JOURNAL OF COMPUTATIONAL PHYSICS 79, 373–416 (1988)

An Algorithm for Tracking Fluid Particles in Numerical Simulations of Homogeneous Turbulence

P. K. YEUNG* AND S. B. POPE

Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, New York 14853

Received March 6, 1987; revised October 26, 1987

Lagrangian statistical quantities are of fundamental physical importance in our understanding of turbulence, but are very difficult to measure and hence infrequently reported in the literature. A particle-tracking algorithm is developed to extract accurate Lagrangian statistics from numerically calculated velocity fields. Lagrangian time-series are obtained from the method of direct numerical simulation, which supplies the Eulerian velocity field on a three-dimensional grid network. The accuracy of the Lagrangian time series depends, primarily, on the accuracy of the interpolation scheme used to calculate fluid-particle velocities. Interpolation schemes based on Taylor series and on cubic splines have been implemented and tested. Errors in computed particle displacements are quantified for simple, frozen velocity fields. The algorithm is applied to stationary homogeneous isotropic turbulence with the energy maintained by artificial forcing. It is demonstrated that with adequate spatial resolution, accurate estimates of Lagrangian statistics such as velocity autocorrelations, structure functions, and frequency spectra can be obtained either with a third-order Taylor series interpolation scheme or with a cubic spline scheme. Cubic splines give higher interpolation accuracy, but they are more difficult to implement in codes that rely on secondary storage.

1. INTRODUCTION

Knowledge of Lagrangian statistical quantities, especially the velocity autocorrelation function, has long been recognised to be of fundamental theoretical importance in the understanding of turbulent diffusion and transport processes (see, for example, Taylor [1], Monin and Yaglom [2]). A Lagrangian description allows us to ask direct questions about the motion of material fluid elements. Unfortunately, Lagrangian statistics are extremely difficult to measure accurately: in an experiment, the instrumentation has to be able to follow the motion of fluid particles without disturbing the flow. By fluid particles we mean mathematical points moving with the local velocity of the fluid continuum (Batchelor [3]). There have been some useful attempts at measuring Lagrangian statistics by Snyder and Lumley [4], Shlien and Corrsin [5], and more recently by Sato and Yamamoto [6], but what they achieved is still limited.

A powerful alternative to experiment is to extract Lagrangian statistics from a

* Present address: Department of Mechanical Engineering, Queen's University at Kingston, Ontario, Canada.

0021-9991/88 $3.00

numerically computed velocity field. The fluctuating Eulerian velocity field must be available, which implies that mean flow closures like Reynolds stress models do not lend themselves to the task readily. Previous efforts have included those of Deardorff and Peskin [7] (who used sub-grid scale modeling) and of Riley and Patterson [8] (who used direct numerical simulations).

In this study we use direct numerical simulation (DNS) of turbulence, an approach pioneered by Orszag and Patterson's work [9] on isotropic turbulence. To meet spatial and temporal resolution requirements adequately (as discussed by Rogallo and Moin [10]), only moderate Ryenolds numbers are treated. We have been building on the code developed by Rogallo [11], who extended DNS to homogeneous turbulence. Many Eulerian statistical quantities can be formed from the instantaneous flow fields, including those like pressure–strain correlations which are inaccessible to measurement. This method is described further in Section 2.

The method of obtaining estimates of Lagrangian statistics is, in principle, straightforward. Given an Eulerian velocity field obtained by direct numerical simulation, we attempt to follow the time trajectories of a large number (M) of tagged fluid particles as the flow evolves. Let $\mathbf{x}^+(\mathbf{y}, t)$ and $\mathbf{u}^+(\mathbf{y}, t)$ denote the position and velocity at time t of the fluid particle originating at position $\mathbf{y}$ at time $t=0$. The equation of motion of the particle is

$$\frac{\partial \mathbf{x}^+(\mathbf{y}, t)}{\partial t} = \mathbf{u}^+(\mathbf{y}, t), \tag{1}$$

subject to the initial condition

$$\mathbf{x}^+(\mathbf{y}, 0) = \mathbf{y}. \tag{2}$$

The Lagrangian velocity $\mathbf{u}^+(\mathbf{y}, t)$ is related to the Eulerian velocity $\mathbf{u}(\mathbf{x}, t)$ by

$$\mathbf{u}^+(\mathbf{y}, t) = \mathbf{u}(\mathbf{x}^+(\mathbf{y}, t), t). \tag{3}$$

The superscript "+" signifies a Lagrangian flow variable. For notational simplicity from now on, whenever it is clear that a single particle is being referred to, we omit the initial position $\mathbf{y}$ and write $\mathbf{x}^+(\mathbf{y}, t)$ as $\mathbf{x}^+(t)$, etc.

Equation (1) is integrated numerically in time (for fixed $\mathbf{y}$). Equation (3) stipulates that the instantaneous particle velocity is the same as the fluid velocity at the instantaneous particle position. From DNS, the Eulerian velocity $\mathbf{u}$ is available on a three-dimensional grid network at each time step; the particle velocity can thus be calculated by interpolating for the fluid velocity at the particle position. An interpolation error is inevitably incurred in this process.

There are, however, non-trivial numerical difficulties to be overcome before we can have confidence in the accuracy of the results. It is known from the theory of material line stretching in turbulent flows (Batchelor [12]) that two neighboring fluid particles tend to move apart exponentially with time on the Kolmogorov time scale. This means numerical errors in particle positions can grow rapidly in time as the numerical integration proceeds.

Any numerical inaccuracies in the Eulerian velocity field supplied by DNS are clearly transmitted to the calculated particle trajectories. In this respect, having insufficient spatial resolution to resolve the higher wavenumber energy components is usually the main problem. However, we assume that the Eulerian calculation is of adequate accuracy and consider errors in particle displacements attributed to the following factors: (1) temporal resolution, or size of time step; and (2) interpolation error in obtaining particle velocities. These are all referred to as deterministic errors. Additionally, when ensemble averages over the set of M particles are used as approximations to the true Lagrangian statistics, statistical sampling errors resulting from having only a finite number of particles also need to be considered.

Preliminary studies showed that interpolation error is by far the most serious of these errors, with ordinary linear interpolation giving unacceptably poor results. This is not surprising when one notes that turbulent velocity fields are inherently highly nonlinear with appreciable higher wavenumber content in the energy spectrum. Thus we are led to spend considerable effort in developing different interpolating schemes and studying their accuracy.

Calculations of single-particle statistics have been reported by Riley and Patterson [8] for decaying isotropic turbulence. They used second-order-accurate linear interpolation for the particle velocity and did not provide direct estimates of the numerical errors. In the present study, we have developed and implemented a third-order-accurate interpolation scheme and have quantified errors in computed particle displacements, showing how they are affected by the various error sources cited above. The performance of cubic interpolating splines has also been evaluated.

A brief exposition of the principles of DNS is given in Section 2. The associated particle-tracking algorithm we developed is described in Section 3. In Section 4, we discuss the interpolation problem in some detail, examine how more accurate interpolation schemes can be constructed, and describe a means of quantifying interpolation errors.

Computational results are presented in Section 5. The performance of various interpolation schemes are compared by directly calculating the errors incurred by each of them on given turbulent velocity fields. The particle-tracking algorithm has been tested by making calculations for three different flows. These are, in order of complexity: (1) a simple steady flow (non-turbulent) with helical streamlines, for which particle paths are known analytically; (2) a frozen velocity field with an energy spectrum resembling that of turbulence; and (3) stationary homogeneous isotropic turbulence, in which viscous decay is compensated for by artificially forcing low wavenumber modes (Eswaran and Pope [13]).

Our results demonstrate that, with a reasonable amount of computational expense incurred on a powerful computer, accurate estimates of single-particle Lagrangian statistics can be obtained. These statistics include particle displacements, velocity autocorrelations, structure functions, and frequency spectra. The emphasis of this paper is on numerical aspects. In further work a yet wider range of statistics is studied (Yeung and Pope [31]).

2. The DNS Numerical Method

The basic principle of direct numerical simulation is to solve the Navier–Stokes equations numerically to determine instantaneous velocities in an incompressible turbulent flow. With this approach no modeling is needed. We are using Rogallo's [11] code for homogeneous turbulence. The account of the algorithm given below is brief; a more detailed exposition has been given by Lee and Reynolds [14].

An approximate solution to the Navier–Stokes equations is sought, in the form of velocity fields represented as discrete Fourier modes in wavenumber space. Periodic boundary conditions are required on the computational domain, which is a cube of length 2π. A pseudo-spectral method is employed, in which bilinear products (e.g., convective terms) are evaluated in physical space instead of using a convolution in wavenumber space. The aliasing errors thus incurred are removed by a combination of truncation and phase-shifting techniques (see [11]).

The transformed velocities in wavenumber space satisfy a set of coupled ordinary differential equations in time. These are solved by an explicit second-order-accurate Runge–Kutta (predictor–corrector) method. Let $\mathbf{k}$ be the wavenumber vector (of a node in wavenumber space), and let $\hat{\mathbf{u}}(\mathbf{k}, t)$ be the corresponding Fourier coefficient of velocity. For each $\mathbf{k}$, the velocity $\hat{\mathbf{u}}$ evolves by the equation

$$\frac{d\hat{\mathbf{u}}}{dt} = \hat{\mathbf{a}}[\hat{\mathbf{u}}(t)], \tag{4}$$

where $\hat{\mathbf{a}}$ represents the acceleration terms in wavenumber space and is dependent on $\hat{\mathbf{u}}$ at all $\mathbf{k}$. A time-stepping scheme advances the solution from time level t_n to $t_{n+1} = t_n + \Delta t$. In the predictor step we calculate $\hat{\mathbf{u}}^*$, a second-order-accurate approximation to $\hat{\mathbf{u}}(t_{n+1})$:

$$\hat{\mathbf{u}}^* = \hat{\mathbf{u}}(t_n) + \Delta t\, \hat{\mathbf{a}}[\hat{\mathbf{u}}(t_n)]. \tag{5}$$

In the corrector step, an improved approximation is formed by

$$\hat{\mathbf{u}}(t_{n+1}) = \hat{\mathbf{u}}(t_n) + \frac{\Delta t}{2}\,[\hat{\mathbf{a}}(\hat{\mathbf{u}}(t_n)) + \hat{\mathbf{a}}(\hat{\mathbf{u}}^*)]. \tag{6}$$

In the pseudo-spectral method, $\hat{\mathbf{a}}$ is obtained by forming the products in physical space. Thus the code swaps between physical space and wavenumber space twice for each of the two Runge–Kutta steps constituting one time step. On each time step the velocity fields $\mathbf{u}(\mathbf{x}, t_n)$, $\mathbf{u}^*(\mathbf{x})$, and $\mathbf{u}(\mathbf{x}, t_{n+1})$ are available successively in physical space ($\mathbf{u}^*(\mathbf{x})$ is the inverse transform of the predictor field $\hat{\mathbf{u}}^*(\mathbf{k})$).

The use of explicit time differencing generally entails a Courant number restriction on the largest time step size permissible (see, e.g., Roache [15]). For a three-dimensional problem the Courant number is defined as

$$C \equiv \Delta t \left[\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z}\right]_{\max}, \tag{7}$$

over all grid points in physical space. For finite-difference methods, numerical stability requires a Courant number not exceeding unity. For spectral methods, stability analysis is not so well established (see, e.g., Peyret and Taylor [16]), but it is in line with conventional wisdom to follow the same criterion. Tests conducted by Eswaran and Pope [13] show that using Courant numbers greater than unity in Rogallo's code can give rise to results grossly in error.

The DNS approach is limited to moderate Reynolds number because of resolution requirements. Briefly, we need to resolve the smallest scales and also to accommodate the largest scales occurring in the turbulence. It is well known that the range of scales widens rapidly as the Reynolds number increases (see, e.g., Tennekes and Lumley [17]).

Resolution of the small scales is an issue for the interpolation used for tracking fluid particles because these scales correspond to higher wavenumber components. For an N^3 calculation (with N grid points in each direction), the highest resolvable wavenumber, which we call $k_{\max}$, is the largest integer not exceeding $\sqrt{2}N/3$. With η denoting the Kolmogorov microscale, the results presented below suggest that the dimensionless combination $k_{\max}\eta$ is a fairly good measure of the spatial resolution, with $k_{\max}\eta \geqslant 1$ being the criterion for acceptably good resolution.

DNS has become viable largely because of the efficient Fast Fourier Transform (FFT) algorithm used for discrete Fourier transformations, as well as the increasing availability of modern powerful supercomputers. Even so, FFTs still remain the most time-consuming part of the code. In an N^3 simulation, the CPU time per time step increases as roughly $N^3 \ln N$. A 64^3 simulation takes about 30 s of CPU per time step on the FPS264 array processors at the Cornell National Supercomputing Facility. On the IBM 3090-600 at the same facility, the code takes only about 80 CPU s per step for a 128^3 simulation. Additional CPU time spent on particle tracking is typically only a small fraction of the time taken by the Eulerian code.

Data Structure and Memory Considerations

The original version of Rogallo's code is such that only one plane of data is available in memory at a time. This structuring circumvents memory limitation difficulties associated with the extremely large number of nodal points on fine grids, but it is at the cost of considerable inconvenience. In the course of our study, we have developed an algebraically equivalent version of the code in which entire 3D arrays are in (virtual) memory. This is feasible when sufficient memory can be allocated, as on the IBM 3090 available to us, where up to 999 megabytes of virtual memory can be accessed.

We refer to these two versions as "planewise" and "full-cube" codes, respectively. The choice has a bearing on the consideration of interpolation schemes for particle tracking discussed later.

3. Particle-Tracking Algorithm

The basic principle involved in following fluid particle trajectories has been set forth in the Introduction. Here we discuss how the task is performed within the framework of the Eulerian DNS.

It is convenient to integrate the equation of motion (1) numerically by the same time-differencing method used in Rogallo's code, i.e., a second-order Runge–Kutta scheme. Recall that $\mathbf{x}^+(t)$ and $\mathbf{u}^+(t)$ denote the position and velocity of a fluid particle at time t. Starting at time t_n, the predictor step yields an estimate $\mathbf{x}^*$ of $\mathbf{x}^+(t_{n+1})$:

$$\mathbf{x}^* = \mathbf{x}^+(t_n) + \Delta t\, \mathbf{u}^+(t_n). \tag{8}$$

In terms of the Eulerian velocity field $\mathbf{u}(\mathbf{x}, t_n)$, according to Eq. (3) the particle velocity is

$$\mathbf{u}^+(t_n) = \mathbf{u}(\mathbf{x}^+(t_n), t_n). \tag{9}$$

From the predictor position $\mathbf{x}^*$ and the predictor Eulerian velocity field $\mathbf{u}^*(\mathbf{x})$ (which approximates $\mathbf{u}(\mathbf{x}, t_{n+1})$), the corrector step calculation is

$$\mathbf{x}^+(t_{n+1}) = \mathbf{x}^+(t_n) + \frac{\Delta t}{2}\,[\mathbf{u}^+(t_n) + \mathbf{u}^*(\mathbf{x}^*)]. \tag{10}$$

We have used $\mathbf{u}^*(\mathbf{x}^*)$ because of the limitations of the "planewise" code; but, if the "full-cube" code is used, $\mathbf{u}^+(\mathbf{x}^*, t_{n+1})$ can be used in place of $\mathbf{u}^*(\mathbf{x}^*)$ to give a better approximation.

Each particle is tagged and assigned an initial position. Its path is then traced by continually updating its position by the time-advancement scheme, using particle velocities interpolated on the three-dimensional Eulerian grid network. Thus over each time step, there are two sources of numerical error: the time-stepping error and the interpolation error involved in evaluating particle velocities. The time-stepping error is of order $(\Delta t)^3$ over one time step (see, e.g., [18]), but of order $(\Delta t)^2$ when a time interval of fixed length is considered, since the number of time steps is inversely proportional to Δt.

