LP=L+1
      DUM1=H(I,J,L)+H(I,J,LP)
      DUM2=HM(I,J,L)+HM(I,J,LP)
      DUM3=H(I,J,L)-H(I,J,LP)
      DUM4=HM(I,J,L)-HM(I,J,LP)
      HDUM(L,1)=DUM1
      HDUM(L,2)=DUM2
      HDUM(LP,1)=DUM3
      HDUM(LP,2)=DUM4
  170 CONTINUE
C     GET ORDERED SET
      DO 180 L=1,LMAX
      LM=NWAVE(L)
      H(I,J,LM)=HDUM(L,1)
      HM(I,J,LM)=HDUM(L,2)
  180 CONTINUE
  200 CONTINUE
      RETURN
      END
```

```
DECK VISCVORT
      SUBROUTINE VISCV(Z0,ZM1,ZP1,Z,COEF2)
C***********************************************************************
C THIS SUBROUTINE CALCULATES THE EDDY VISCOSITY BY VORTICITY MODEL     *
C***********************************************************************
      INTEGER Z0,ZM1,ZP1,Z
      REAL K, NLEN,NAVG
      DIMENSION U(16,16,5),V(16,16,5),W(16,16,5),SIG(16,16,3),K(16,16,3)
      COMMON/DATA1/U,V,W
      COMMON/DATA3/SIG,K
      COMMON/DATA9/IMAX,JMAX,LMAX,NHALF,NAVG,NLEN,NSPEC
C     COEF2=C*DELTA*0.5
      DO 210 JJ=1,JMAX
      JM1=JJ-1
      JP1=JJ+1
      IF (JJ .EQ. 1) JM1=JMAX
      IF (JJ .EQ. JMAX) JP1=1
      DO 200 I=1,IMAX
      IM1=I-1
      IP1=I+1
      IF (I .EQ. 1) IM1=IMAX
      IF (I .EQ. IMAX) IP1=1
      S1= (W(I,JP1,Z0)-W(I,JM1,Z0)-V(I,JJ,ZP1)+V(I,JJ,ZM1))**2
      S2= (U(I,JJ,ZP1)-U(I,JJ,ZM1)-W(IP1,JJ,Z0)+W(IM1,JJ,Z0))**2
      S3=(V(IP1,JJ,Z0)-V(IM1,JJ,Z0)-U(I,JP1,Z0) +U(I,JM1,Z0))**2
      S4=S1+S2+S3
      K(I,JJ,Z)=COEF2*SQRT(S4)
  200 CONTINUE
  210 CONTINUE
      RETURN
      END
```

```
*DECK VISCSMAG
      SUBROUTINE VISCS(Z0,ZM1,ZP1,Z,COEF2)
C***********************************************************************
C THIS SUBROUTINE CALCULATES THE EDDY VISCOSITY BY SMAGORINSKY MODEL.  *
C***********************************************************************
      INTEGER Z0,ZM1,ZP1,Z
      REAL K, MLEN,MAVG
      DIMENSION U(16,16,5),V(16,16,5),W(16,16,5),SIG(16,16,3),K(16,16,3)
      COMMON/DATA1/U,V,W
      COMMON/DATA3/SIG,K
      COMMON/DATA9/IMAX,JMAX,LMAX,NHALF,NAVG,NLEN ,NSPEC
C     COEF2=C*DELTA*0.5
      JTEST=JMAX-1
      ITEST=IMAX-1
      DO 210 JJ=1,JMAX
      J=JJ
      JM2=JJ-2
      JM1=JJ-1
      JP1=JJ+1
      JP2=JJ+2
      IF (JJ .GT. 2) GO TO 30
      GO TO (10,20),JJ
   10 JM2=JMAX-1
      JM1=JMAX
      GO TO 60
   20 JM2=JMAX
      GO TO 60
   30 IF (JJ .LT. JTEST) GO TO 60
      J1=JJ-JMAX+2
      GO TO (40,50), J1
   40 JP2=1
      GO TO 60
   50 JP2=2
      JP1=1
   60 CONTINUE
      DO 200 I=1,IMAX
      IM2=I-2
      IM1=I-1
      IP1=I+1
      IP2=I+2
      IF (I .GT. 2) GO TO 90
      GO TO (70,80), I
   70 IM2=IMAX-1
      IM1=IMAX
      GO TO 120
   80 IM2=IMAX
      GO TO 120
   90 IF (I .LT. ITEST) GO TO 120
      I1=I-IMAX+2
      GO TO (100,110),I1
  100 IP2=1
