JOURNAL OF COMPUTATIONAL PHYSICS 83, 96–125 (1989)

Methods for Evaluating Fluid Velocities in Spectral Simulations of Turbulence

S. BALACHANDAR

Division of Engineering, Brown University, Providence, Rhode Island 02912

AND

M. R. MAXEY

Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912

Received September 16, 1987; revised August 12, 1988

In studying particle motion with spectral similations of turbulence, it is necessary to evaluate the local fluid velocity at the instantaneous particle position. In general, this point does not coincide with a mesh point. A direct summation of the spectral Fourier series is the most accurate method but is very time consuming. Various approximate methods are tested and comparisons made of both their accuracy and the computational effort required.

1. INTRODUCTION

Spectral simulation methods for the computation of turbulent flows are the most efficient and accurate for problems where they may be applied [5]. They make the best use of available spatial resolution and, where possible, are the method of choice in investigating turbulent flows. For homogeneous turbulence the flow field $\mathbf{u}(\mathbf{x}, t)$ is specified by a Fourier series representation

$$\mathbf{u}(\mathbf{x}, t) = \sum_{\mathbf{k}} \hat{\mathbf{u}}(\mathbf{k}, t) \exp(i\mathbf{k} \cdot \mathbf{x}) \tag{1.1}$$

where the wavenumber components are integer multiples of the form

$$k_i = \pm n_i \left(\frac{2\pi}{L}\right) \qquad n_i = 0, 1, 2, ..., N/2, \; i = 1, 2, 3, \tag{1.2}$$

and periodic boundary conditions are applied on a box of side L. The spectral coefficients $\hat{\mathbf{u}}(\mathbf{k}, t)$ then evolve in time according to the Navier–Stokes equation for incompressible flow.

The study of turbulent diffusion and other related problems requires the com-

0021-9991/89 $3.00

putation of Lagrangian particle trajectories $\mathbf{X}(t)$, which move in response to the local fluid velocity

$$\frac{d}{dt}\mathbf{X}(t) = \mathbf{u}(\mathbf{x} = \mathbf{X}(t), t). \tag{1.3}$$

Central to this is the ability to evaluate $\mathbf{u}(\mathbf{x} = \mathbf{X}(t), t)$ accurately and efficiently for a large number of particles. Equation (1.1) evaluates the fluid velocity at the particle position, $\mathbf{u}(\mathbf{x} = \mathbf{X}(t), t)$, most accurately, up to the accuracy available from the flow simulation. For a small number of particles the direct summation of (1.1) is feasible, but the computation time makes this prohibitive for a large number of particles, say 1000.

In a pseudo-spectral simulation of turbulence, the fluid velocity $\mathbf{u}(\mathbf{x}, t)$ is computed at the N^3 grid points:

$$(x_i, y_j, z_1) = \left(\frac{L}{N} i, \frac{L}{N} j, \frac{L}{N} 1\right), \qquad i, j, 1 = 1, 2, ..., N \tag{1.4}$$

from the spectral coefficients by a fast Fourier transform (FFT). Generally the evaluation of these velocities at the grid points does not involve extra computational effort, since this evaluation is carried out as part of the simulation of the turbulent flow. The fluid velocity at the particle position, which in general will not coincide with the grid points, must then be evaluated by some other means. One such approach is to use an interpolation scheme on the grid point velocity data. For example, Riley and Patterson [9], in their turbulent diffusion experiment, used a 3-dimensional linear interpolation method to evaluate the local fluid velocity. Until now, direct summation (Haidvogel [2]) and linear interpolation have been the most widely used methods, because of their simplicity. Computationally linear interpolation is the faster scheme but at the expense of accuracy. More recently Yeung and Pope [11, 12] have shown that the accuracy of linear interpolation is rather poor and have suggested an alternative interpolation scheme, which is discussed later.

The question of accuracy is more acute in studies of the relative diffusion of two particles or processes which depend on the relative motion of two particles, such as particle coagulation [1]. In these instances the relative velocities of two particles must be determined accurately. For two particles close together it is possible that the approximation for the individual particle velocities (1.3) may be sufficiently accurate in absolute terms, but that the estimate for the relative velocity may be significantly in error. The determination of relative velocities emphasises the role of high wavenumber contributions to the flow field (1.1), and while low-order approximation schemes may work reasonably well for the low wavenumber components, they are generally poor for higher wavenumbers.

In this study we will consider alternative interpolation schemes such as partial Hermite interpolation, Lagrangian interpolation, and a shape function method.

These schemes lie somewhere between direct summation and linear interpolation both in accuracy and the computational time required. Specifically we are interested in interpolation schemes for which the time to evaluate (1.3) does not exceed the time taken to compute the dynamics of the turbulent flow field. The aim of this paper is to examine these interpolation schemes for both accuracy and the computational effort required, and to see how well these schemes perform in simulations of two-particle and one-particle turbulent dispersion. In doing so this paper goes beyond the work of Yeung and Pope [11]. In Section 2 the interpolation schemes considered and their numerical implementations are described. In Section 3 the accuracy of these schemes for individual Fourier modes at different wavenumbers is examined and these results are combined to estimate accuracy for a typical spectrum of Fourier modes. Then in Section 4 specific numerical tests are made for one-particle and two-particle dispersion in various flow fields generated by a direct simulation of homogeneous turbulence. Finally some guidelines as to the suitability of the various schemes are given for different types of problems. All the computations reported here were performed on a Cyber 205 computer at the John von Neumann Center in Princeton.

2. Velocity Evaluation Methods

In this study five principal methods of velocity evaluation are considered. These are direct summation (DS), Lagrangian interpolation (LGI), partial Hermite interpolation (PHI), linear interpolation (LNI), and shape function method (SFM). In each of these methods the 3-dimensional fluid velocity $\mathbf{u}(x, y, z, t)$ is approximated by a series of the form

$$\mathbf{v}(x, y, z, t) = \sum_n \sum_m \sum_l \mathbf{a}(n, m, l; t)\, f_n(x)\, g_m(y)\, h_l(z), \tag{2.1}$$

evaluated at the coordinate position (x, y, z). The series (2.1) is written in a standard format where $f_n(x)$, $g_m(y)$, and $h_l(z)$ are the basis functions and $\mathbf{a}(n, m, l, t)$ are the appropriate coefficients. The choice of the basis functions and the corresponding coefficients depends on the velocity evaluation method. The standard format (2.1) permits an easier theoretical comparison of these methods. The actual numerical implementation specific to each method will differ from this and is chosen for computational efficiency. Each of the methods is elaborated in the rest of this section.