To extract Lagrangian statistics, a set of M particles is introduced into the flow, with initial positions specified according to a uniform statistical distribution. In working with particle statistics we are implicitly concerned with the transitional probability distribution of stepwise particle displacements, conditioned on the particle position and the Eulerian velocity field at the current time level (see [2] for a discussion). Spatial homogeneity is crucial because it removes the probabilistic dependence on current (or initial) position, enabling us to treat each particle as representing a different realisation of the underlying Lagrangian stochastic process. This justifies taking ensemble averages over all the M particles. Appeal to the central limit theorem shows that the statistical sampling error decreases as $M^{-1/2}$.

Thus for 1 % accuracy, say, we require of order 10^4 particles, which is much less than the number of grid points ($N^3 \approx 2.6 \times 10^5$, for $N=64$). Since M is much smaller than $N^3 \ln N$, considerable computational effort per particle can be expended to track the trajectories accurately, without significantly increasing the total computational requirement. How large M needs to be is judged by monitoring the statistical convergence of the overall statistics.

In using Rogallo's code for the Eulerian DNS, we need to take into account the data structure and the periodic boundary conditions on the velocity field. However, we stress that the particle-tracking algorithm is general. It depends solely on the velocity fields being available in physical space.

Each simulation yields a set of Lagrangian time series from which Lagrangian statistics such as velocity autocorrelations are estimated. Some questions on this data analysis are addressed in Appendix A.

4. Interpolation Schemes

Accurate interpolation in the context of particle tracking is a difficult task, since the Eulerian velocity fields vary greatly in space in a nonlinear manner, with infinitely many higher order derivatives present. As stated in the Introduction, numerical errors in particle tracking grow rapidly in time, mainly as a result of the diffusive nature of turbulence. It is thus imperative to have an accurate interpolation scheme. A close look at the interpolation problem is needed to devise more sophisticated schemes of higher accuracy.

Spectral (Exact) Interpolation

Let us first point out why in practice particle velocities cannot be calculated exactly. Unlike finite difference methods which yield only the nodal values on a grid, spectral methods do provide a complete description of the dependent variable throughout the solution domain. Consider a finite one-dimensional Fourier series defining a function f on an interval $[0, L]$ on the real line, divided into N subintervals of length Δx, such that $x_n = n\,\Delta x$ and $x_N = L$:

$$f(j\,\Delta x) \equiv f_j = \Delta x \sum_{n=0}^{N-1} \hat{f}_n e^{2\pi i n j/N}, \qquad j=0, 1, ..., N-1. \tag{11}$$

The sequences $\{f_j, j=0, 1, ..., N-1\}$ and $\{\hat{f}_n, n=0, 1, ..., N-1\}$ form a discrete Fourier transform pair (see, e.g., [19]). With the Fourier coefficients being specified, this formula defines a continuous function of x in the interval $[0, L]$,

$$f[(j+\alpha)\,\Delta x] = \Delta x \sum_{n=0}^{N-1} (\hat{f}_n e^{2\pi i n\alpha/N})\, e^{2\pi i n j/N}, \tag{12}$$

where $0 \leqslant \alpha \leqslant 1$. The quantities in brackets on the right-hand side of Eq. (12), $\hat{f}_n(\alpha)$, say, are the Fourier coefficients of $f(x)$ on the grid shifted by an amount $\alpha\,\Delta x$, and

are equal to the coefficients $\hat{f}_n$ shifted by a phase $n\alpha/N$. Thus the value of the function f at any location within the interval may be computed exactly by applying a phase shift in wavenumber space before transforming to physical space. This procedure merits the term "exact" interpolation only when the function f is defined (or can be represented) by the finite Fourier series: as is the case in the tests described below.

Unfortunately, this technique of spectral interpolation requires a full set of three-dimensional FFTs in order to obtain just one velocity component of a single fluid particle per Runge–Kutta step. The computational cost involved, which increases as $MN^3 \ln N$, is obviously unacceptably large. Still, spectral interpolation does provide a valuable means of obtaining exact values needed for a direct evaluation of the interpolation error and forms the basis of an algorithm to test and compare interpolation errors of different schemes on a typical velocity field, as described in Section 5.1.

General Considerations for Interpolation Schemes

Consider a particle located within a cubic interpolation cell of unit grid spacing, as sketched in Fig. 1. Let $\mathbf{x}^{(i)}$ be the local coordinates of node i, relative to the center of the cell (i.e., node 9) and normalised by the grid spacing Δx. Thus, for example, node 9 and node 1 have coordinates $(0, 0, 0)$ and $(-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2})$, respectively. The function value at node i is denoted by $f_i = f(\mathbf{x}^{(i)})$. In a p-point scheme, the values at p nodal points are used. Let them be numbered from 1 to p, and $a_i(\mathbf{x})$ be the interpolation weight at node i. Then an approximating function $g(\mathbf{x})$ is defined in the form of a linear combination of nodal values:

$$g(\mathbf{x}) = \sum_{i=1}^{p} a_i(\mathbf{x})\, f(\mathbf{x}^{(i)}). \tag{13}$$

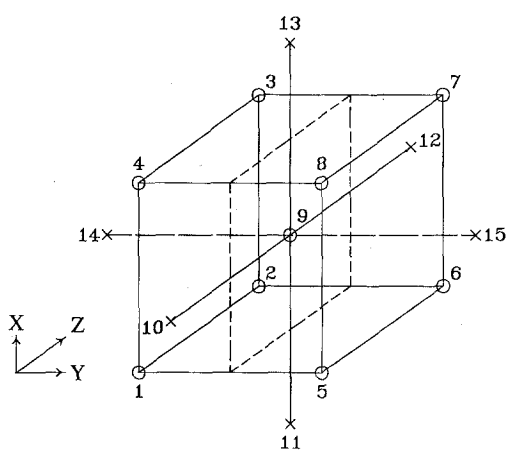

FIG. 1. Sketch of a typical interpolation cell showing the location of nodal points. Coordinate axes have origin at node 9, with directions as indicated.

We first review the general properties of interpolation schemes and then consider several specific choices. It is usual to require the interpolation weights $\{a_i\}$ to satisfy a collocation condition at nodes lying in the cell or on its boundary and, also, to meet a consistency constraint by having their sum equal to unity (consider the case of a uniform field). Another desirable property is for $g(\mathbf{x})$ to be continuous across the interfaces of neighboring interpolation cells. This property, called C^0 continuity in finite-element analysis terminology, ensures that calculated particle velocities change smoothly, without abrupt jumps, as a particle crosses an interface. It may also be desirable for some derivatives of $g(\mathbf{x})$ to be continuous across cell boundaries as well: if the nth derivative of $g(\mathbf{x})$ is continuous, it is said to be C^n continuous (see, e.g., [20]).

Different schemes differ in the choice of nodes and interpolation weights. Several schemes are discussed below. Apart from splines, all of them use some or all of the fifteen nodes shown in Fig. 1. Nodes 1 to 8 correspond to the nodes present in the Eulerian DNS; the role of the others (9 to 15) are stated in the paragraphs below. Cubic splines use the basis function coefficients at the 64 nodes that form a cube surrounding $\mathbf{x}$, but these coefficients depend on all of the N^3 nodal values through the solution of matrix equations described later.

Linear Scheme

Linear interpolation in three directions has been widely used [7, 8] because of its simplicity. This linear scheme involves only the nodal values at the 8 vertices of the interpolation cell (nodes 1 to 8). It approximates the dependent variable within the cell as a linear function in each coordinate. Since such a function has 8 coefficients, matching the number of nodal points used, enforcing the collocation condition suffices to determine the interpolation weights uniquely. The consistency constraint is also satisfied.

Nonlinear variations on scales smaller than one grid spacing are completely ignored. This scheme is well known to be second-order accurate, i.e., the magnitude of the interpolation error decreases asymptotically as $(\Delta x)^2$ as the grid spacing Δx tends to zero. It is C^0 continuous because of the piecewise linear nature of the approximating function.

A Third-Order Taylor Series 13-Point Scheme

In the search for better interpolation schemes it is natural to start by examining the use of Taylor series expansions. In view of the weaknesses of ordinary linear interpolation, it is intuitively clear that we can get better results by (1) improving spatial resolution through adding one or more nodes inside the cell so that the distance between a particle and the nearest node is reduced on the average; (2) including nodes outside the cell so that higher order derivatives may be expressed more accurately.

Both of these objectives are achieved by adding an extra staggered grid of data shifted from the regular grid in Rogallo's code by half a grid spacing in each direction. Nodes 9 to 15 in Fig. 1 are part of the staggered grid, with the nodal

values obtained by phase-shifting in wavenumber space and then transforming to physical space.

A scheme of third-order accuracy is desired. By manipulating the Taylor series expansions of $f_1 \cdots f_{11}$ about node 9, the three first and six second partial derivatives can be expressed in terms of these nodal values. The same is done for the simplest mixed third derivative, $\partial^3 f/\partial x\, \partial y\, \partial z$, which is first order in each coordinate. In addition, the interpolation weights are to satisfy the consistency condition. We then have a set of 11 equations or constraints for interpolation weights at 11 nodes and can solve for those weights uniquely.

The 11-point scheme using nodes 10 and 11 (for the second derivatives in x and z) is asymmetric. The same is true if nodes 12 and 13 are used instead. This unwanted asymmetry is removed by taking the averages of the interpolation weights resulting from these two choices, to achieve a 13-point scheme which is symmetric in x and z (but not y). This scheme, which we refer to as the "TS13" scheme, is found to be very successful in the numerical tests conducted. The complete formula is given in Appendix B.

A careful examination of the formula reveals that the resulting approximating function does not possess the C^0 continuity property. The approximation $g(\mathbf{x})$ thus has jumps as the location $\mathbf{x}$ passes across interpolation cell boundaries. Such jumps contribute partly to numerical noise effects affecting the estimation of frequency spectra, but are of significant impact only when a very (unrealistically) small Courant number of order $\frac{1}{10}$ or less is used in a poorly resolved velocity field. Since an effective corrective procedure has been developed (see Appendix A), we consider this a tolerable drawback.

In principle we could have also included nodes 14 and 15 to yield a completely symmetric scheme, but this would be very cumbersome to implement in the "planewise" version of Rogallo's code since the solution domain is spanned by marching through successive y-planes. Moreover, numerical tests showed little additional benefit.

Finite-Element Interpolation Functions

A related approach parallels the use of interpolation functions in finite-element methods. The cubical cell can be divided into 6 congruent pyramids, each having its four vertices drawn from only the nodes 1 to 8. Within each such pyramid, the consistency condition and the three first derivatives can be used to determine the four interpolation weights. This is referred to as the "TS8" scheme. It is less accurate than the usual linear scheme because the average internodal distance is effectively increased. The resolution can be improved by incorporating node 9. The resulting "TS9" scheme divides the cube into 12 pyramids of smaller size.

Since both schemes are piecewise linear in nature, continuity of the interpolated function across the boundaries between adjacent pyramids is guaranteed. Piecewise linearity also implies second-order accuracy for both schemes. The details of each are given in Appendix B.

Optimal Schemes

The rationale behind Taylor series schemes concerns the rate at which interpolation error decreases as the grid size is refined. In an mth-order scheme, the error decreases as $(\Delta x)^m$ as the grid spacing Δx tends to zero. However, with Δx being of the same order as the smallest length scale of variation of the function (the Kolmogorov microscale η), there is no assurance that high-order Taylor series schemes will perform well. The weights in such schemes depend on spatial position only and take no account of the nature of spatial variations of the field variable.

In a different approach, we seek to determine an interpolation scheme that is optimal (i.e., giving least error) for a given velocity field. The interpolation weights and the resulting error then depend on the statistical characteristics of the velocity field, which are well represented by the energy spectrum. Using the notation of Eq. (13), the squared interpolation error is given by

$$\varepsilon(\mathbf{x}) = [g(\mathbf{x}) - f(\mathbf{x})]^2. \tag{14}$$

For a specified function $f(\mathbf{x})$ and a selection of p nodes, by solving a symmetric p^2 linear system of equations, we can determine the weights $\{a_i\}$ such that the squared error averaged over all N^3 cells of the whole computational box is minimum. The collocation property holds automatically since when $\mathbf{x}$ is the position of a nodal point, the smallest error is clearly zero and it occurs with $g(\mathbf{x})$ equalling $f(\mathbf{x})$ at the nodal point. The consistency condition has to be explicitly imposed. Algebraic details are given in Appendix B.

Three factors limit the usefulness of this class of "optimal" schemes. First, the coefficients are expensive to generate, since the exact values $[f(\mathbf{x})]$ have to be calculated via spectral interpolation. Second, when the statistics of the velocity field change appreciably, the same set of coefficients may no longer be close to optimal. Third, the p^2 symmetric matrix involved is usually found to be quite ill-conditioned, especially for large p and N, requiring an inordinate amount of effort to generate the interpolation weights successfully.

Cubic Splines

The "TS13" scheme was developed with the practical restrictions of the "planewise" code in mind. However, with the "full-cube" version we developed, the use of interpolating splines, which compare favorably with other schemes, is greatly facilitated.

Cubic splines provide C^2 continuous approximations. For the one-dimensional case, the spline approximation $g(x)$ is

$$g(x) = \sum_{i=0}^{N_b - 1} b_i(x)\, e_i, \tag{15}$$

where $b_i(x)$ is the ith basis function, e_i is the corresponding coefficient, and the sum is taken over N_b basis functions [21]. Each basis function is constructed as a

piecewise cubic that is twice continuously differentiable. At any location x, only four basis functions are non-zero. Algebraic expressions are given in Appendix B.

The fields being considered (e.g., $\mathbf{u}(\mathbf{x}, t)$) are periodic, and hence the basis functions and their coefficients can also be considered to be continued periodically outside the interval considered. There being N equispaced nodes, we choose $N_b = N+3$ basis functions centered at the nodes and their periodic extensions. Given the values of the periodic function f at the N nodes, the basis function coefficients can be determined from the N collocation conditions. This leads to an N^2 symmetric periodic tridiagonal system of equations for the basis functions

$$\sum_{j=1}^{N} T_{ij} e_j = f_i \qquad (i = 1, ..., N), \tag{16}$$

where

$$T_{ij} = \begin{cases} \frac{2}{3} & i = j \\ \frac{1}{6} & |i-j| = 1 \\ \frac{1}{6} & (i, j) = (1, N) \text{ or } (N, 1) \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

and $\{f_i\}$ are the nodal values. An efficient procedure for solving such a system is described in [22].

A three-dimensional tensor product spline can be formed as [21]

$$g(x, y, z) = \sum_{k=0}^{N_b-1} \sum_{j=0}^{N_b-1} \sum_{i=0}^{N_b-1} b_i(x)\, c_j(y)\, d_k(z)\, e_{ijk}, \tag{18a}$$

where $\{b_i\}$, $\{c_j\}$, and $\{d_k\}$ are the one-dimensional basis functions, and e_{ijk} are the basis function coefficients. At the collocation points (i.e., grid nodes) denoted by x_I, y_J, $z_K (I, J, K = 1, ..., N)$, Eq. (18a) can be written as

$$g(x_I, y_J, z_K) = \sum_{i=0}^{N_b-1} b_i(x_I)\, E_{iJK}, \tag{18b}$$

where

$$E_{iJK} = \sum_{j=0}^{N_b-1} c_j(y_J)\, F_{iJK}, \tag{18c}$$

and

$$F_{iJK} = \sum_{k=0}^{N_b-1} d_k(z_K)\, e_{ijk}. \tag{18d}$$

From these equations it is readily seen that the N_b^3 basis function coefficients e_{ijk} can be obtained by forming a sequence of $3N_b^2$ one-dimensional splines. Approximately $21N^3$ floating point operations are required in the algorithm used. The time spent increase as N^3.