      GO TO 120
  110 IP2=2
      IP1=1
  120 CONTINUE
      S1=(U(IP1,J,Z0)-U(IM1,J,Z0))**2 +(V(I,JP1,Z0)-V(I,JM1,Z0))**2
     1  +(W(I,J,ZP1)-W(I,J,ZM1))**2
```

```
      S2=(U(I,JP1,Z0)-U(I,JM1,Z0)+V(IP1,J,Z0)-V(IM1,J,Z0))**2
      S3=(U(I,J,ZP1)-U(I,J,ZM1)+W(IP1,J,Z0)-W(IM1,J,Z0))**2
      S4=(V(I,J,ZP1)-V(I,J,ZM1)+W(I,JP1,Z0)-W(I,JM1,Z0))**2
      S5=2.*S1+S2+S3+S4
      K(I,J,Z)=COEF2*SQRT(S5)
  200 CONTINUE
  210 CONTINUE
      RETURN
      END
```

```
* SAMPLE INPUT
  1 16 16 16  1 15  1
 1.5        0.00625   0.26      2.        0.        1000.     0.
 65.        130.      301.      386.      379.      325.      280.      248.
 214.       183.      89.       44.       22.       12.       6.4       3.5
 1.7        0.8       0.        0.        0.        0.        0.        0.
 65.        130.      301.      386.      379.      325.      280.      248.
 214.       183.      89.       44.       22.       12.       6.4       3.5
 1.7        0.8       0.        0.        0.        0.        0.        0.
```

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES

Arakawa, A., 1966, "Computational design of long-term numerical integration of the equations of fluid motions: I. Two-dimensional incompressible flow," J. Comp. Phys., Vol. 1, 1, p. 119.

Cochran, W. T., et al, 1967, "What is the Fast Fourier Transform?," Proc. IEEE, Vol. 55, 10, p. 1664.

Comte-Bellot, G. and S. Corrsin, 1971, "Simple Eulerian time correlation of full- and narrow-band velocity signals in grid-generated 'isotropic' turbulence," J. Fluid Mech., Vol. 48, part 2, p. 273.

Deardorff, J. W., 1970a, "A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers," J. Fluid Mech., Vol. 41, p. 453.

Deardorff, J. W., 1970b, "A three-dimensional numerical investigation of the idealized planetary boundary layer," Geophys. Fluid Dyn., Vol. 1, p. 377.

Deardorff, J. W., 1971, "On the magnitude of the subgrid scale eddy coefficient," J. Comp. Phys., Vol. 7, p. 120.

Deardorff, J. W., 1973, "The use of subgrid transport equations in a three-dimensional model of atmospheric turbulence," J. Fluid Engineering, Sept. 1973, p. 429.

Fox, D. G. and J. W. Deardorff, 1972, "Computer method for solution of multidimensional, nonlinear, subsonic incompressible flow," J. Heat Transfer, Nov. 1972, p. 377.

Fox, D. G. and D. K. Lilly, 1972, "Numerical simulation of turbulent flows," Rev. Geophys. Space Phys., Vol. 10, 1, p. 51.

Grant, H. L. and I. C. T. Nisbet, 1957, "The inhomogeneity of grid turbulence, J. Fluid Mech., Vol. 2, p. 263.

Jain, P., 1967, "Numerical study of the Navier-Stokes equations in three dimensions," MRC Technical Summary Rep. No. 751, University of Wisconsin, Madison, Wisconsin.