2.1. *Direct Summation*

In this method the local fluid velocity is evaluated by directly summing the spectral coefficients over all wavenumbers as shown below

$$\mathbf{v}(x, y, z, t) = \sum_{k_1} \sum_{k_2} \sum_{k_3} \hat{\mathbf{u}}(k_1, k_2, k_3, t)\, e^{ik_1 x} e^{ik_2 y} e^{ik_3 z}. \tag{2.2}$$

Here the basis functions are sine and cosine functions and the corresponding coefficients are the spectral coefficients. Equation (2.2) is simply a restatement of Eq. (1.1) and there is no error involved in the velocity evaluation, so that for direct summation,

$$\mathbf{v}(x, y, z, t) = \mathbf{u}(x, y, z, t). \tag{2.3}$$

To evaluate the sum (2.2) for the three components of velocity and for M fluid particles, the number of floating point operations required is of the order $3MN^3$. In addition some $3MN$ operations are required to evaluate the sine and cosine functions. In a vector computer the problem can be vectorized either over the number of particles, giving vectors of length M, or over a wave number, giving vectors of length N. Let us denote the former as method DSA and the latter as method DSB. When the number of fluid particles is greater than N, method DSA will be computationally faster than method DSB.

2.2. *Lagrangian Interpolation*

In this methods, denoted as LGI, a sixth-order, 3-dimensional (3D) Lagrangian interpolation scheme is used and the velocity approximated by

$$\mathbf{v}(x, y, z, t) = \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{l=0}^{N} \mathbf{u}(x_i, y_j, z_l, t)\, L_i(x)\, L_j(y)\, L_l(z), \tag{2.4}$$

where x_i, y_j, z_l are the grid points specified by (1.4). The basis functions L_i, L_j, L_l are given in the Appendix and are the standard functions for a six-point Lagrangian scheme. In a one-dimensional (1D) context this interpolation scheme approximates the value of some function $f(x)$ by the series

$$f(x) \approx \phi(x) = \sum_{i=0}^{n} f(x_i)\, L_i(x).$$

The scheme uses the function values at the three grid points lying either side of the particle position to fit a polynomial of degree five through these six nodal values. The interpolation scheme is exact at the nodes. Asymptotically as the spacing, h between the grid points goes to zero, the error of the approximation decreases as $O(h^6)$ [8].

This scheme is extended to three dimensions by (2.4). Equation (2.4) represents a global Lagrangian interpolation scheme, whereas in the actual implementation of the scheme, a careful look at the basis function L_i, L_j, L_l reveals that only the velocity values at $6 \times 6 \times 6$ grid points, arranged on a cubic lattice with the center cube containing the particle, are needed to fit the multidimensional polynomial. The periodic boundary conditions on the velocity field allow the velocity values at the grid points to be extended beyond the original box of side L. Again the interpolation scheme is exact at the nodes. Since for each particle 6^3 velocity values are needed, the evaluation of the sum for the local fluid velocity for M particles requires

of the order of $3 \times 6^3 \times M$ operations. Additionally $3 \times 6^2 \times M$ operations are required to evaluate the corresponding polynomials $L_i(x)$, $L_j(y)$, and $L_l(z)$. In our program written for a Cyber 205, the 6^3 velocity values for all the particles are gathered to contiguous memory locations and then vectorization is performed over the number of particles.

2.3. *Partial Hermite Interpolation*

This scheme combines cubic Hermite interpolation in the two directions y and z with direct summation in the final direction x. In a 1-dimensional context cubic Hermite interpolation approximates the value of some function $f(x)$ by the series

$$f(x) \approx \phi(x) = \sum_{i=0}^{N} [f(x_i)\, H_i(x) + f'(x_i)\, G_i(x)],$$

where H_i and G_i are the corresponding basis functions given in the Appendix. For a point x in the interval (x_i, x_{i+1}) the scheme approximates $f(x)$ by a cubic polynomial which is specified by the values of the function and its derivative $f'(x)$ at the two endpoints. Over the whole range x_0 to x_n the approximation $\phi(x)$ is continuously differentiable and is exact for both the function values and its derivatives at the grid points. Asymptotically as the spacing, h between grid points goes to zero the error of the approximation decreases as $O(h^4)$.

Hermite interpolation may be continued to higher dimensions. Here we apply it to two dimensions for the y, z-directions and combine this with direct summation in the x-direction. This scheme is denoted as PHI. The approximation scheme is specified as

$$\begin{aligned}\mathbf{v}(x, y, z, t) = \sum_{k_1} \sum_{j=0}^{N} \sum_{l=0}^{N} e^{ik_1 x} [&\mathbf{a}(k_1, j, l, t)\, H_j(y)\, H_l(z) \\ &+ \mathbf{b}(k_1, j, l, t)\, G_j(y)\, H_l(z) + \mathbf{c}(k_1, j, l, t)\, H_j(y)\, G_l(z) \\ &+ \mathbf{d}(k_1, j, l, t)\, G_j(y)\, G_l(z)].\end{aligned} \tag{2.5}$$

The coefficients **a**, **b**, **c**, **d** are specified in terms of the partial inverse transform of $\hat{\mathbf{u}}(k_1, k_2, k_3, t)$,

$$\tilde{\mathbf{u}}(k_1, y, z, t) = \sum_{k_2} \sum_{k_3} \hat{\mathbf{u}}(k_1, k_2, k_3, t) \exp(ik_2 y + ik_3 z), \tag{2.6}$$

inverted with respect to k_2 and k_3. The coefficient **a** is then given by the values of $\tilde{\mathbf{u}}$ at the grid points (y_j, z_l):

$$\mathbf{a}(k_1, j, l, t) = \tilde{\mathbf{u}}(k_1, y_j, z_l, t). \tag{2.7}$$

The coefficients **b** and **c** are given by the first-order spatial derivatives of $\tilde{\mathbf{u}}$ evaluated at the grid points (y_j, z_l)

$$\mathbf{b}(k_1, j, l, t) = \partial\tilde{\mathbf{u}}/\partial y, \tag{2.8}$$

$$\mathbf{c}(k_1, j, l, t) = \partial\tilde{\mathbf{u}}/\partial z. \tag{2.9}$$

Finally the coefficient **d** is given by the second-order spatial derivative of $\tilde{u}$ evaluated at the grid points (y_j, z_l)

$$\mathbf{d}(k_1, j, l, t) = \partial^2\mathbf{u}/\partial y\,\partial z. \tag{2.10}$$

The numerical implementation of this method requires three separate steps.