Besides having desirable continuity properties, the splines are formally fourth-order accurate. First and second derivatives of the function f (considered below) can be approximated either by differentiating the spline or fitting a spline to the derivatives. These two methods are denoted by CSA and CSB, respectively. The CSB method is more accurate, but also more time-consuming since the nodal values of the derivatives have to be generated (by FFTs).

5. Results and Discussion

5.1. *Direct Measurement of Interpolation Errors*

A precise comparison of the performances of the various interpolation schemes considered can be made through computing the error incurred by each scheme in interpolating a typical velocity field. For a prescribed velocity field we can obtain exact values by spectral interpolation and then calculate the error averaged over a network of prescribed locations within the cubical cell. We subdivide each of the N^3 interpolation cells into $8 \times 8 \times 8$ sub-cells, compute the averaged squared error over all cells, and take its root-mean-square (rms) over these sub-cells to get a global estimate of the interpolation accuracy. A measure of local spatial variation within an interpolation cell is given by the difference between the largest and the smallest of the 9 nodal values occurring in it. The rms interpolation errors are normalised with respect to the rms value of this difference over all cells. As a measure of spatial variation over the interval Δx, this normalising quantity scales asymptotically as Δx gets sufficiently small.

For the choice of velocity field, we want one having an energy spectrum resembling (at least roughly) the spectra occurring in turbulence. A convenient spectral form is that proposed by Pao [23], which is implemented as

$$E(k) = \begin{cases} k^{-5/3} \exp[-1.5\alpha(k\eta)^{4/3}], & (\alpha = 2.45),\ 0 < k < k_{\max} \\ 0, & k \geqslant k_{\max}, \end{cases} \tag{19}$$

where k is the magnitude of the wavenumber vector $\mathbf{k}$ and $E(k)$ is the energy spectrum function. Multiplicative constants have been omitted since we are only interested in relative magnitudes. For use in Rogallo's code, the spectrum has to be truncated at $k = k_{\max}$. The specification of $E(k)$ is made complete through the selection of η, or, for given N (hence known $k_{\max}$), through the dimensionless parameter $k_{\max}\eta$.

A velocity field having the given energy spectrum is constructed in wavenumber space, using the initialisation mechanism of Rogallo's code [11]. The amplitudes of $\hat{\mathbf{u}}(\mathbf{k})$ are set deterministically to conform with Eq. (19), while the directions of $\hat{\mathbf{u}}$ in wavenumber space are specified randomly in the plane normal to $\mathbf{k}$ so that the

continuity equation $\mathbf{k} \cdot \hat{\mathbf{u}} = 0$ is satisfied. Although a given $E(k)$ does not specify the velocity field uniquely in the deterministic sense, the shape of the spectrum is a useful indicator of the degree of difficulty of interpolation, which is generally considerable if there is appreciable high wavenumber content.

Ten interpolation schemes are under consideration, namely, "TS13," "TS8," "TS9," "linear," splines, and "OPT15," "OPT13," "OPT11,", "OPT9," "OPT8" (the last five are "optimal" schemes with the specified number of nodes). A significant nondimensional group is $k_{\max}\eta$, which is proportional to the ratio between the smallest scale of variation of the velocity field (η) and the smallest scale resolvable by the grid ($1/k_{\max}$). Normalised rms interpolation errors were calculated for several choices of $k_{\max}\eta$, and are plotted in Fig. 2 for $N = 32$. For all schemes, errors decrease rapidly (especially for "TS13" and splines) as $k_{\max}\eta$ increases beyond 1. This suggests the use of the condition $k_{\max}\eta \geqslant 1$ as a suitable criterion for "good" spatial resolution, meaning that (loosely speaking) when $k_{\max}\eta$ exceeds 1, adequate accuracy may be achieved with the use of high-order schemes like "TS13" or splines at our disposal. Splines are found to perform best, while "TS13" also gives small errors. Close similarity is found for $N = 16$, thus supporting the choice of $k_{\max}\eta$ as a characteristic parameter.

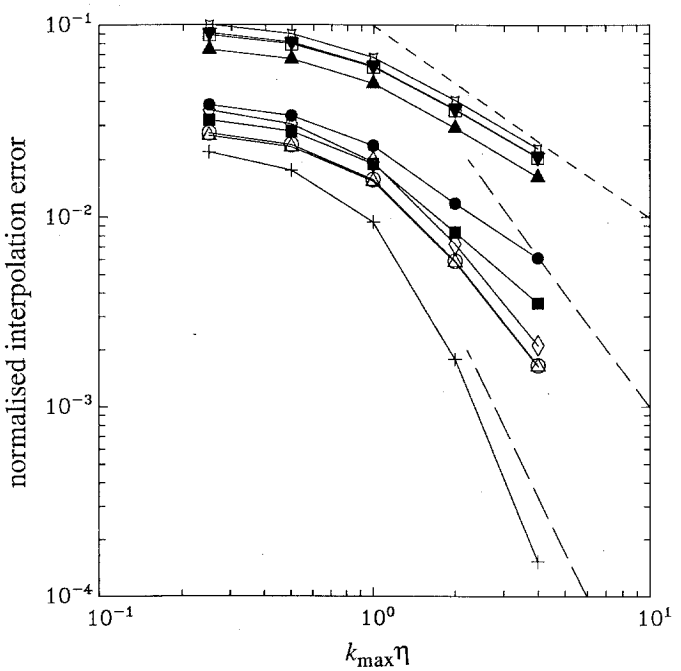

FIG. 2. Normalised interpolation error on velocity field specified by Pao's spectrum, against $k_{\max}\eta$, using a 32^3 grid. Interpolation schemes are identified by symbols: △ OPT15; ○ OPT13; ■ OPT11; ● OPT9; □ OPT8; ◇ TS13; ▲ TS9; ⌑ TS8; ▼ linear; + splines; ---; —; ——: lines of slopes −1, −2, −3, respectively.

For an mth order scheme, as Δx tends to zero, the unnormalised error decreases as Δx^m. But since the normalising factor used scales as Δx, the normalised error is expected to scale as Δx^{m-1}. In Fig. 2, slopes for schemes of various orders appear to approach the correct asymptotic behaviour as $k_{\max}\eta$ increases beyond 2.

The data of Fig. 2 appear as percentages in Table I. The errors of the optimum schemes decrease with the number of nodes, since, for $p_1 < p_2$, all p_1-point schemes are in fact members of the wider class of p_2-point schemes (with the extra nodes taking on zero interpolation weights). Evidently, addition of node 9 alone contributes to a substantial error reduction. It turns out that "OPT15" is only marginally better than "OPT13," justifying the omission of nodes 14 and 15 of Fig. 1 in our development of Taylor series schemes. Moreover, the "TS13" scheme is seen to perform remarkably well, its error at $k_{\max}\eta = 1$, being less than 2 % and only about 25 % larger than that of "OPT15." The splines consistently have the best performance, their error being 40 % less than "OPT15" at $k_{\max}\eta = 1$.

Figure 3 shows interpolation errors for the first derivative. For splines, CSA is less accurate than CSB, as expected. However, the difference in accuracy is small, and given its numerical efficiency, CSA still appears to be the preferred option.

We conclude that the addition of the extra staggered grid of data giving more nodal points has achieved a large improvement over the linear interpolation scheme.

5.2. *Particle Displacement Errors in Steady Helical Flow*

This non-turbulent flow provides a simple test case in which errors in particle displacements can be precisely quantified by comparing with an easily derived analytical solution.

In this flow, particles lying within the cylinder with axis in the y-direction and inscribed by the computational box move in axisymmetric circular motion in the

TABLE I

Normalised Interpolation Errors (%) for Different Values of $k_{\max}\eta$ for Pao's Spectrum on 32^3 Grid

Scheme	4.0	2.0	1.0	0.5	0.25
OPT15	0.1642	0.5847	1.547	2.330	2.670
OPT13	0.1661	0.5925	1.577	2.391	2.753
OPT11	0.3552	0.8385	1.908	2.810	3.221
OPT9	0.6143	1.186	2.370	3.383	3.857
OPT8	2.060	3.638	6.083	8.014	8.932
TS13	0.2124	0.7284	1.948	3.064	3.620
TS9	1.631	2.929	4.990	6.687	7.508
TS8	2.295	4.082	6.822	9.044	10.11
Linear	2.071	3.658	6.149	8.176	9.153
Splines	0.0154	0.1788	0.9472	1.762	2.194

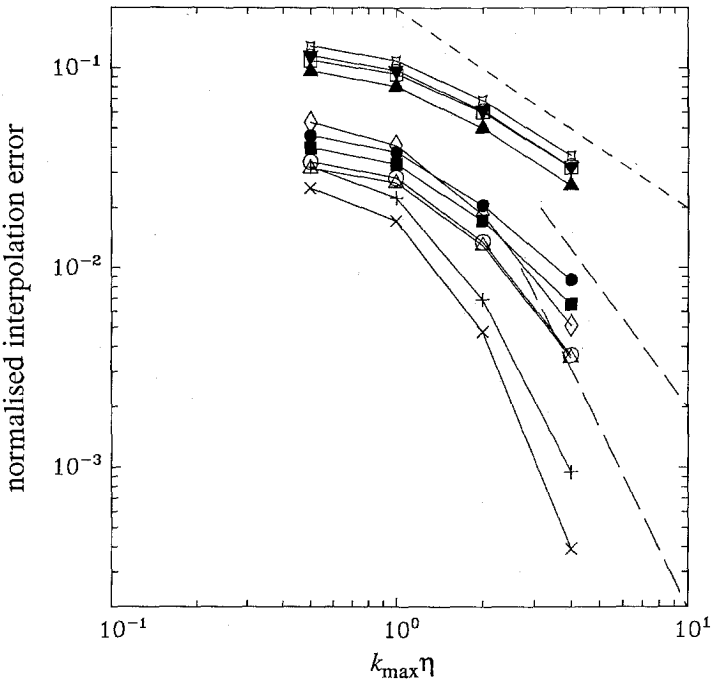

FIG. 3. Same as Fig. 2, but for velocity gradient: + splines (CSA); × splines (CSB).

x–z plane, and in uniform translational motion in the y direction. A given particle moves on a cylinder of radius r, with angular velocity ω depending only on r, measured from the axis of the cylinder. Let $\theta^+(t)$ be the angular displacement at time t, then the equations of motion are simply

$$\frac{d\theta^+}{dt} = \omega(r), \tag{20a}$$

$$\frac{dy^+}{dt} = v^+ = \text{const.} \tag{20b}$$

(The value of v^+ is picked to give a desired time step size based on the Courant number limit.) These equations are readily integrated to give the particle position in Cartesian coordinates.

The angular velocity $\omega(r)$ is chosen to have a smooth profile,

$$\omega(r) = \begin{cases} 1 - 3\left(\frac{r}{\pi}\right)^2 + 3\left(\frac{r}{\pi}\right)^4 - \left(\frac{r}{\pi}\right)^6, & \text{for } r \leqslant \pi, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Note that the inscribed cylinder is of radius π. Although circular motion in a plane is not periodic, the cutoff at $r = \pi$ (where $\omega(r)$ and its first two derivatives are zero)

ensures compatibility with periodic boundary conditions in the DNS algorithm. The Eulerian velocity field is frozen in time, i.e., steady.

In a series of numerical tests, 400 particles are tracked for a time interval of approximately 2π, which is the time taken by a particle lying midway between the perimeter and the axis of symmetry to complete one revolution. This is deemed sufficient to bring out all evolutionary features of the particle motion, and indeed

Our goal is to isolate and characterise the effects of three factors on particle displacement errors: (1) choice of interpolation scheme; (2) size of time step; and (3) grid spacing. All the plots presented show root-mean-square errors in displacements normalised by the root-mean-square displacements from their initial positions of all particles up to the corresponding time.

Error contributions from different factors add up to give the overall error. Thus, for example, to isolate the magnitudes of interpolation errors, it is necessary to find a time step small enough so that time-differencing errors are not large enough to obscure or distort the conclusions. For a frozen velocity field, a fixed time step is equivalent to a fixed Courant number.

Figure 4a shows the error growth at a Courant number of $\frac{1}{2}$, and $N=16$, for various interpolation schemes. The errors shown are averages over the x and z directions. Although absolute errors increase with time (roughly linearly), normalised errors do not show appreciable increases until about half of the whole tracking time period has elapsed. By that time the root-mean-square displacement curves (not shown) have begun to level off, partly contributing to the increase in normalised errors.

Except for the "OPT15" scheme which we decided not to implement in the code, all schemes considered in Section 5.1 appear in Fig. 4a. Splines are seen to give the least errors among all the schemes considered, while the "TS13" scheme gives errors as much as 10 times less than those of the simple linear scheme and is only slightly inferior to the "OPT13" scheme. All 8-point schemes are distinctly worse than the others, and are not very different among themselves. This means little can be achieved without using more nodal points. The big difference between "OPT9" and "OPT8" again shows that the presence of additional nodal points is very valuable. These observations are all consistent with the results reported in Section 5.1.

Besides tracking the fluid particles directly, as we have done, another method, not mentioned so far, is provided, in principle, by using non-diffusive scalars. We call this approach tracking by non-diffusive scalars and describe it further in Appendix C. Figure 4b shows the errors in particle displacements calculated by this method, for the same conditions as in Fig. 4a. A disastrous rate of error growth is manifest. At larger times, the Eulerian scalar field itself is so inaccurate that any advantages of higher order interpolation schemes are lost. This method has been rejected.

More detailed analysis of the results (see [24]) shows that the time-stepping error is generally much less than the interpolation errors, especially for coarser

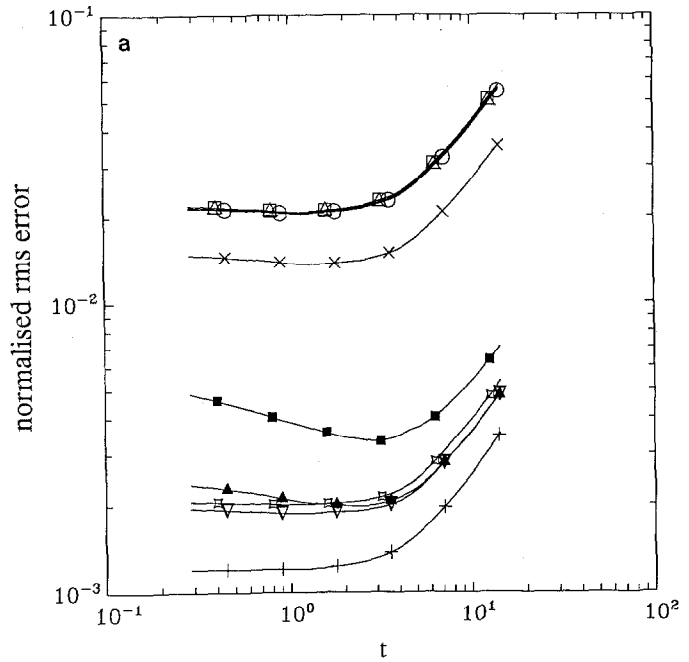

FIG. 4a. Helical flow: normalised rms error in particle displacement against time, for various interpolation schemes. Results are obtained by directly following particle trajectories: △ linear; ▲ TS13; □ TS8; × TS9; ⧈ OPT13; ▽ OPT11; ■ OPT9; ○ OPT8; + splines.