Leonard, A., 1973, "On the energy cascade in large-eddy simulations of turbulent fluid flows," Report No. TF-1, Mechanical Engineering Department, Stanford University or Advances in Geophys., Vol. 18A. p. 237.

Leonard, A., 1975, "Numerical study of the energy loss of large eddies in turbulent flow simulations," to be published.

Lilly, D. K., 1964, "Numerical solutions for the shape preserving two-dimensional thermal convection element," J. Atmos. Sci., Vol. 21, p. 83.

Lilly, D. K., 1965, "On the computational stability of numerical solutions of time-dependent non-linear geophysical fluid dynamics problems," Mon. Weather Rev., Vol. 93, 1, p. 11.

Lilly, D. K., 1966, "On the application of the eddy viscosity concept in the inertial sub-range of turbulence," NCAR Manuscript No. 123.

Lilly, D. K., 1967, "The representation of small-scale turbulence in numerical simulation experiments," Proc. of the IBM Scientific Computing Symposium on Environmental Sciences, IBM Form No. 320-1951, p. 195.

Lumley, J. L. and B. Khajeh-Nouri, 1974, "Computational modeling of turbulent transport," Adv. in Geophysics, Vol. 18A, p. 169.

Orszag, S. A., 1969, "Numerical methods for the simulation of turbulence," Phys. Fluids, Sup. II, p. II250.

Orszag, S. A., 1971a, "Numerical simulation of incompressible flows within simple boundaries. I. Galerkin (spectral) representation," Studies in Applied Math., Vol. 1, 4, p. 293.

Orszag, S. A., 1971b, "Numerical simulation of incompressible flows within simple boundaries: accuracy," J. Fluid Mech., Vol. 49, p. 75.

Orszag, S. A. and M. Israelli, 1974, "Numerical simulation of viscous incompre-sible flows," Ann. Rev. Fluid Mech., Vol. 6, p. 281.

Phillips, N. A., 1959, The Atmosphere and Sea in Motion, Rockefeller Institute Press, New York, 501-504.

Reynolds, W. C., 1974a, "Recent advances in the computation of turbulent flows," Advances in Chem. Engrg., Vol. 9, (Reprint of Report MD-27, Mechanical Engineering Department, Stanford University.)

Reynolds, W. C., 1974b, "Computation of turbulent flows," AIAA 7th Fluid and Plasma Dynamics Conference, AIAA paper No. 74-556.

Reynolds, W. C., 1975, "Computation of turbulent flows," Report No. TF-4 Mechanical Engineering Department, Stanford University, or Annual Review of Fluid Mech., Vol. 8.

Rotta, J. C., 1951, Zeitschrift für Physik, I: 129, p. 547.

Schumann, U., 1973, "Ein Verfarhren zur direckten numerischen simulation turbulenter strömungen in platten- und ringspaltkanälen and über seine anwendung zur untersuchung von turbulenzmodellen," Univesität Karlsruhe. (NASA Technical Translation, NASA TT F 15,391.)

Schumann, U., 1974, "Subgrid scale model for finite difference simulation of turbulent flows in plane channels and annuli," in press.

Smagorinsky, J., 1963, "General circulation experiments with the primitive equations," Mo. Weather Rev., Vol. 93, 3, p. 99.

Smagorinsky, J., S. Manabe, and J. L. Holloway, 1965, Mon. Weather Rev., Vol. 93, p. 727.

Tennekes, H. and J. L. Lumley, 1972, A First Course in Turbulence, M.I.T. Press.

Tucker, H. J. and A. J. Reynolds, 1968, "The distortion of turbulence by irrotational plane strain," J. Fluid Mech., Vol. 32, part 4, p. 657.

Uberoi, M. S., 1963, "Energy transfer in isotropic turbulence," Phys. Fluids, Vol 6, 8, p. 1048.

Uzkan, T. and W. C. Reynolds, 1967, "A shear-free turbulent boundary layer," J. Fluid Mech., Vol. 28, part 4, p. 803.