Step 1. Obtain the coefficients $\mathbf{a}(k_1, j, l, t)$, $\mathbf{b}(k_1, j, l, t)$, $\mathbf{c}(k_1, j, l, t)$, and $\mathbf{d}(k_1, j, l, t)$ from the spectral velocity coefficients $\tilde{\mathbf{u}}(\mathbf{k}, t)$, using FFTs and following the stages shown in Fig. 1. Of the six operations involved in this step, the two

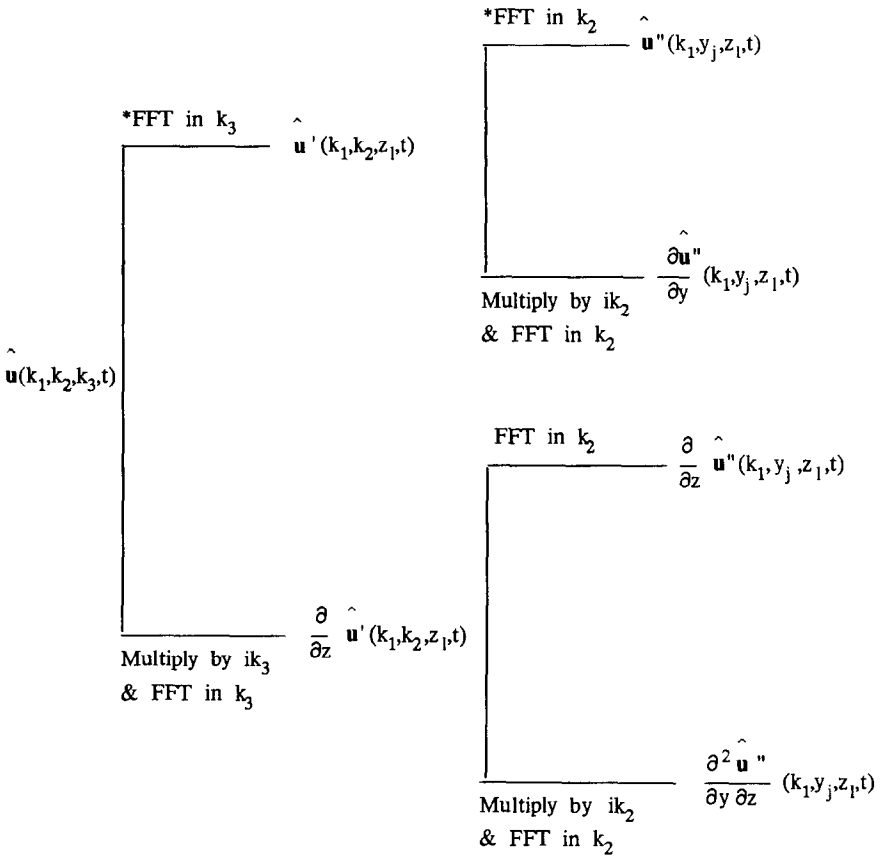

FIG. 1. First of the three steps involved in the implementation of partial Hermite interpolation.

operations, marked by asterisks, are in general part of a normal time-step for the turbulence computation and do not involve extra computational effort.

Step 2. For each particle, identify the square, in the $y-z$ plane, containing the particle coordinates (y, z) such that

$$y_J \leqslant y \leqslant y_{J+1}, \qquad z_L \leqslant z \leqslant z_{L+1},$$

and gather together the corresponding coefficients **a**, **b**, **c**, **d** for the four corners at $j=J,\ J+1$ and $l=L,\ L+1$. With these coefficients perform the Hermite interpolation, evaluating

$$\begin{aligned}\hat{\mathbf{v}}(k_1, y, z, t) = \sum_{j=J}^{J+1} \sum_{l=L}^{L+1} [&\mathbf{a}(k_1, j, l, t)\, H_j(y) H_l(z) \\ &+ \mathbf{b}(k_1, j, l, t)\, G_j(y)\, H_l(z) + \mathbf{c}(k_1, j, l, t)\, H_j(y)\, G_l(z) \\ &+ \mathbf{d}(k_1, j, l, t)\, G_j(y)\, G_l(z)].\end{aligned} \tag{2.11}$$

The evaluation of (2.11) involves 48 scalar coefficients for each particle and for each wavenumber k_1. To use the vector capabilities of the Cyber 205, these 48 coefficients for all the particles are gathered to contiguous memory locations to form 48 arrays of length M. Interpolation is then carried out as a vector operation. This computation is repeated for all values of k_1 given by (1.2).

Step 3. In this final step, the particle velocity is evaluated by a direct summation over k_1

$$\mathbf{v}(x, y, z, t) = \sum_{k_1} \hat{\mathbf{v}}(k_1, y, z, t)\, \exp(ik_1 x). \tag{2.12}$$

Step 1 represents a fixed overhead, independent of the number of particles, and involves $12N^2$ additional 1-dimensional FFTs, each of length N. Step 2 requires approximately $48MN$ operations and Step 3 approximately $3MN$ operations. In addition, allowance should be made for the operations to evaluate the polynomial terms in (2.11) and the sine and cosines in (2.12). A fully 3-dimensional Hermite interpolation scheme is possible but involves twice as many FFTs and is generally too costly. Further, the direct summation over k_1 of Step 3 involves only a summation of a half complex series. Since the velocity field **u** is real valued and satisfies the condition

$$\hat{\mathbf{u}}^*(\mathbf{k}, t) = \hat{\mathbf{u}}(-\mathbf{k}, t),$$

for the complex conjugate $\hat{\mathbf{u}}^*$ of the spectral coefficients, it suffices to save only the spectral coefficients for $k_1 \geqslant 0$. Hence a direct summation over k_1 requires half the computational effort of that needed for direct summation over k_2 or k_3.