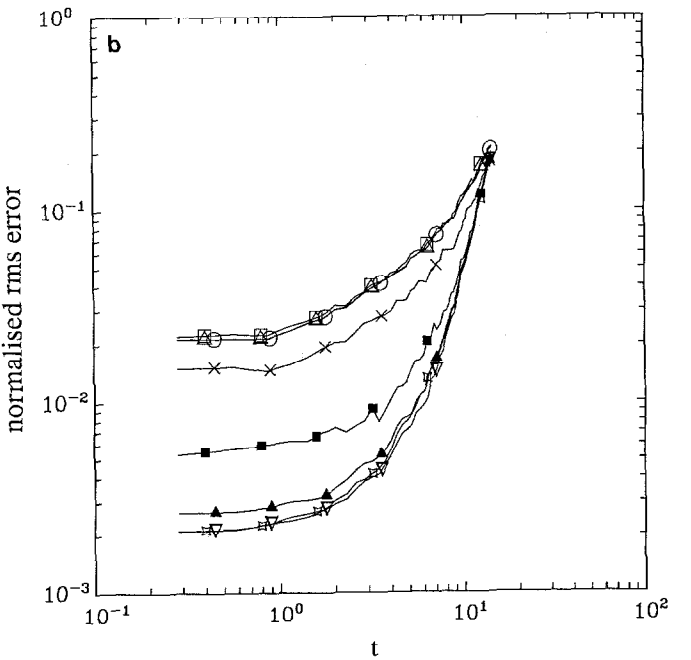

FIG. 4b. Same as Fig. 4a, but for a method of particle tracking by non-diffusive scalars.

grids and less accurate schemes. Data obtained on 8^3, 16^3, and 32^3 grids also provide ample evidence of the claimed third-order accuracy of the "TS13" scheme.

This simple helical flow, though not representative of turbulence, has nevertheless provided much useful information on the performance of various schemes. In the next subsection a velocity field bearing more resemblance to turbulence is considered.

5.3. *Frozen Velocity Field with Typical Energy Spectrum*

We need to assess the performance of the various interpolation schemes under consideration when applied to the highly nonlinear turbulent velocity fields we are ultimately interested in. Useful insight may be gained from working with an energy spectrum at least qualitatively representative of turbulence. Numerical tests are again conducted using the spectral form due to Pao [23], as implemented in Section 5.1.

The velocity field is frozen in time. We study the particle displacement errors first and then discuss the particle statistics obtained from the simulations.

In this case, as for almost all turbulent flows, a closed-form solution for the velocity field is not available. Exact particle paths are thus unknown. The best substitute we have is a calculation employing a very accurate interpolation scheme ("TS13" or splines), a very small time step ($C = \frac{1}{8}$), and a very fine grid spacing ($N = 256$). Other calculated particle trajectories are then compared against this "best" calculation. It is important that the "best" calculation be extremely accurate, for otherwise the error comparisons may be distorted. For 128^3 grids it is found to be necessary to use splines on 256^3 as the "best" calculation for a more accurate comparison.

As discussed before, $k_{\max}\eta \geqslant 1$ is considered to yield adequate resolution of the small scales. Our base case is one realisation of frozen homogeneous isotropic turbulence having the energy spectrum of Eq. (19), on a 32^3 grid with $k_{\max}\eta = 1$. The time step and grid size are systematically varied to study their effects on particle displacement errors. Courant numbers used were between 1 and $\frac{1}{8}$.

A means of exact grid refinement for our spectral method is needed. Referring to Eq. (19), our energy spectrum is nonzero only up to $k = 15$ (the value of $k_{\max}$ for $N = 32$). The same velocity field is represented, but on a finer grid, if we extend the wavenumber range at the high end, but use the same $E(k)$ up to $k = 15$, and specify $E(k) = 0$ for $k > 15$. This method is used to generate the flow fields for 64^3, 128^3, and 256^3 simulations, which have $k_{\max}\eta = 2$, 4, and 8, respectively. In general, in expanding to a finer grid, identical wavenumber spectra (and therefore the same functional representation in physical space) are achieved by filling the extra higher wavenumber modes present in the finer grid with zeroes.

The results of the previous subsections suggest that "TS13" and splines are both very accurate interpolation schemes. Numerical tests for this subsection were therefore conducted for four schemes: "TS13," splines, "linear," and "OPT13." The linear scheme serves as a benchmark reference indicating the magnitude of the improvement achieved in other schemes, while "OPT13" is used to provide a measure of

how much more accuracy is possible using only the 13 points in "TS13." Unfortunately, for $N=64$ and $N=128$, the system of equations used to generate the optimal coefficients was found to be ill-conditioned, making a satisfactory solution unfeasible. Instead we use the optimal coefficients from a 32^3 grid, but with the value of $k_{\max}\eta$ appropriate to the finer grid.

Particle Displacement Errors

Particle displacement errors are averaged over the three coordinate directions to reduce statistical sampling fluctuations arising from using only one flow realisation in each simulation. In Fig. 5, such averaged errors, normalised with respect to the rms particle displacement up to the corresponding times, are plotted against time for the three schemes mentioned above, and for $N=32$, 64, and 128 grids. The quantity plotted, i.e., the "averaged normalised rms error," is denoted by ε_p. A very low Courant number of $\frac{1}{8}$ was used. The vertical arrows indicate the magnitude of the difference between the linear and "TS13" schemes for each grid. This difference was found to increase greatly with N. The "OPT13" schemes for both $N=64$ and $N=128$ runs being not truly optimal (since the coefficients were determined from a 32^3 grid), and it may be seen that they are out-performed by the "TS13" scheme.

Figure 6 illustrates the time-step effect for $N=128$. For the "TS13" scheme and splines, errors are plotted at several time levels as indicated. An appreciable effect is evident only in the range of Courant number $\frac{1}{2}$ and 1. Since splines give less interpolation error, the relative importance of time step size appears stronger. On a coarser grid, one might expect the grid-size effect to be so dominant that the relatively weak time-step effect might not be discernible at all. This is confirmed by a similar plot of $N=64$ results given in [24].

The relative insignificance of the time-stepping errors observed may also be partly attributed to the fact that numerical stability constraints have, through the Courant number limit, already placed a quite severe limit on the time step size.

Figure 7 shows particle displacement errors plotted against the number of grid points, for the "TS13" scheme, again at five different time levels. In the $N=8$ and $N=16$ cases, the velocity fields were specified by truncating the spectrum at $k_{\max}$ (3 and 7, respectively); consequently they do not have the same spectra as the other cases. Since some of the higher wavenumber content is lost, the errors are less than would otherwise have been expected. The corresponding data points thus lie below the general trend extrapolated. For the 32^3, 64^3, and 128^3 grids, good agreement with the claimed second-order accuracy is evident.

Figure 8 is the corresponding plot for splines, for 32^3, 64^3, and 128^3 grids only. There is fair agreement with the fourth-order accuracy predicted.

Lagrangian Statistics

Since only one realisation is used, significant anisotropy can be expected in the Lagrangian statistics extracted for this frozen field, despite a statistically isotropic velocity field specification through the energy spectrum. An aggravating factor is the fact that Pao's spectrum has no lower cutoff: most of the energy is concentrated

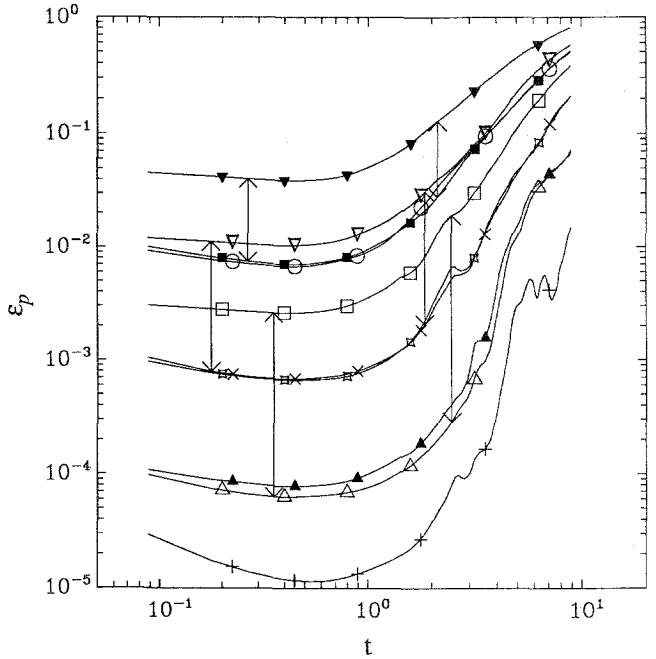

FIG. 5. Frozen turbulent velocity field (Pao's spectrum): averaged normalised rms error in particle displacement (ε_p) against time (t). Grid sizes and interpolation schemes: ■ $N=32$, TS13; ○ $N=32$, OPT13; ▼ $N=32$, linear; × $N=64$, TS13; ⊡ $N=64$, OPT13; ▽ $N=64$, linear; △ $N=128$, TS13; ▲ $N=128$, OPT13; □ $N=128$, linear; + $N=128$, splines.

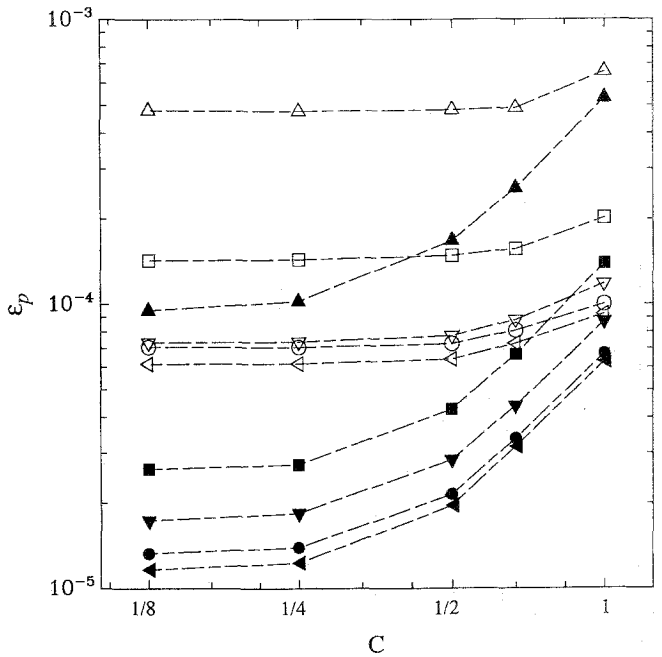

FIG. 6. Frozen velocity field: averaged normalised rms error (ε_p) against Courant No. (C) on 128^3 grid. Time levels expressed as percentages of the total tracking time T ($=8.89$) are (open symbols for "TS13"): ▽ 2; ○ 5; ◁ 10; □ 20; △ 33. Corresponding closed symbols are for splines.

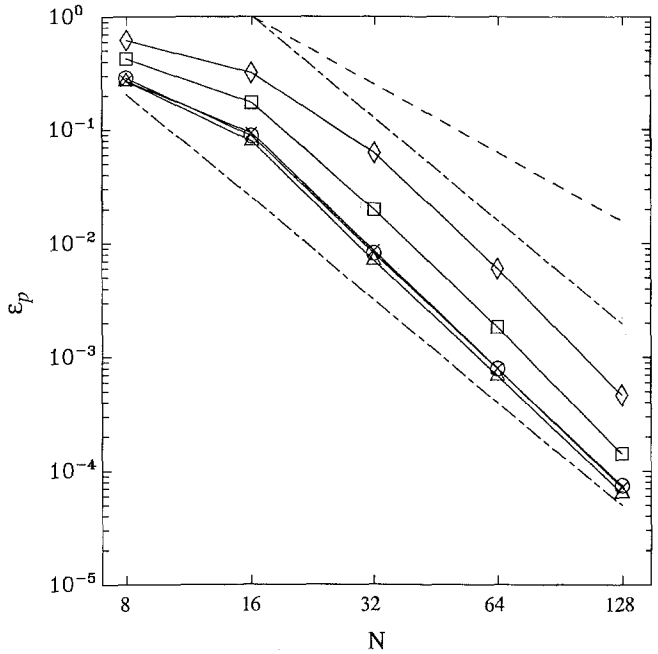

FIG. 7. Frozen velocity field: averaged normalised rms error (ε_p) against grid size (N), for "TS13" scheme, using $C = \frac{1}{2}$, at same selected time levels as in Fig. 6 (see Fig. 6 for symbols): ----- lines of slope -3; --- line of slope -2.

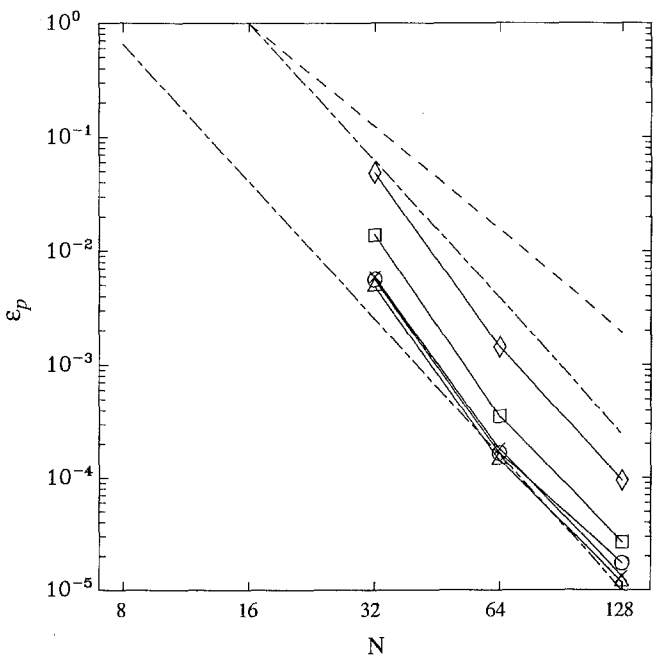

FIG. 8. Same data as Fig. 7, but for splines: ------ lines of slope -4; ------ line of slope -3.

in the first few wavenumbers. Consequently in any realisation considerable anisotropy is present, even though the fields are statistically isotropic. However, these effects have no substantial bearings on the conclusions we seek from directionally averaged results. Velocity autocorrelations were computed using the formulae stated in Appendix A. Figure 9 shows the result derived from the 256^3 "best" run. Integral time scales were obtained as the area under the autocorrelation curves. The fact that the sampled autocorrelation remains less than 0.05 from zero for more than half of the time lag range suggests that the total tracking time period (T) is sufficiently long for memory effects of the initial conditions on particle velocities to die out. Increasing T any further would not alter the results significantly. In fact, we would be interested in particle statistics perhaps only for a few times the integral scale, before their velocities become statistically independent of (and hence uncorrelated with) their initial values.

We notice from Fig. 5 that using the "TS13" scheme, with $k_{\max}\eta = 1$ (on 32^3 grid), particle displacements are mostly within 1 % rms error, especially for early times. This suggests that errors in velocity autocorrelations are likely to be small. Indeed, the corresponding autocorrelation curve (not shown) differs very little from that derived from the 256^3 calculation, with less than 3 % difference in integral time scales.

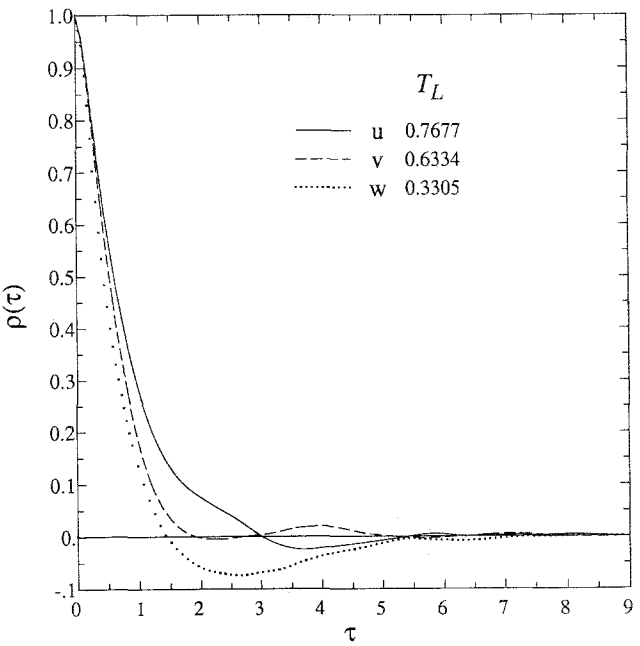

FIG. 9. Frozen velocity field: computed Lagrangian velocity autocorrelations $\rho(\tau)$ against time lag(τ), for the "best" calculation (Conditions: $N = 256$, $C = \frac{1}{8}$, TS13 scheme). Calculated integral time scales (T_L) are indicated for each velocity component.