2.4. *Linear Interpolation*

This method, denoted by LNI, is simply a 3-dimensional, second-order accurate, Lagrangian interpolation scheme and the approximation $\mathbf{v}$ for the velocity field is obtained by evaluating

$$\mathbf{v}(x, y, z, t) = \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{l=0}^{N} \mathbf{u}(x_i, y_j, z_l, t)\, P_i(x)\, P_j(y)\, P_l(z), \tag{2.13}$$

where the coefficients are the velocity values at the grid points. The basis functions P_i, P_j, P_l are given in the Appendix. These are linear functions of position and $P_i(x)$ is zero for x lying outside the interval (x_{i-1}, x_{i+1}). In this scheme, velocity data at the corners of the cube immediately surrounding the particle are used in evaluating the particle velocity. For M particles approximately $24M$ operations are required.

2.5. *Shape Function Method*

In the shape function method (SFM), the velocity data $\mathbf{u}(\mathbf{x}, t)$ and its spatial derivatives $\partial\mathbf{u}/\partial x$, $\partial\mathbf{u}/\partial y$, and $\partial\mathbf{u}/\partial z$ at the eight corners of the cube immediately surrounding the particle are used in evaluating the local fluid velocity [13]. The particle velocity is computed as

$$\begin{aligned}\mathbf{v}(x, y, z, t) = \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{l=0}^{N} \Bigg[& \mathbf{u}(x_i, y_j, z_l, t)\, H_i(x)\, H_j(y)\, H_l(z) \\ & + \frac{\partial \mathbf{u}}{\partial x}(x_i, y_j, z_l, t)\, G_i(x)\, H_j(y)\, H_l(z) \\ & + \frac{\partial \mathbf{u}}{\partial y}(x_i, y_j, z_l, t)\, H_i(x)\, G_j(y)\, H_l(z) \\ & + \frac{\partial \mathbf{u}}{\partial z}(x_i, y_j, z_l, t)\, H_i(x)\, H_j(y)\, G_l(z) \Bigg],\end{aligned} \tag{2.14}$$

where H and G are the basis functions, also called shape functions, given previously in the Appendix. In one dimension, the SFM is the same as the cubic Hermite interpolation discussed previously (PHI). But in three dimensions the SFM is less accurate. This is because the 3D Hermite interpolation also requires the evaluation of the second derivatives such as $\partial^2\mathbf{u}/\partial x\, \partial y$, $\partial^2\mathbf{u}/\partial y\, \partial z$, and $\partial^2\mathbf{u}/\partial z\, \partial x$ and the third derivative $\partial^3\mathbf{u}/\partial x\, \partial y\, \partial z$ at the eight grid points. The PHI method is more accurate than the 3D Hermite interpolation, so it is certainly more accurate than SFM. The actual implementation of the shape function method, globally represented by Eq. (2.14), requires the values of the three components of velocity and their first-order spatial derivatives at the eight corners of the cube containing the particle, 96 scalars in total. The evaluation of the spatial derivatives involves nine additional

3-dimensional FFTs each requiring $N^3 \log N$ operations and the SFM method then requires approximately $96M$ operations for the evaluation of the sum (2.14) for M particles.

2.6. *Other Methods*

Other methods of approximation are possible besides those mentioned above. One other scheme in particular that will be referred to is the method described by Yeung and Pope [11], designated by them as TS13. This is a third-order accurate, 13-point, interpolation scheme that uses not only the velocity values at the grid points specified by (1.4), but also the values on a staggered grid of N^3 points offset by half a mesh spacing in each direction. Thus each node on the staggered grid lies at the center of a cubic element formed by the original grid. Interpolation for a particle lying in a specific cubic element uses the values at the eight corners of that cube, the value at the center of the cube, and four of the six nearest neighboring points lying at the center of the adjoining cubes. Full details are given in the reference. This scheme is suited to simulations of homogeneous turbulence that incorporate de-aliasing procedures (Patterson and Orszag [7]) and that make use of such a staggered grid.

3. Numerical Tests for Accuracy

The accuracy of an interpolation scheme depends on the function that is being interpolated. In a spectral simulation of homogeneous turbulence, velocities are represented by a Fourier sum of sines and cosines. If the simulation is carried out in a box of side 2π, the wavenumbers of these Fourier modes are from (1.2)

$$k_i = 0, \ \pm 1, \ \pm 2, \ ..., \pm N/2, \qquad i = 1, 2, 3.$$

The magnitude of a wavenumber vector with components k_i is

$$k = |\mathbf{k}| = (k_1^2 + k_2^2 + k_3^2)^{1/2}.$$

In general, any interpolation scheme interpolates the low wavenumber modes more accurately than the higher wavenumbers. For a given flow field, two main ingredients determine the overall accuracy of a velocity interpolation scheme. First there is the energy spectrum $E(k)$ of the velocity field. For isotropic turbulence this depends only on $|\mathbf{k}|$ and is defined such that

$$\tfrac{3}{2}\langle u_1^2 \rangle = \int_0^\infty E(k)\, dk, \tag{3.1}$$

where $\langle u_1^2 \rangle$ is the ensemble-averaged, mean square fluctuation of u_1. Equation (3.1) may also be regarded as the turbulence kinetic energy per unit mass. Second, there is the accuracy with which the scheme interpolates various modes. While the

first factor gives the relative importance of a mode, the second determines how well that mode is interpolated.