Root-mean-square particle displacements, i.e., $\langle (x_i^+(t) - x_i^+(0))^2 \rangle^{1/2}$, normalised by the length of the computational box, are shown in Fig. 10 for different ensemble sizes. We have averaged over the three directions. Conditions were $N = 128$, $C = 1$, using the "TS13" scheme. Statistical errors due to having a finite number of particles M appear to be fairly small. For calculations with a frozen velocity field, the choice of M is a compromise between statistical error and computational cost. We have chosen to use 1600 particles for all calculations of this subsection.

The well-known theory due to Taylor [1] on diffusion in stationary homogeneous turbulence predicts asymptotic behaviour for very small and very large times (relative to the integral time scale T_L).

$$\text{For small } t/T_L: \quad \langle (x_i^+(t) - x_i^+(0))^2 \rangle^{1/2} \approx \langle u_i^2 \rangle^{1/2} t, \tag{22}$$

$$\text{For large } t/T_L: \quad \langle (x_i^+(t) - x_i^+(0))^2 \rangle^{1/2} \approx \langle u_i^2 \rangle^{1/2} \sqrt{2T_L t}. \tag{23}$$

The straight lines on Fig. 10 show that our computations agree well with these theoretical results. In fact, agreement is very good even in a quantitative sense, as Fig. 11 shows for the y-direction displacements.

To identify further the effect of interpolation errors on the particle statistics, we

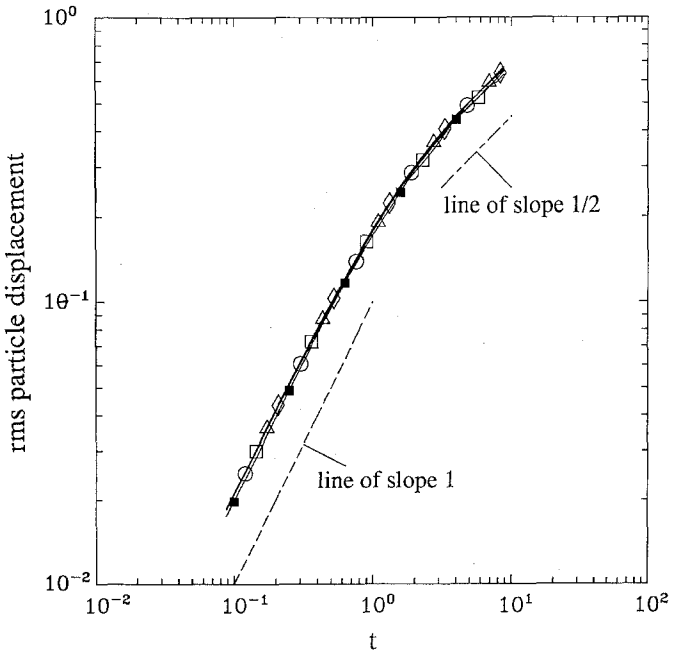

FIG. 10. Frozen velocity field: rms particle displacement against time (t), averaged over the three coordinate directions. Number of particles (M) indicated by symbols: ■ 144; ○ 400; □ 800; △ 1600; ◇ 3200 (conditions: $N = 128$, $C = 1$, TS13 scheme). The $M = 144$ line lies slightly apart from the others.

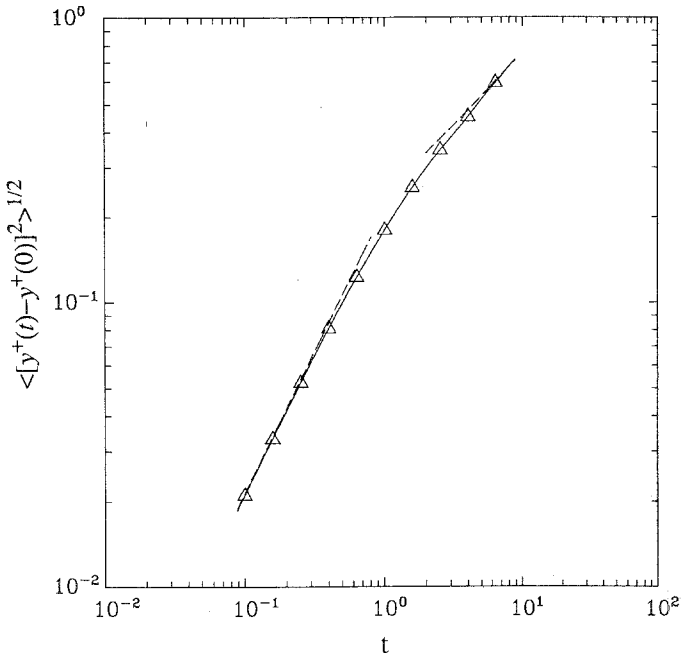

FIG. 11. Frozen velocity field: rms particle y-displacement $\langle (y^+(t)-y^+(0))^2\rangle^{1/2}$ against time (t), using 1600 particles. Dashed lines indicate predicted asymptotic behaviour (conditions: $N=128$, $C=1$, TS13 scheme).

have also calculated second- and fourth-order Lagrangian structure functions. These are defined as

$$D^L_{uu}(\tau) = \langle [u^+(t+\tau)-u^+(t)]^2 \rangle, \tag{24}$$

$$D^L_{uuuu}(\tau) = \langle [u^+(t+\tau)-u^+(t)]^4 \rangle, \tag{25}$$

and are shown together in Fig. 12 for three cases, all using $C=\frac{1}{8}$: (1) the 256^3 "best" calculation; (2) 64^3 with "TS13"; and (3) 32^3 with the linear scheme.

These three cases are chosen with a view towards attempting to parametrise the effect of errors in computed particle displacements on the computed particle statistics. From Fig. 5, we note that cases (2) and (3) give rms displacement errors of order 0.1 % and 5 % (relative to case (1)) up to time $T/2$. An error of 0.1 % is certainly very good accuracy indeed, while 5 % is marginal.

Results for the first two cases hardly differ from each other, but the third is seen to show some appreciable differences while having the same qualitative behaviour. The differences for both structure functions are as much as about 15 % for smaller times.

With the data in this frozen field study, we have thus shown that with $k_{\max}\eta = 1$, using the "TS13" scheme and a small time step, particle displacement errors can be

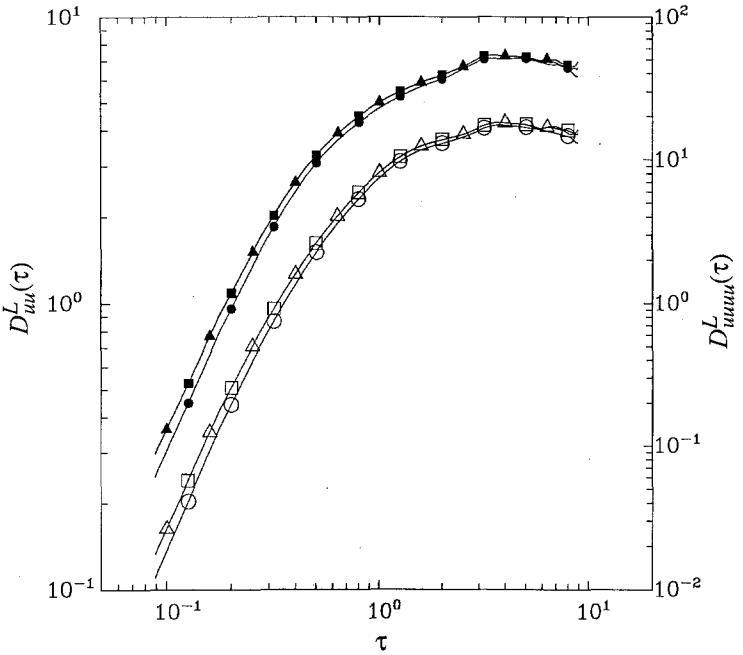

FIG. 12. Frozen velocity field: second- and fourth-order Lagrangian structure function for x-velocity component against time lag(τ), using $C = \frac{1}{8}$. Grid size and interpolation scheme: (open symbols for 2nd order) $\triangle$ $N = 256$, TS13 ("best" calculation); $\square$ $N = 64$, TS13; $\bigcirc$ $N = 32$, linear. Corresponding closed symbols are for 4th order.

kept at about 1 % and hence reasonably accurate (to within a few percent) estimates of Lagrangian statistics can be obtained.

5.4. *Stationary Homogeneous Turbulence with Artificial Forcing*

Having achieved satisfactory results for frozen velocity fields, we now apply our algorithm to a flow with more physical significance: namely, stationary homogeneous turbulence without mean velocity gradients. The energy in the turbulence is maintained by artificially adding energy to the low wavenumber components to compensate for viscous decay. We use the scheme recently proposed by Eswaran and Pope [13], who has verified that the small scales are insensitive to the details of the "forcing" scheme, as they should be.

We give only an abbreviated account of the forcing scheme here; more information is given in [13]. An artificial acceleration $\hat{\mathbf{a}}_F(\mathbf{k})$ is added to the true acceleration $\hat{\mathbf{a}}(\mathbf{k})$ given by the Navier–Stokes equations (Eq. (4)). This forcing acceleration is zero except at the (nonzero) low-wavenumber vector nodes lying within a spherical shell of chosen radius ($2\sqrt{2}$) in wavenumber space, i.e., $\hat{\mathbf{a}}_F(\mathbf{k}) = 0$ unless $0 < k < 2\sqrt{2}$. At each node with forcing, $\hat{\mathbf{a}}_F(\mathbf{k})$ is a vector-valued Ornstein–Uhlenbeck stochastic diffusion process (see, e.g., [25]). The infinitesimal

parameters of such a process are prescribed through the selection of a time scale (T_F) and an amplitude (σ), or, alternatively, T_F and $\sigma^2 T_F$ (also called ε^*).

Starting from selected initial conditions the flow is found to reach a statistically stationary state after a few eddy-turnover times (the eddy turnover time T_E is defined as L/u', where L is the integral length scale and u' is the rms velocity component). Temporal fluctuations in spatially averaged Eulerian statistics persist, but become relatively small and centered around a fairly steady level, thus enabling us to define time averages. All of these Eulerian statistics, e.g., Taylor-microscale (λ), Reynolds number (Re_λ) and dissipation rate (ε), can be treated as stationary random functions of time. A set of fluid particles is then "released" into the simulations, which are subsequently continued with particle tracking to obtain the Lagrangian time series.

We are interested in the effect of $k_{\max}\eta$ on the results. A suitable parametrisation enables us to choose the values of the forcing parameters (T_F and ε^*) to obtain (approximately) desired values of the Kolmogorov microscale. Temporal fluctuations in the flow precludes an a priori prescription of η (or $k_{\max}\eta$), but it is possible to come close to a desired value of η. Let k_0 be the lowest wavenumber present (unity), N_F be the number of wavenumber modes forced (92), and define the quantities

$$\begin{aligned} T_F^* &= T_F \varepsilon^{*1/3} k_0^{2/3}, \\ \gamma &= \frac{4N_F}{1 + T_F^*(N_F)^{1/3}/0.8}, \qquad \varepsilon_T^* = \gamma \varepsilon^*. \end{aligned} \tag{26}$$

The kinematic viscosity ν is also to be prescribed. Figure 13 shows a plot of the relationship found between the nondimensional groups $k_0\eta$ and $(\nu^3/\varepsilon_T^*)^{1/4} k_0$, for three different values of the non-dimensional forcing time scale T_F^* (0.05, 0.15, 0.25). We have fixed T_F^* at 0.15 and ν at 0.025 for our simulations. From Fig. 13, we can select a value of ε_T^* for desired $k_0\eta$, and determine the forcing parameters through the relations (rearranging Eq. (26))

$$\begin{aligned} T_F &= T_F^* \nu^{-1} \gamma^{1/3} (\nu^3/\varepsilon_T^*)^{1/3} k_0^{-2/3}, \\ \varepsilon^* &= \nu^3 \gamma^{-1} (\nu^3/\varepsilon_T^*)^{-1}. \end{aligned} \tag{27}$$

Simulations have been performed on 16^3, 32^3, and 64^3 grids. The forcing parameters used and some of the relevant data are given in Table II. All Eulerian quantities shown are time-averaged values over the tracking time period T, which is about 10 eddy-turnover times. The interpolation scheme used was linear for runs labelled F25 to F28, "TS13" scheme for all other cases. Since time-series analyses are most conveniently performed on data sampled at equal time intervals, the time step is fixed for each simulation. Its value is chosen such that the Courant number fluctuates in a small range near 0.5.

The simulated turbulence is isotropic. Our data agree very well, to within about

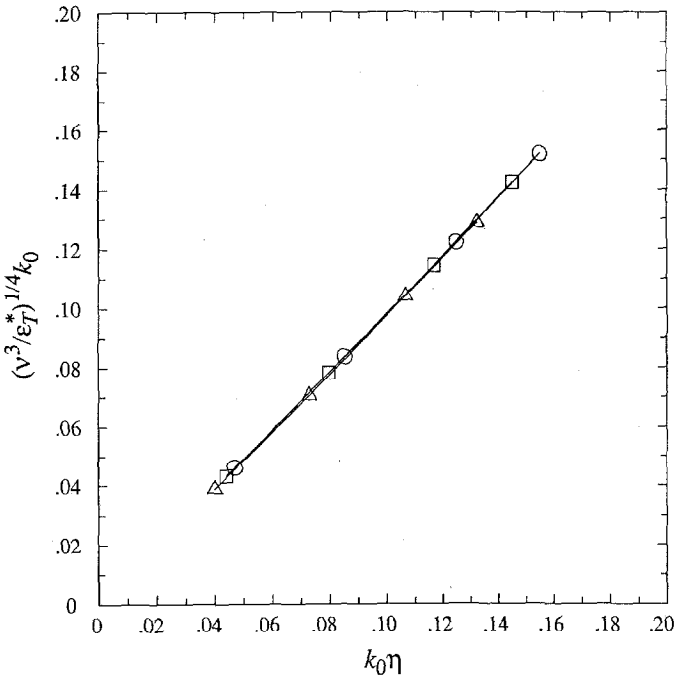

FIG. 13. Forced turbulence: plot of $k_0\eta$ against $(\nu^3/\varepsilon_T^*)^{1/4}\, k_0$, obtained on a 16^3 grid. Three values of T_F^* used: △ 0.05; □ 0.15; ○ 0.25.

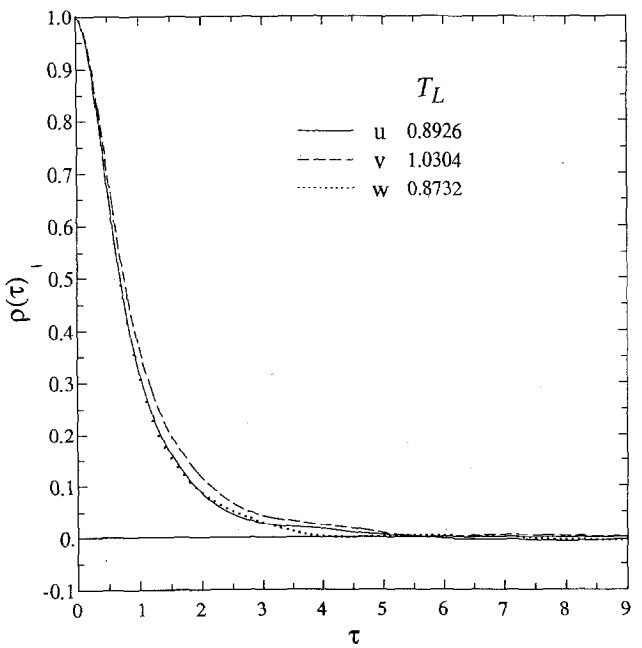

FIG. 14. Forced turbulence: velocity autocorrelations $\rho(\tau)$ against time lag (τ), for the case F22 (conditions: $N = 32$, TS13, $k_{max}\eta = 1$). Calculated integral time scales (T_L) as indicated.