To compare the accuracy of different interpolation schemes some quantitative measure is required. Qualitatively the convergence properties of these schemes are known. For example, in terms of the interval h between grid points the truncation error for the LGI scheme decreases asymptotically as $O(h^6)$. However, in applications it is important to know the coefficient of proportionality and at what value of h such asymptotic behavior may be obtained. For a fixed grid resolution, h, it is possible that a lower order scheme may give a smaller truncation error than that for some higher order scheme and that the asymptotic range of covergence may apply at a coarser grid resolution.

3.1. *Single Mode Accuracy*

The accuracy of an interpolation scheme for a mode of wavenumber $\mathbf{k}$ here is measured in terms of the root mean square error (RMSE) for the interpolation of $\sin(\mathbf{k}\cdot\mathbf{x})$ averaged over all points x in the box. In 1D this RMSE may be written as

$$\mathrm{Er}=\left[\frac{1}{2\pi}\int_0^{2\pi}[\sin(k_1x)-v(x;k_1)]^2\,dx\right]^{1/2}, \tag{3.2}$$

where $v(x;k_1)$ is the interpolation function approximating $\sin(k_1x)$ over the interval $0\leqslant x\leqslant 2\pi$. Similarly in 2D this RMSE can be expressed as

$$\mathrm{Er}=\left[1/(2\pi)^2\int_0^{2\pi}dx\int_0^{2\pi}dy\,[\sin(k_1x+k_2y)-v(x,y;k_1,k_2)]^2\right]^{1/2}, \tag{3.3}$$

where $v(x,y;k_1,k_2)$ is the approximation for $\sin(k_1x+k_2y)$; and correspondingly in 3D, where $v(x,y,z;k_1,k_2,k_3)$ is the approximation for $\sin(k_1x+k_2y+k_3z)$,

$$\mathrm{Er}=\left[1/(2\pi)^3\int_0^{2\pi}dx\int_0^{2\pi}dy\int_0^{2\pi}dz\,[\sin(k_1x+k_2y+k_3z)-v(x,y,z;k_1,k_2,k_3)]^2\right]^{1/2}. \tag{3.4}$$

For selected wavenumber vectors $\mathbf{k}$ these expressions for the RMSE have been evaluated for the various approximation schemes described in Section 2, using the appropriate basis functions given in the Appendix. These results were obtained with the aid of the MACSYMA program for algebraic manipulations.

The results for the RMSE of the various schemes are given in terms of the non-dimensional magnitude $\hat{k}$ of the wavenumber vector $\mathbf{k}$. The scaling for $\mathbf{k}$ is defined by comparing the wavelength of the mode being interpolated to the spacing between the grid points $h=L/N$, thus

$$\hat{k}=\frac{1}{2\pi}|k|\,L/N. \tag{3.5}$$

For 1D the value of $\hat{k}$ varies between 0 and $\frac{1}{2}$. In Fig. 2 the RMSE is plotted against $1/\hat{k}$ for 1D, 2D, and 3D linear interpolation schemes. The RMSE shows little variation with the number of dimensions when plotted in this way. For low values of $\hat{k}$, the RMSE tends to zero as $\hat{k}$ decreases and conforms to the usual asymptotic rule for the truncation error

$$\mathrm{Er} \propto \hat{k}^2.$$

As the wavenumber k increases the RMSE reaches a maximum value of $\mathrm{Er} = 2^{-1/2}$, and this occurs for the maximum admissible value of $\hat{k}$, namely $\hat{k} = \frac{1}{2}$, $2^{1/2}/2$, and $3^{1/2}/2$ in 1D, 2D, and 3D, respectively. In these limiting cases the components of $\mathbf{k}$ are all equal to their maximum values $N/2$, and at every grid point $\sin(\mathbf{k} \cdot \mathbf{x})$ is zero. Thus the interpolating function $v(\mathbf{x}; \mathbf{k})$ is identically zero and Er is simply given by the mean-square value of $\sin(\mathbf{k} \cdot \mathbf{x})$. Further, for a fixed value of $\hat{k}$, the computed RMSE is least when the wavenumber components k_i are all equal and increases when they are unequal. This feature contributes to the undulations in Er, that may be noted in Fig. 2, as $\hat{k}$ becomes large.

Figure 3 shows the RMSE against $1/k$ for 1D and 2D Hermite interpolation, with the undulations smoothed out. As before, for $\hat{k}$ less than 0.2 both these curves become straight lines in the log-log plot, indicating asymptotic convergence. Though the slope of both lines are the same, they are well separated, unlike the case of linear interpolation (LNI). Similar behavior is expected for three dimensions. The RMSE for 2D Hermite interpolation may be used as a guide to the accuracy of the PHI scheme, which combines 2D Hermite interpolation and direct summa-

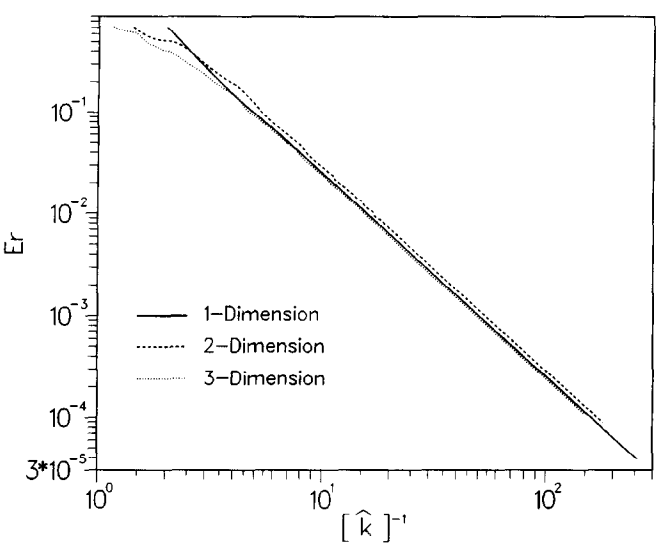

FIG. 2. Root mean square error against nondimensional wavenumber vector for linear interpolation schemes.

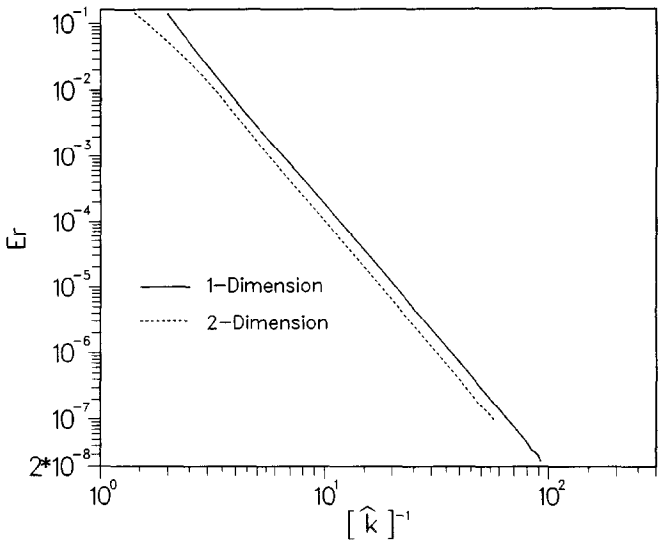

FIG. 3. Root mean square error against nondimensional wavenumber vector for Hermite interpolation schemes.

tion. Let x be the direction of direct summation in PHI, then since direct summation does not introduce any error for given values of k_2 and k_3 the RMSE does not increase as k_1 increases. For a constant value of $\hat{k}$ the error will be a maximum when $k_1 = 0$ and will reach its minimum when $k_2 = k_3 = 0$. The maximum error is the same as that for 2D Hermite interpolation and the minimum error is equal to zero. Hence the error remains the same over a range of $\hat{k}$ and is determined only

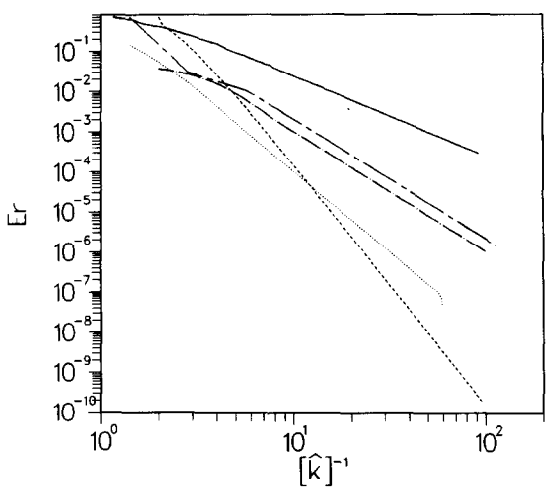

FIG. 4. Root mean square error against nondimensional wavenumber vector for the four different schemes. ———: 3D Linear interpolation; – – –: 1D Lagrangian; ···: 2D Hermite interpolation; — — — — — —: maximum error for TS13; ——·——·——: minimum error for TS13.

by k_2 and k_3. Overall, for any value of $\hat{k}$, the error in PHI will be less than the corresponding RMSE for 2D Hermite interpolation.

Figure 4 shows the RMSE for four different schemes: 3D linear interpolation, 2D Hermite interpolation, sixth-order Lagrangian interpolation, and the 3-dimensional method TS13 due to Yeung and Pope [11]. As discussed above, the RMSE for PHI will be less than that for 2D Hermite interpolation, so this figure shows a proper comparison for the accuracy of these methods. The evaluation of the RMSE for the LGI method proved too complicated, even with the aid of MACSYMA, in three dimensions and so only the RMSE for one dimension (1D) is shown. Based on the results for linear interpolation shown in Fig. 2 there should be little difference between 1D and 3D.

For small values of $\hat{k}$, the error for all four methods decreases with decreasing wavenumber $\hat{k}$ and the RMSE matches the expected asymptotic convergence relations,

$$\text{3D—Linear (LNI):} \qquad \text{Er} \propto \hat{k}^2, \tag{3.6a}$$

$$\text{2D—Hermite:} \qquad \text{Er} \propto \hat{k}^4, \tag{3.6b}$$

$$\text{1D—Lagrangian(LGI):} \qquad \text{Er} \propto \hat{k}^6, \tag{3.6c}$$

$$\text{3D—TS13:} \qquad \text{Er} \propto \hat{k}^3. \tag{3.6d}$$

Linear interpolation is the least accurate of the methods. In three dimensions the LGI method will have its maximum error at $\hat{k} = 3^{1/2}/2$, the same as for LNI, and approach $3^{1/2}/2$, the error estimate for LGI in 1D for small values of $\hat{k}$. For all values of $\hat{k}$ the LGI method will be more accurate than LNI in 3D. For the TS13 method two error estimates are given since this method treats the y direction differently from the x or z direction. The upper curve represents the maximum error that occurs when $k_2 = 0$ and the lower curve represents the minimum error that occurs when $k_1 = k_3 = 0$. For any other combination of k_1, k_2, and k_3 the error lies within this band. It is also interesting to note that for large values of $\hat{k}$ the error involved in 2D Hermite interpolation is substantially less than those of the other three methods. This is because, Hermite interpolation makes use of the velocity derivatives in addition to velocity values. For large values of $\hat{k}$ 2D Hermite interpolation is more accurate than the LGI method but as $\hat{k}$ decreases the LGI method becomes more accurate. In this range though the errors are quite small in absolute terms, and for a flow simulation the truncation errors of the simulation may be more significant.

3.2. *Spectrum of Modes*

The second factor governing the accuracy of an interpolation scheme is the relative distribution of energy between the individual modes. Some estimate of this effect may be obtained by weighting the error estimate for the individual modes,

found above, by a typical energy spectrum $E(k)$ for homogeneous isotropic turbulence. The normalized error, in a loose average sense, is defined then as

$$\frac{\int_0^{k_{\max}} E(k)\,\mathrm{Er}(\hat{k})\,dk}{\int_0^{k_{\max}} E(k)\,dk}. \tag{3.7}$$

It is assumed as before that the box length L is equal to 2π and the simulation is for a cubic array of N^3 grid points. The wavevector components k_i are given by (1.2) and take integer values between $-N/2$ and $N/2$, the value of $\hat{k}$ is related to k by (3.5). There is a limit to the smallest turbulence scale that can be accurately resolved by the simulation and this sets an upper limit, $k_{\max}$ to the magnitude of the largest resolvable wavenumber. Following Patterson and Orszag [7], $k_{\max}$ is taken to be

$$k_{\max} = (2 \times 2^{1/2}/3) \times N/2.$$

The particular choice of energy spectrum is taken to be that proposed by Pao [6]

$$\begin{aligned} E(k) &= k^{-5/3} \exp[-1.5\alpha(k\eta)^{4/3}], \qquad k < k_{\max}, \\ &= 0, \qquad\qquad\qquad\qquad\quad k \geqslant k_{\max}, \end{aligned} \tag{3.8}$$

where $\alpha = 2.45$ and η is the Kolmogorov microscale.