1 %, with many well-known isotropic relations, e.g., $\varepsilon = 15\nu u'^2/\lambda^2$, where u' is the rms velocity.

Figure 14 shows a typical set of velocity autocorrelation curves, for the case F22 (refer to Table II for conditions). Comparing with Fig. 9 for the frozen fields of Section 5.3, we observe a much greater degree of similarity between the curve shapes for the three directions. Directional differences appear to be largely averaged out by the temporal fluctuations, and statistical isotropy prevails strongly. Referring to Table II, in each case the averaged (over three directions) integral time scale is somewhat shorter than the eddy-turnover time.

Each set of forcing parameters represents a different physical situation and Reynolds number. In Fig. 15, second-order Lagrangian structure functions are plotted for the three 64^3 simulations: F31, F32, and F33. We define $D_2^L(\tau)$ to be the average of the three components (D_{uu}^L, D_{vv}^L, D_{ww}^L), and use Kolmogorov scaling, i.e., plotting $D_2^L(\tau)/v_\eta^2$ against τ/τ_η, where v_η and τ_η are the Kolmogorov velocity and time microscales, respectively. It is well known that Kolmogorov's first similarity hypothesis predicts universal small-time behaviour for sufficiently high Reynolds number (see, e.g., Monin and Yaglom [26]). Moreover, for even higher Reynolds numbers, Kolmogorov's second hypothesis predicts the existence of an inertial subrange showing up as a linear segment of the second-order structure functions for

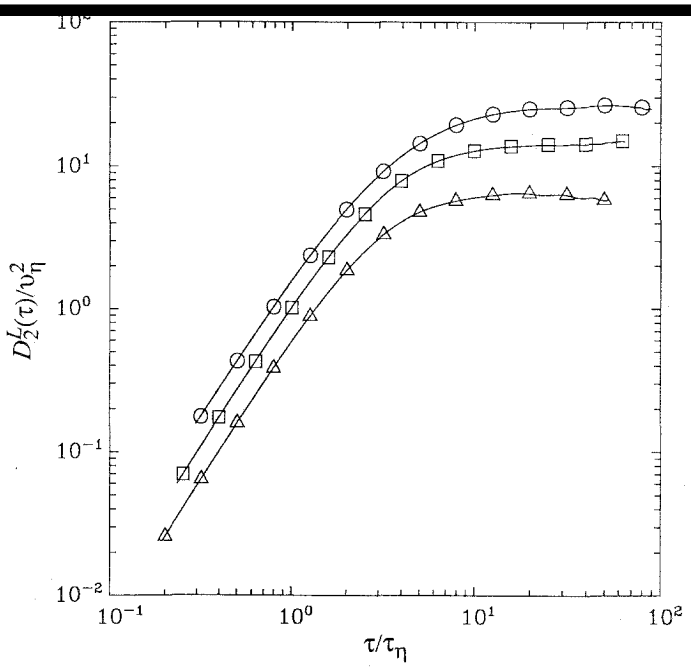

FIG. 15. Forced turbulence: averaged second-order Lagrangian velocity structure function $D_2^L(\tau)$, against time lag (τ), under Kolmogorov scaling. Results from 64^3 simulations: △ F31; □ F32; ○ F33 (see Table II for conditions).

TABLE IIA

Summarised Data for Forced Turbulence Simulations

Case	F11	F21	F31	F12	F22	F32	F13	F23	F33
N	16	32	64	16	32	64	16	32	64
T_F	2.694	2.694	2.694	1.0595	1.0595	1.0595	0.4166	0.4166	0.4166
ε^*	0.000173	0.000173	0.000173	0.002838	0.002838	0.002838	0.04666	0.04666	0.04666
η	0.1428	0.1441	0.1443	0.0705	0.0710	0.0717	0.0352	0.0347	0.0357
$k_{\max}\eta$	0.999	2.161	4.330	0.493	1.065	2.150	0.247	0.520	1.072
Re_λ	12.7	12.2	12.3	25.9	28.6	28.0	53.8	54.7	49.9
S_ε	0.725	0.773	0.760	0.399	0.552	0.568	0.143	0.360	0.509
τ_η	0.8127	0.8283	0.8296	0.1977	0.2010	0.2049	0.0497	0.0481	0.0509
T_E	4.362	4.573	4.371	1.228	1.321	1.308	0.333	0.446	0.385
T	48.0	48.0	44.0	12.5	14.0	13.2	3.5	4.6	4.6
Δt	0.08	0.04	0.02	0.025	0.0125	0.006	0.007	0.0038	0.0019
τ_η/h	3.39	4.14	5.19	3.95	4.02	4.27	3.55	3.16	3.35
T_L	2.970	3.052	2.987	0.919	0.952	0.935	0.259	0.299	0.280

Note. Interpolation scheme used was "TS13" for all 9 simulations listed in this table.

581/79/2-11

TABLE IIB

Summarised Data for Forced Turbulence Simulations (Continued)

Case	F24	F25	F26	F27	F28	F29	F39
N	32	32	32	32	32	32	64
Scheme	TS13	Linear	Linear	Linear	Linear	TS13	
T_F	0.6369	2.694	1.0595	0.4166	0.6369	1.5406	1.5406
ε^*	0.01306	0.000173	0.002838	0.04666	0.01306	0.000923	0.000923
η	0.0485	0.1441	0.0710	0.0347	0.0485	0.0963	0.0920
$k_{\max}\eta$	0.727	2.161	1.065	0.520	0.727	1.445	2.761
Re_λ	39.3	12.2	28.6	54.7	39.3	20.3	21.5
S_ε	0.465	0.773	0.552	0.360	0.465	0.627	0.620
τ_η	0.0932	0.8283	0.2010	0.0481	0.0932	0.3701	0.3384
T_E	0.688	4.573	1.321	0.446	0.688	2.239	2.127
T	7.0	48.0	14.0	4.6	7.0	21.0	
Δt	0.005	0.04	0.0125	0.0038	0.005	0.021	
τ_η/h	3.73	4.14	4.02	3.16	3.73	4.41	
T_L	0.483	3.070	0.984	0.329	0.512	1.526	

Note. For the case F39, only the Eulerian calculation was performed.

highest Re_λ we attained was only about 50 (for the case F33), the Reynolds number range is presumably too low for the hypothesis to apply. Figure 15 clearly shows a Reynolds-number dependence. Still, with the curves corresponding to $\mathrm{Re}_\lambda \approx 12$, 28, and 50, the data do show that this dependence weakens as the Reynolds number is increased. For similar reasons, no inertial subrange (i.e., slope 1) is observed. For comparison, Re_λ was about 80 in the work of Kerr [27], whose results are consistent with the existence of a short inertial subrange.

To investigate the effects of $k_{\max}\eta$ as a measure of spatial resolution and hence difficulty of interpolation, on both Eulerian and Lagrangian statistics, suitable quantities are needed to characterise the quality of the numerical results obtained. For the Eulerian DNS, it is useful to study the normalised dissipation skewness, defined as

$$S_\varepsilon = \frac{4}{35}(15\nu/\varepsilon)^{3/2}\,\nu\int_0^{k_{\max}} k^4 E(k)\,dk. \tag{28}$$

This quantity, as defined, is based on the fourth moment of the energy spectrum, and is thus influenced by how well the small scales (high wavenumber components) are resolved. Presumably, as the Reynolds number increases, more stringent resolution requirements make it necessary to use finer grids. Figure 16 is a plot of S_ε

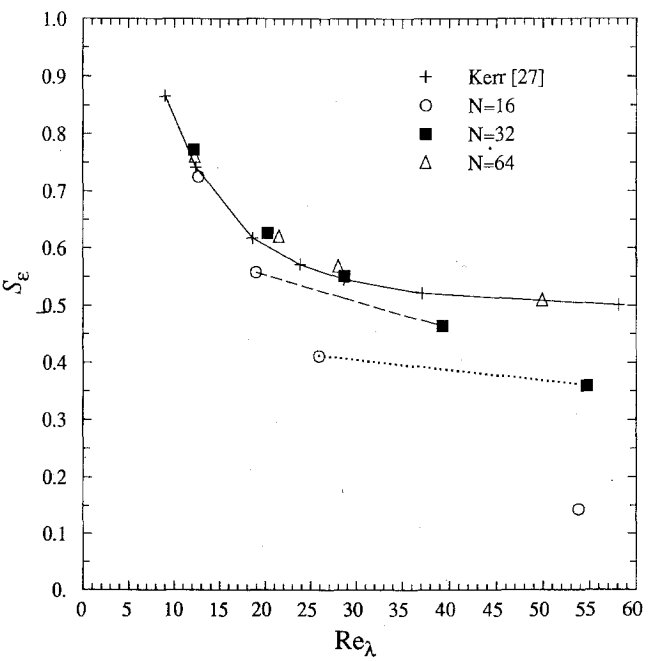

FIG. 16. Forced turbulence: plot of dissipation skewness (S_ε) against microscale Reynolds number (Re_λ); $N=16$: simulations F11, F12, F13; $N=32$: simulations F21, F22, F23, F24, F29; $N=64$: simulations F31, F32, F33, F39. (See Table II for $k_{\max}\eta$ values.) Dashed line: data with $k_{\max}\eta \approx 1/\sqrt{2}$; dotted lines: data with $k_{\max}\eta \approx \frac{1}{2}$.

against Re_λ, for the three grids we used. For $k_{max}\eta > 1$ on the 32^3 and 64^3 grids, there is good agreement with data reported in Kerr's study [27]. However, results are seen to be entirely unsatisfactory for $k_{max}\eta$ appreciably less than 1.

Evidently, then, a value of $k_{max}\eta$ at least 1 is needed to achieve adequate accuracy in the calculated Eulerian velocity fields, on which the particle-tracking calculations are based. Consequently accurate Lagrangian statistics call for $k_{max}\eta$ of at least 1, possibly more.

In Fig. 17, calculated averaged (over three directions) autocorrelations at time lags $\tau/\tau_\eta = 1$, 5, and 10 are plotted against Re_λ, and compared with the most accurate 64^3 results available. Error bars illustrate the magnitude of the statistical errors in the autocorrelations, which are manifested as variability (one standard deviation) amongst the three directions. Clearly, the trend is for the errors to increase with Re_λ. This is readily related to the role of spatial resolution, since higher Re_λ implies smaller η (the small-scale turbulence structure becomes finer in size) and hence smaller $k_{max}\eta$ for a given grid.

Systematic differences are observed between results using the "TS13" and linear schemes on a 32^3 grid. The linear scheme is found to yield invariably a higher autocorrelation, as illustrated in Fig. 17, and the integral time scales are in fact larger by 5 to 10 %. This is attributed to the fact that the linear scheme generates

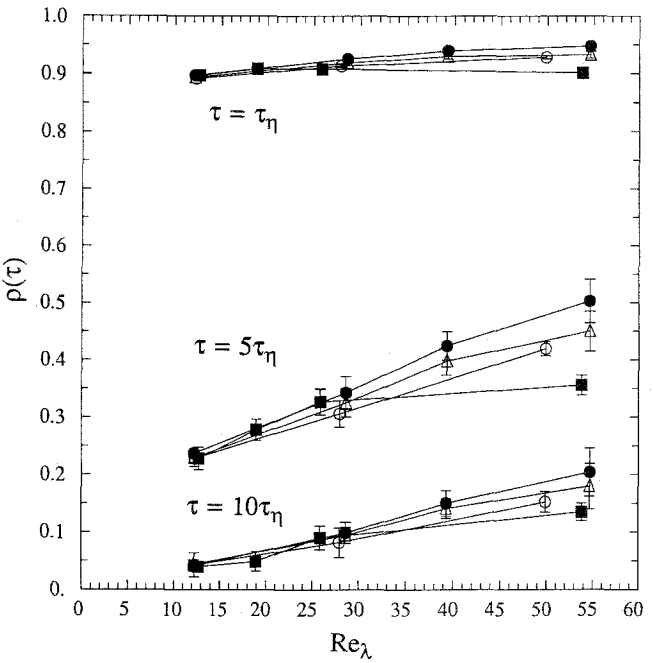

FIG. 17. Forced turbulence: velocity autocorrelations $\rho(\tau)$ against Re_λ, at selected time lags as indicated. Grid size and interpolation scheme: ■ $N = 16$, TS13; △ $N = 32$, TS13; ○ $N = 64$, TS13; ● $N = 32$, linear (see Table II for $k_{max}\eta$ values).

a piecewise-linear interpolated velocity field which varies less rapidly in space, so that the interpolated particle velocity changes less rapidly as the particle moves around. Thus the memory time for particle velocities becomes longer than it should be, and is reflected in a larger sample autocorrelation. Similarly, relative to the 64^3 grid, use of the 32^3 grid tends to lead to overpredicted autocorrelations. The 16^3 grid results behave erratically, presumably because the grid is really too coarse, there being only 7 useful wavenumber modes in each direction.

The effect of $k_{\max}\eta$ on computed velocity structure functions is illustrated in Fig. 18 by the cases F12, F22, and F32 which were solved on 16^3, 32^3, and 64^3 grids, respectively, for the same set of forcing parameters using the "TS13" scheme. For second-order structure functions, the cases F22 and F32 ($k_{\max}\eta \approx 1$ and ≈ 2) virtually collapse together, thus lending further support to our assertion that $k_{\max}\eta = 1$ is a suitable criterion for adequate spatial resolution. The case F12 with $k_{\max}\eta \approx \frac{1}{2}$ deviates considerably from the expected τ^2 slope and the other two cases.

The case F26 has the same conditions as F22, but the linear scheme was used. Significant differences between the two cases are seen in Fig. 18. The structure function is consistently underestimated by the linear scheme, just as the autocorrelations are found to be overestimated. The differences are about 13 % for most of the small time lag ($\tau \leqslant \tau_\eta$) range.