Figure 5 shows how the relative error defined by (3.7) decreases as the number of grid points in each direction, and hence the grid resolution, increases. The value of η is set to be $\eta = \frac{1}{15}$. The figure gives a comparison between the accuracy of the

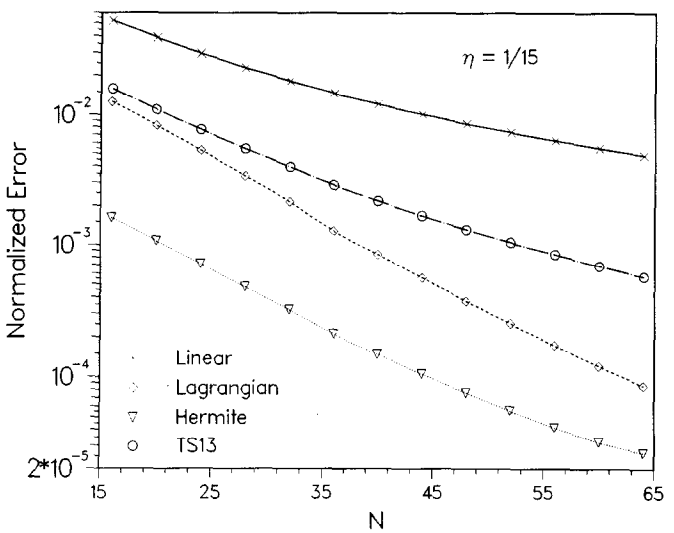

FIG. 5. Effect of increasing the number of grid points on the normalized error for a fixed turbulence energy spectrum ($\eta = \frac{1}{15}$).

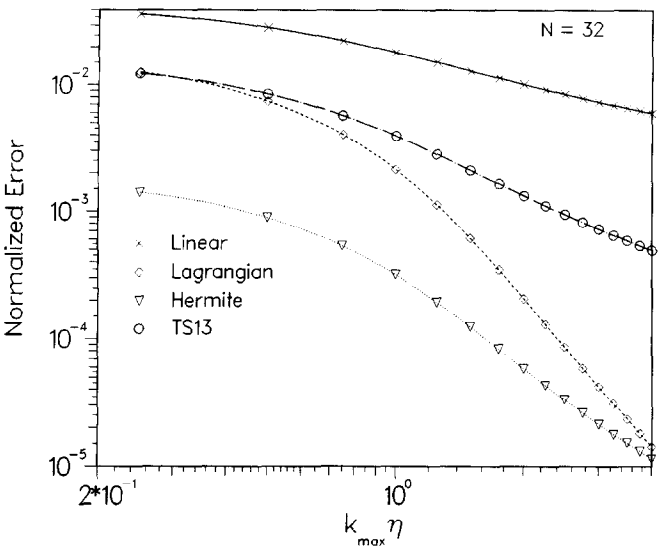

FIG. 6. Effect of spatial resolution on the normalized error for a $(32)^3$ simulation.

different schemes; and over the range shown the 2D Hermite interpolation is clearly the most accurate. Since the maximum error involved in PHI is equal to the error involved in 2D Hermite interpolation, the PHI scheme will be even more accurate. Similar comparisons can be made for other values of η.

Generally in turbulence simulations, as N is increased the turbulence Reynolds number is also increased and the Kolmogorov microscale decreases. The spatial resolution of the grid should then be compared with the value of η for any fixed number of grid points. Following Yeung and Pope [11, Fig. 3] we show in Fig. 6 the normalized error for different values of $k_{max}\eta$. The value of $k_{max}\eta$ should be about one or larger for adequate resolution of the small-scale turbulence dynamics. The range of values of $k_{max}\eta$ shown in Fig. 6 lie between 0.2 and 3.0, though values appreciably less than 1.0 would not be encountered in a properly resolved simulation. Again the value of this diagram is as a comparison of the various schemes. [redacted] LGI scheme is faster as the value of η increases and, by implication, the Reynolds number is decreased.

4. NUMERICAL EXPERIMENTS

The preceding error estimates give a measure of the accuracy of these different interpolation schemes for evaluating the Lagrangian fluid velocity. From the point of view of applications though, it is important to see how well they perform in typical numerical experiments. For this purpose we have carried out direct

numerical simulations of decaying, homogeneous, isotropic turbulence at two different Reynolds numbers. Particles moving as Lagrangian fluid tracers, according to (1.3), were introduced into the flow field and various velocity statistics were obtained. Specifically, error estimates were determined for the absolute velocity of a single particle, the relative velocity of two particles, and a comparison made between the various schemes in determining the dispersion statistics for single particle turbulent diffusion.

The flow fields for decaying, homogeneous turbulence were generated by numerically solving the Navier–Stokes equations for incompressible flow with a pseudo-spectral method similar to that described by Riley and Patterson [9]. A mesh of $32 \times 32 \times 32$ grid points was used to represent the flow, in a box of side 2π and with periodic boundary conditions imposed on the flow field. The flow $\mathbf{u}(\mathbf{x}, t)$ is specified in spectral form by a Fourier series (1.1), and the flow evolves according to the Navier–Stokes equations written in rotational form

$$\partial \mathbf{u}/\partial t = \mathbf{u} \times \boldsymbol{\omega} - \nabla(p/\rho + \tfrac{1}{2}\mathbf{u}^2) + \nu \nabla^2 \mathbf{u}, \tag{4.1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{4.2}$$

where $\boldsymbol{\omega}$ is the fluid vorticity, p the pressure, ρ the density, and ν the kinematic viscosity. The flow was computed using a leapfrog scheme for the nonlinear terms and a Crank–Nicholson scheme for the viscous terms. The flow was specified initially as a pseudo-random, incompressible flow field with Gaussian statistics and zero mean flow. Statistically the initial flow was homogeneous and isotropic, and the energy spectrum was chosen [4] to be

$$E(k) = 16(2/\pi)^{1/2}\, U_0^2 k^4 k_0^{-5} \exp(-2k^2/k_0^2), \tag{4.3}$$

where U_0^2 is the initial mean square value of the velocity fluctuation u_1.