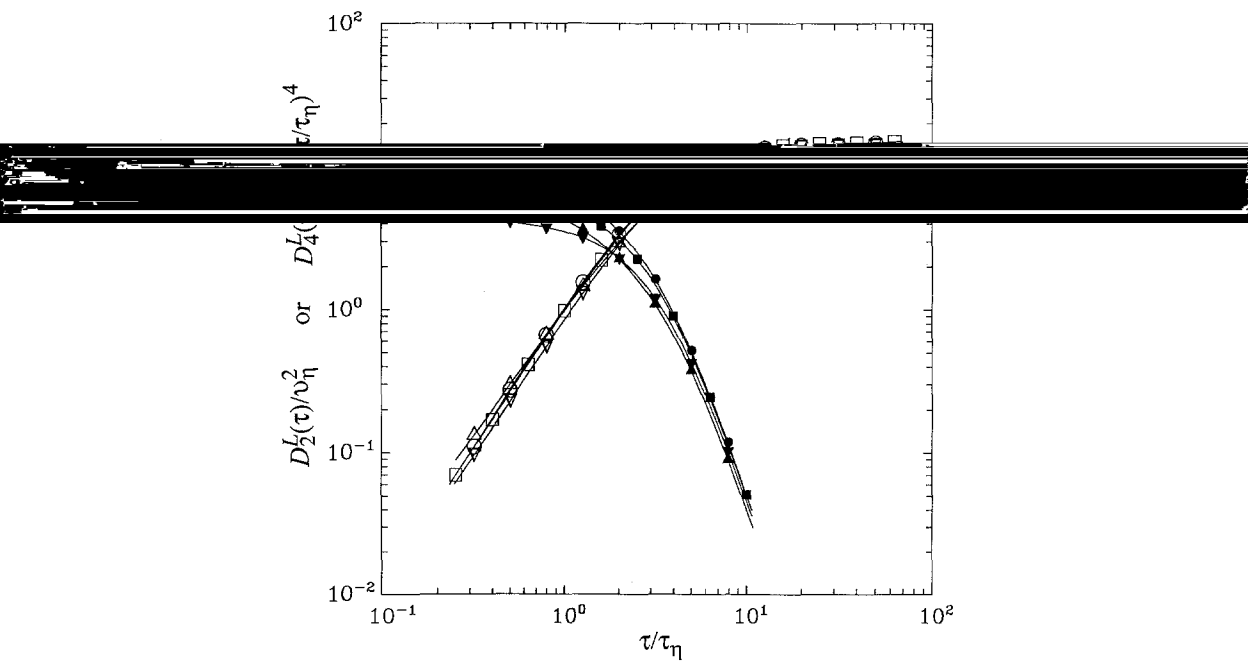

FIG. 18. Forced turbulence: averaged second- ($D_2^L(\tau)$: open symbols) and fourth- ($D_4^L(\tau)$: closed symbols) order Lagrangian velocity structure functions, under Kolmogorov scaling. Results from 4 simulations: △ F12; □ F22; ○ F32; ▽ F26 (see Table II for conditions). The F22 and F32 lines collapse together, but F26 is consistently lower. $D_4^L(\tau)$ is divided by $(\tau/\tau_\eta)^4$.

The evidence so far indicates that with $k_{\max}\eta = 1$, we can achieve adequate accuracy for second-order quantities like the autocorrelation and second-order structure functions, by using the "TS13" scheme for interpolation. Considerable errors would result if the linear scheme is used instead. This is an important finding because it could mean that, by virtue of the higher interpolation accuracy with "TS13," we can simulate flows of a given Reynolds number on a coarser grid (hence much less computational costs) than what might have been necessary otherwise.

Corresponding fourth-order structure functions are also shown under Kolmogorov scaling in Fig. 18, up to $\tau/\tau_\eta = 10$. We denote the average over the three components by $D_4^L(\tau)$. In view of the predicted τ^4 slope for small times, the ordinates are divided by $(\tau/\tau_\eta)^4$ to highlight the differences, i.e., $D_4^L(\tau)/v_\eta^4/(\tau/\tau_\eta)^4$ is plotted against τ/τ_η. At small times a flat line is predicted. Relative to the case F32 with $k_{\max}\eta \approx 2$, $k_{\max}\eta \approx 1$ gives results 27 % in error for smaller times lags ($\tau \leqslant \tau_\eta$) using the "TS13" scheme, but as much as 78 % using the linear scheme. Results for $k_{\max}\eta \approx \frac{1}{2}$ do not even give the correct qualitative behaviour.

As a higher order quantity, it is not surprising to find that accurate calculation of the fourth-order structure function requires a better degree of spatial resolution, and that the simple linear interpolation scheme yields unacceptable results. Apparently, even using "TS13," $k_{\max}\eta$ needs to be at least 2 for acceptable accuracy

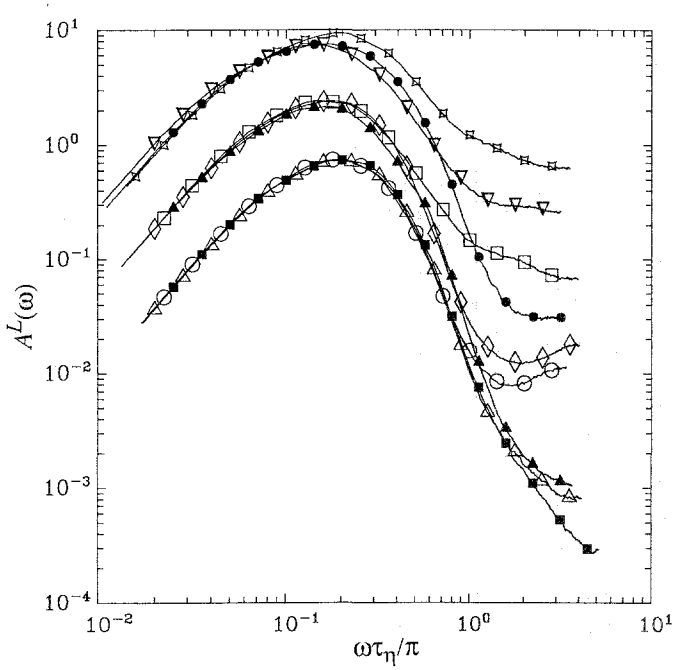

FIG. 19. Forced turbulence: computed Lagrangian acceleration spectrum $A^L(\omega)$ (averaged over 3 directions) against Kolmogorov scaled frequency, for 9 simulations: ○ F11; △ F21; ■ F31; □ F12; ◇ F22; ▲ F32; ⊡ F13; ▽ F23; ● F33 (see Table II for conditions).

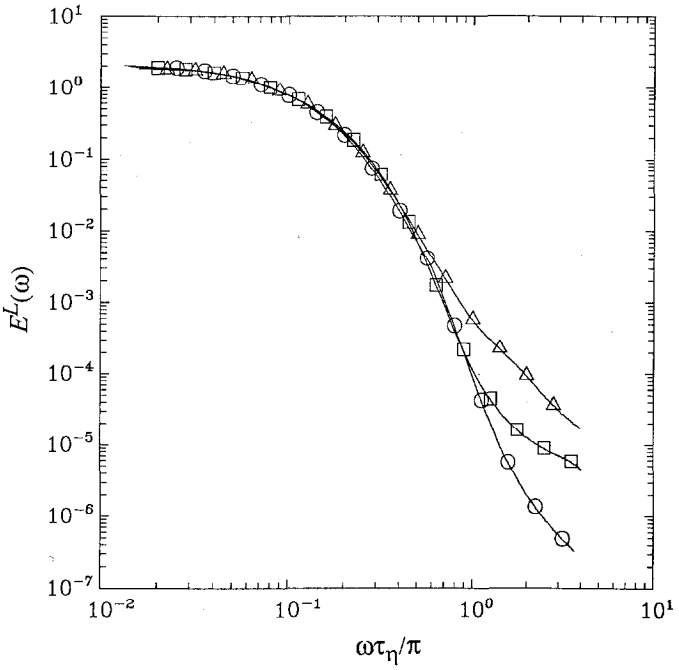

FIG. 20. Forced turbulence: computed Lagrangian velocity spectrum $E^L(\omega)$ (averaged over 3 directions) against Kolmogorov scaled frequency, for 3 simulations: △ F12; □ F22; ○ F32 (see Table II for conditions).

in the fourth-order structure functions. In fact, the case F21 ($k_{\max}\eta \approx 2$) still shows about 10 % difference from the case F31 ($k_{\max}\eta \approx 4$).

Acceleration frequency spectra, after applying a partial noise removal treatment described in Appendix A, for nine cases (3 sets of forcing parameters, 3 grid sizes) are plotted in Fig. 19, with the angular frequency ω scaled by the Kolmogorov frequency (π/τ_η). Evidently the curves tend to converge with increasing $k_{\max}\eta$, and large deviations occur for cases with $k_{\max}\eta$ appreciably smaller than unity. It appears that $k_{\max}\eta$ of at least 2 is needed for really satisfactory results. Flat plateaus expected in an inertial subrange are not observed, because the Reynolds numbers are not high enough, as mentioned before.

A small sampling interval, about $\frac{1}{4}$ of Kolmogorov time scale, is used to capture the high-frequency motions adequately. Thus, despite practical difficulties at the high frequency end in spectral estimation (see, e.g., [28, 29]), we have achieved spectra largely satisfactory up to the Kolmogorov frequency (π/τ_η). The effect of $k_{\max}\eta$ on the velocity spectra is illustrated in Fig 20, for the cases F12, F22, and F32, of differing $k_{\max}\eta$ but nearly equal Re_λ. Again, $k_{\max}\eta$ of 1 appears adequate.

6. Conclusions

Applied within the framework of Eulerian direct numerical simulation of homogeneous turbulence, our particle-tracking algorithm has been thoroughly tested, as described in Section 5. The main conclusions that can be drawn are as follows.

First, interpolation by cubic splines gives the least errors on all the numerical tests conducted and has the additional advantage of being C^2 continuous.

Second, since splines are difficult to use in the original "planewise" version of Rogallo's DNS code (owing to data structure and memory limitations), an accurate alternative scheme is needed. For use with the "planewise" code, the "TS13" (Taylor series 13-point) interpolation scheme we developed is highly successful, yielding results demonstrably superior to those given by linear interpolation, which is unacceptably poor. This improvement is largely derived from the reduction of internodal distances by the addition of shifted planes of data to Rogallo's DNS code. Third-order accuracy has been ascertained with the flows of Sections 5.2 and 5.3. While it does not yield continuous approximations, "TS13" is considered of adequate accuracy for most purposes.

Third, studies with the helical flow have enabled us to isolate separate contributions to errors in computed particle trajectories. Interpolation error is seen to be the major source of errors. The time-stepping error is generally much less significant. This is presumably because the time step is restricted to small values by enforcement of the Courant number stability limit. A Courant number of $\frac{1}{2}$ appears small enough for the time-step dependence of the results to be negligible.

Fourth, our algorithm has been sucessfully applied to extract Lagrangian statistics for frozen turbulence and stationary homogeneous turbulence without mean velocity gradients. A successful forcing scheme is used for the latter flow to compensate for viscous decay. Results agree well with Taylor's theory of one-dimensional turbulent diffusion, as well as the small-time behaviour of the structure functions as predicted by Kolmogorov's first similarity hypothesis. The Reynolds numbers treated are not high enough for an inertial subrange to be observed.

Finally, the nondimensional group $k_{\max}\eta$ is found to be a useful parameter for characterising the degree of spatial resolution in the velocity fields. A criterion for acceptable resolution of most small-scale statistics may be proposed as $k_{\max}\eta \geqslant 1$. When this criterion is met, adequate accuracy may be achieved using the "TS13" scheme and a sufficiently small time step, for second-order quantities such as the velocity autocorrelation and second-order structure functions. A higher value of $k_{\max}\eta$ is needed for higher order quantities. For instance, it needs to be at least 2 for about 10 % error in the fourth-order structure functions.

The emphasis of this paper has been on using simple flows to demonstrate means of attaining adequate numerical accuracy. This objective has been achieved. In subsequent work, we have used [31] 128^3 simulations to attain higher Reynolds numbers without sacrificing spatial resolution. Many important physical questions can be addressed as we apply our algorithm to more realistic and interesting flows. This

algorithm can be readily extended to study the processes of scalar mixing and material line stretching, for example.

APPENDIX A: Estimation of Lagrangian Statistics

Some issues of statistical estimation are addressed here. As a fluid particle is convected in a homogeneous turbulent velocity field, its velocity may be regarded as a [illegible] $u(\tau)$. For such a process, the velocity autocovariance $R(\tau)$ and autocorrelation $\rho(\tau)$ can be computed as a function of time lag (τ) only, and, moreover, it becomes asymptotically uncorrelated with itself for large time lags. The Lagrangian integral time scale T_L (defined as $\int_0^\infty \rho(\tau)\, d\tau$) is estimated by integrating the autocorrelation with respect to time, provided the integral converges.

In practice, the tools of time-series analysis are used to estimate these statistics from finite-length data records of particle velocities taken at equal sampling intervals of size h. In view of the small size time step and the large number of steps taken in each simulation, it is usually neither necessary nor practical to extract output at every time step. Instead, the interval h is chosen as a multiple of the time step Δt. Let the jth sampled time level be $t_j = jh$, with $j = 0, 1, ..., n$ (with $j = 0$ representing the initial values), and $\langle\ \rangle_M$ denote averaging over M particles. Sample autocovariances $\hat{R}$ and autocorrelations $\hat{\rho}$ (which approximate $R(\tau)$ and $\rho(\tau)$, respectively) are calculated by the formulae

$$\hat{R}(jh) = \left\langle \frac{1}{n} \sum_{r=0}^{n-|j|} u^+(rh)\, u^+((r-|j|)h) \right\rangle_M, \tag{A1}$$

$$\hat{\rho}(jh) = \frac{\hat{R}(jh)}{\hat{R}(0)}, \qquad j = 0, 1, ..., n. \tag{A2}$$

This choice of estimator, though biased for finite n, is in popular use (see, e.g., [28]). Essentially, it weights down the sample autocovariances at larger time lags (which are more erratic because of fewer samples being available) to achieve the positive semi-definiteness property possessed by the true autocovariance function.

The Fourier cosine transform of the velocity autocovariance (an even function) gives the Lagrangian velocity frequency spectrum $E^L(\omega)$ (ω is the angular frequency). For $\omega_k = k\pi/nh$ ($k = 0, \pm 1, ..., \pm n$), we calculate the spectral estimate as

$$E^L(\omega_k) = \frac{h}{2\pi}\left[2\left(\sum_{j=1}^{n-1} \hat{R}(jh)\cos(\omega_k t_j)\right) + \hat{R}(0) + \hat{R}(nh)\right]. \tag{A3}$$

Similarly, transforming the acceleration autocovariance yields the acceleration

spectrum $A^L(\omega)$. Since the calculated spectra are, to some extent, contaminated with numerical noise, they do not precisely satisfy the exact relationship

$$A^L(\omega) = \omega^2 E^L(\omega). \tag{A4}$$

It is found that $A^L(\omega)/\omega^2$ provides a more accurate estimate of the high frequency part of the velocity spectrum than does $E^L(\omega)$ obtained directly from the velocity autocovariance. Hence we follow Mestayer [30] in obtaining the low frequency part of the velocity spectrum from the velocity autocovariance, and the high frequency part from the acceleration autocovariance. Noise effects causing unphysical overestimation of the high frequency part are much reduced.

The sampling time interval h chosen determines the highest frequency we can estimate, viz., π/h. Figure A1 shows the effect of varying h on the velocity spectra derived for the forced turbulence simulation F22, with $N = 32,400$ particles and the "TS13" scheme being used. Values of h/τ_η are approximately $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, and 1. Figure A2 shows the corresponding acceleration spectra, the range of spurious behaviour (due to residual noise effects) is "delayed" towards the higher frequencies as h is decreased, i.e., as the sampling rate is made faster. The velocity spectra are

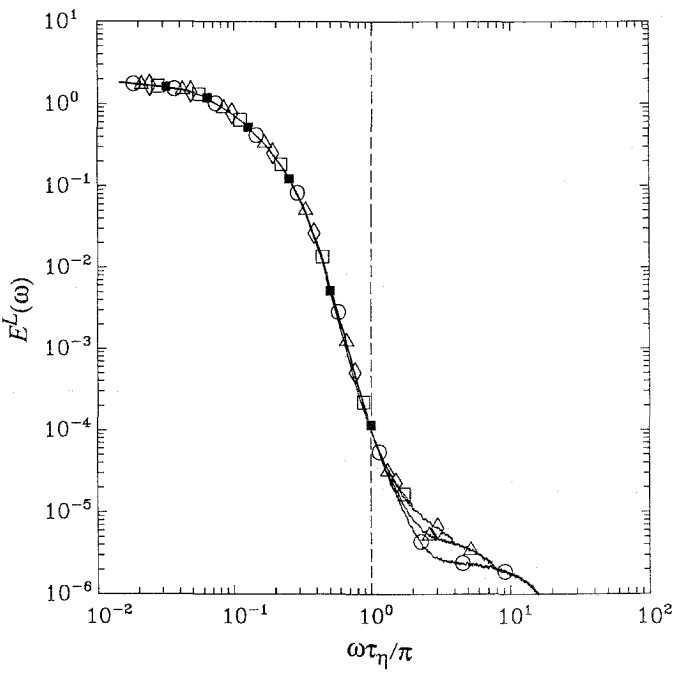

FIG. A1. Forced turbulence: computed Lagrangian velocity spectrum $E^L(\omega)$ (averaged over 3 directions) against Kolmogorov scaled frequency, for conditions of F22, but different sampling time intervals, as fractions of Kolmogorov time scale: ■ 1; □ $\frac{1}{2}$; ◇ $\frac{1}{4}$; △ $\frac{1}{8}$; ○ $\frac{1}{16}$. Dashed vertical line indicates Kolmogorov frequency π/τ_η.

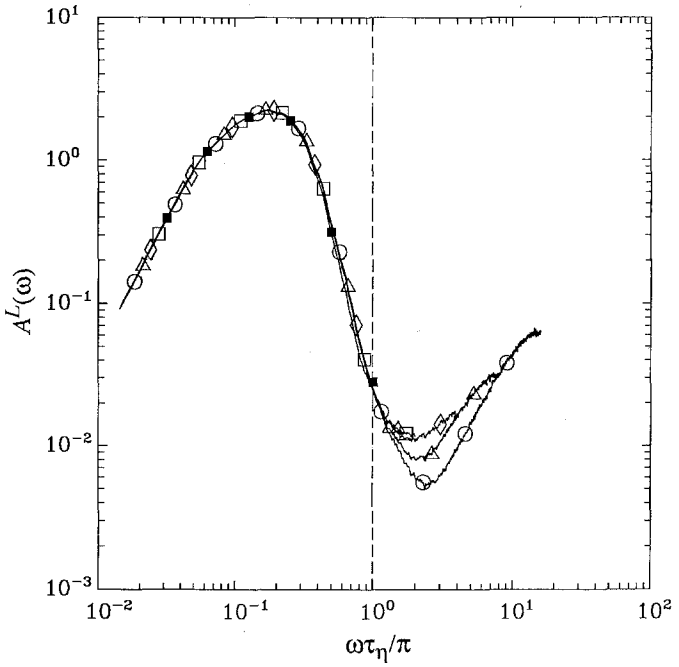

FIG. A2. Forced turbulence: same as Fig. A1, but for acceleration spectrum.

deemed satisfactory if they decrease steadily at least up to the Kolmogorov frequency. For the case presented, $h/\tau_\eta = \frac{1}{4}$ is small enough to ensure this.

APPENDIX B: DETAILS OF INTERPOLATION SCHEMES

Complete Formula for "TS13" Scheme

We follow the notation of Section 4, using local position coordinates with origin at node 9 (of the interpolation cell in which the fluid particle lies) and in units of one grid spacing. The position of the particle is denoted as $\mathbf{x} = (x, y, z)$. It suffices to give the interpolation weights for nodes 1 to 13. We have

$$a_1 = [-x-y-z+2y^2+2xy+2xz+2yz-4xyz]/4,$$
$$a_2 = [-x-y+z+2y^2+2xy-2xz-2yz+4xyz]/4,$$
$$a_3 = [x-y+z+2y^2-2xy+2xz-2yz-4xyz]/4,$$
$$a_4 = [x-y-z+2y^2-2xy-2xz+2yz+4xyz]/4,$$
$$a_5 = [-x+y-z+2y^2-2xy+2xz-2yz+4xyz]/4,$$
$$a_6 = [-x+y+z+2y^2-2xy-2xz+2yz-4xyz]/4,$$

$$a_7 = [x + y + z + 2y^2 + 2xy + 2xz + 2yz + 4xyz]/4,$$
$$a_8 = [x + y - z + 2y^2 + 2xy - 2xz - 2yz - 4xyz]/4,$$
$$a_9 = 1 - x^2 - 2y^2 - z^2,$$
$$a_{10} = (z^2 - y^2)/2,$$
$$a_{11} = (x^2 - y^2)/2,$$
$$a_{12} = a_{10},$$
$$a_{13} = a_{11}.$$

It may be noted that the collocation condition holds for all nodes lying in the cell or on its boundary (1 to 9), but not the external nodes (10 to 13).

"TS8" and "TS9" Schemes

The essential principles involved have been set forth in Section 4. For each scheme, depending on in which pyramid the fluid particle lies, a different interpolation formula is used, but all such formulae are clearly similar and are derived in the same manner. For brevity we will give only one such formula for each. Moreover, although there are many different ways to accomplish the required partitioning of the interpolation cell, the precise choice is quite arbitrary and is not expected to have any effect on the interpolation accuracy.

For the above reasons it suffices to state the partitions that we have chosen and the resulting formula in one of those pyramids for each scheme.

The partitioning is uniquely determined by specifying the vertices of each of the pyramids (there are 6 such pyramids for "TS8" and 12 for "TS9"). For "TS8," the vertex combinations used are: nodes (1, 2, 4, 5), (2, 3, 4, 5), (3, 4, 5, 8), (2, 5, 6, 7), (2, 5, 7, 8), and (2, 3, 7, 8). Within the pyramid (1, 2, 4, 5), the interpolation formula is

$$g(\mathbf{x}) = (1 - x - y - z) f_1 + z f_2 + x f_4 + y f_5, \tag{A5}$$

(the subscripted f's stand for nodal values). The leading truncation error is of second order.

For "TS9", the vertex combinations used are: nodes (1, 2, 5, 9), (1, 2, 3, 9), (1, 3, 4, 9), (1, 4, 5, 9), (4, 5, 8, 9), (3, 4, 8, 9), (2, 5, 6, 9), (5, 6, 7, 9), (5, 7, 8, 9), (2, 3, 6, 9), (3, 6, 7, 9), and (3, 7, 8, 9). Within the pyramid (1, 2, 5, 9), the interpolation formula is

$$g(\mathbf{x}) = (1 - y - z) f_1 + (z - x) f_2 + (y - x) f_5 + 2x f_9. \tag{A6}$$

The leading truncation error is also of second order.

Derivation for "Optimal" Schemes

Let an overbar denote an average over N^3 interpolation cells. We use the notation of Eqs. (13) and (14) and write $f(\mathbf{x})$ as f. For given $\mathbf{x}$, the task is to minimise

$\bar{\varepsilon}$ subject to the constraint $\sum_{i=1}^{p} a_i = 1$ (with the nodes numbered from 1 to p). Substituting Eq. (13) into (14) gives

$$\bar{\varepsilon} = \sum_{i=1}^{p} \sum_{k=1}^{p} a_i a_k \overline{f_i f_k} - 2 \sum_{i=1}^{p} a_i \overline{f f_i} + \overline{f^2},$$

from which

$$\frac{\partial \bar{\varepsilon}}{\partial a_j} = 2 \sum_{k=1}^{p} a_k \overline{f_j f_k} - 2\overline{f f_j}.$$

We are to minimise

$$\chi = \bar{\varepsilon} + \lambda \left(\sum_{i=1}^{p} a_i - 1 \right),$$

where λ is a Lagrange multiplier.

Forming $\partial\chi/\partial a_j$ and $\partial\chi/\partial\lambda$ leads to the matrix equation

$$\begin{bmatrix} \mathbf{C} & \mathbf{I}/2 \\ \mathbf{I}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{B} \\ 1 \end{bmatrix}, \tag{A7}$$

where $C_{ik} = \overline{f_i f_k}$, $A_i = a_i$, $B_i = \overline{f f_i}$, $\mathbf{I}$ is a vector of p unity elements and $\mathbf{I}^T$ is its transpose. Equation (A7) is readily solved by methods appropriate for symmetric linear systems of equations, such as Cholesky factorisation.

Cubic Spline Basis Functions

One-dimensional expressions are given below. Let the nodal points be numbered as x_1 to x_N, and define $x_0 = x_1 - \Delta x$, $x_{N+1} = x_N + \Delta x$, $x_{N+2} = x_N + 2\,\Delta x$. With ξ defined as $(x - x_i)/\Delta x$, for $i = 0, 1, ..., N+1, N+2$, the ith basis function at position x is given by

$$b_i(x) = \begin{cases} \frac{1}{6}(2+\xi)^3 & -2 < \xi < -1 \\ \frac{1}{6}(-3\xi^3 - 6\xi^2 + 4) & -1 < \xi < 0 \\ \frac{1}{6}(3\xi^3 - 6\xi^2 + 4) & 0 < \xi < 1 \\ \frac{1}{6}(2-\xi)^3 & 1 < \xi < 2 \\ 0 & \text{otherwise.} \end{cases} \tag{A8}$$

APPENDIX C: PARTICLE-TRACKING BY NON-DIFFUSIVE SCALARS

Besides tracking fluid particles by directly calculating incremental displacements at each time step, as we have done, we have also considered an indirect approach using non-diffusive scalars, By definition, the material derivative of a non-diffusive scalar is zero. Thus wherever a fluid particle goes, the value of the scalar associated

with it remains unchanged. This invariance property is evidently shared by the initial position coordinates of the fluid particle.

Let $\boldsymbol{\phi}(\mathbf{x}, t)$ be the initial position (at $t=0$) of a fluid particle located at position $\mathbf{x}$ at time t. Thus $\boldsymbol{\phi}(\mathbf{x}^+[t], t) = \mathbf{x}^+(0) = \mathbf{y}$. Since position $\mathbf{x}$ is occupied by different particles at different times, $\boldsymbol{\phi}(\mathbf{x}, t)$ fluctuates and may be decomposed into mean and fluctuating parts as $\boldsymbol{\phi}(\mathbf{x}, t) = \langle \boldsymbol{\phi}(\mathbf{x}, t) \rangle + \boldsymbol{\phi}'(\mathbf{x}, t)$. For the flows considered (homogeneous isotropic turbulence), the expected location of a fluid particle is always its initial location. Hence $\langle \boldsymbol{\phi}(\mathbf{x}, t) \rangle = \mathbf{x}$. Differentiating, we see that it is appropriate to specify the mean gradients $\partial \langle \phi_i \rangle / \partial x_j$ as the Kronecker delta tensor. Moreover, the initial fluctuations $\boldsymbol{\phi}'(\mathbf{x}, 0)$ are zero since $\boldsymbol{\phi}(\mathbf{x}, 0) = \mathbf{x}$ by definition.

With these relationships it follows that $-\boldsymbol{\phi}'(\mathbf{x}, t)$ gives the displacement (from its initial position) of a fluid particle which at time t is at position $\mathbf{x}$. When a specific particle with given initial position $\mathbf{y}$ is concerned, the displacement is $-\boldsymbol{\phi}'(\mathbf{x}^+(\mathbf{y}, t), t)$. The fluctuating scalars following the fluid particle are obtained by interpolation, in exactly the same manner as for particle velocities.

While this method is sound in principle, in practice it is numerically ill-conditioned. Without the action of a molecular diffusivity, transport is solely by advection, and large gradients of the fluctuating scalars develop. The scalar spectra are observed to exhibit excessive growth at the high wavenumber end, because of the absence of a molecular dissipative mechanism. Spatial resolution thus quickly deteriorates, making the Eulerian scalar fields become progressively degraded in numerical accuracy.

This "Eulerian" approach, even if successful, would not be numerically economical because incorporating the scalars leads to a substantial increase in the computational requirements for the Eulerian code. We originally intended to use it to check against the directly calculated particle trajectories. However, as we can see from Fig. 4b, the accuracy is so poor that this method is not a feasible alternative at all, and had to be abandoned.

ACKNOWLEDGMENTS

This work was supported by the U.S. Air Force Office of Scientific Research Grant AFOSR-85-0083. Computations conducted during the research were performed on the Cornell National Supercomputer Facility, which is supported in part by the National Science Foundation, New York State, and the IBM Corporation.

REFERENCES

1. G. I. TAYLOR, *Proc. Lond. Math. Soc. Ser. 2* **20**, 196 (1921).
2. A. S. MONIN AND A. M. YAGLOM, *Statistical Fluid Mechanics, Vol. 1*, edited by J. L. Lumley (MIT Press, Cambridge, MA, 1971).
3. G. K. BATCHELOR, *An Introduction to Fluid Dynamics* (Cambridge Univ. Press, London, 1967).
4. W. H. SNYDER AND J. L. LUMLEY, *J. Fluid Mech.* **48**, 41 (1971).
5. D. J. SHLIEN AND S. CORRSIN, *J. Fluid Mech.* **62**, 255 (1974).

6. Y. SATO AND K. YAMAMOTO, *J. Fluid Mech.* **175**, 183 (1987).
7. J. W. DEARDORFF AND R. L. PESKIN, *Phys. Fluids* **13**, 584 (1970).
8. J. J. RILEY AND G. S. PATTERSON, JR., *Phys. Fluids* **17**, 292 (1974).
9. S. A. ORSZAG AND G. S. PATTERSON, JR., *Phys. Rev. Lett.* **28**, 76 (1972).
10. R. S. ROGALLO AND P. MOIN, *Annu. Rev. Fluid Mech.* **14**, 99 (1984).
11. R. S. ROGALLO, National Aeronautics and Space Administration Technical Memorandum No. 81315, 1981 (unpublished).
12. G. K. BATCHELOR, *Proc. Roy. Soc. Lond. Ser. A* **213**, 349 (1952).
13. V. ESWARAN AND S. B. POPE, *Comput. Fluids* **16**, 257 (1988).
14. M. J. LEE AND W. C. REYNOLDS, Report No. TF-24, Dept. of Mech. Engg., Stanford University, 1985 (unpublished).
15. R. J. ROACHE, *Computational Fluid Dynamics* (Hermosa, Albuquerque, NM, 1976).
16. R. PEYRET AND T. D. TAYLOR, *Computational Methods for Fluid Flow* (Springer-Verlag, New York, 1983).
17. H. TENNEKES AND J. L. LUMLEY, *A First Course in Turbulence* (MIT Press, Cambridge, MA, 1972).
18. M. ABRAMOWITZ AND I. A. STEGUN, *Handbook of Mathematical Functions* (National Bureau of Standards, Washington, DC, 1964).
19. E. O. BRIGHAM, *The Fast Fourier Transform* (Prentice–Hall, Englewood Cliffs, NJ, 1974).
20. K. H. HUEBNER AND E. A. THORNTON, *The Finite Element Method for Engineers* (Wiley-Interscience, New York, 1982).
21. C. DE BOOR, *A Practical Guide to Splines* (Springer-Verlag, New York, 1978).
22. J. H. AHLBERG, E. N. WILSON, AND J. L.WALSH, *The Theory of Splines and Their Applications* (Academic Press, New York, 1967).
23. Y. H. PAO, *Phys. Fluids* **8**, 1063 (1965).
24. P. K. YEUNG AND S. B. POPE, Report No. FDA-87-6, Sibley School of Mech. & Aero Engg., Cornell University, 1987 (unpublished).
25. S. KARLIN AND H. M. TAYLOR, *A Second Course in Stochastic Processes* (Academic Press, New York, 1981).
26. A. S. MONIN AND A. M. YAGLOM, *Statistical Fluid Mechanics, Vol. 2*, edited by J. L. Lumley, MIT Press, Cambridge, MA, 1975).
27. R. M. KERR, *J. Fluid Mech.* **153**, 31 (1985).
28. M. B. PRIESTLEY, *Spectral Analysis and Time Series* (Academic Press, London, 1981).
29. G. M. JENKINS AND D. G. WATTS, *Spectral Analysis and its Applications* (Holden–Day, San Francisco, 1968).
30. P. MESTAYER, *J. Fluid Mech.* **125**, 475 (1982).
31. P. K. YEUNG AND S. B. POPE, Report No. FDA-88-16, Sibley School of Mech. & Aero Engg., Cornell University, 1988 (unpublished).