In the experiments reported here the turbulent flow was computed from the initial flow for a fixed number of time steps till the velocity statistics, velocity derivative skewness, and other statistical quantities had reached a level characteristic of a developed turbulent flow. At this point the particles were introduced and this was then taken to be the reference time, $t = 0$. Two sets of initial conditions were specified. For both sets of simulations the box side L was 2π cm, k_0 was $2.378\ \mathrm{cm}^{-1}$, and the viscosity ν was $0.6\ \mathrm{cm^2 s^{-1}}$ in dimensional units. For the first experiment $U_0^2 = 240\ \mathrm{cm^2 s^{-2}}$, and the flow evolved for 75 time steps before the particles were introduced. For the second experiment $U_0^2 = 1200\ \mathrm{cm^2 s^{-2}}$ and the flow evolved 192 time steps before the particles were introduced. Figures 7 and 8 show the evolution in time of the Reynolds number, Re_λ based on the Taylor microscale, λ, and the evolution of λ for the two experiments. The quantities λ and Re_λ are defined by

$$\langle u_1^2 \rangle / \lambda^2 = \langle (\partial u_1/\partial x_1)^2 \rangle, \tag{4.4}$$

$$\mathrm{Re}_\lambda = \langle u_1^2 \rangle^{1/2} \lambda / \nu. \tag{4.5}$$

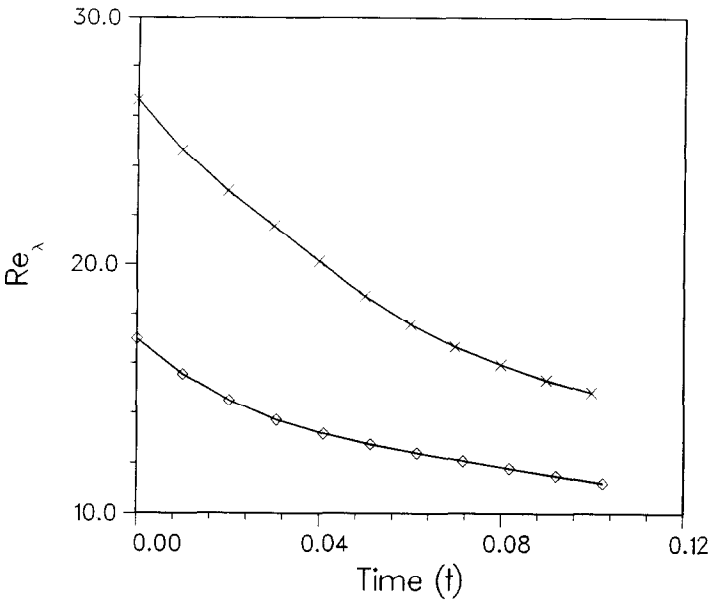

FIG. 7. Evolution of the microscale Reynolds number Re_λ, see (4.5), in a 32^3 grid point simulation of decaying, homogeneous, isotropic turbulence: ◇, first simulation with $Re_\lambda = 17$ at $t = 0$; ×, second simulation with $Re_\lambda = 26.5$ at $t = 0$.

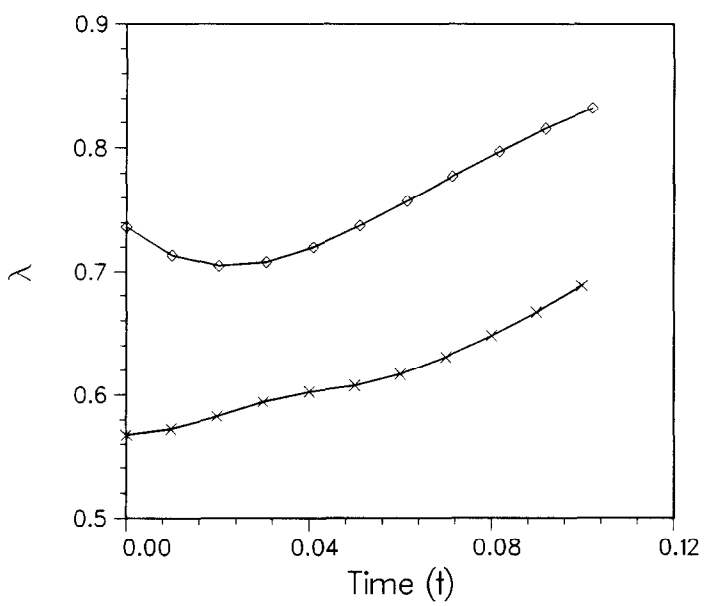

FIG. 8. Evolution of the Taylor microscale λ, see (4.4), in a 32^3 grid point simulation of decaying, homogeneous, isotropic turbulence: ◇, first simulation with $Re_\lambda = 17$ at $t = 0$; ×, second simulation with $Re_\lambda = 26.5$ at $t = 0$.

For both experiments the Taylor microscale grows in time, while the Reynolds number decreases from a value of $\mathrm{Re}_\lambda = 17$ at the reference time $t = 0$ for the first experiment, and from $\mathrm{Re}_\lambda = 26.5$ for the second experiment. In both experiments $k_{\max}\eta$ increases with time; for the lower Reynolds number case, $k_{\max}\eta$ increases from a value of 1.38, at $t = 0$, to its final value of 1.64 at $t = 0.1$; for the higher Reynolds number case, $k_{\max}\eta$ increases from 0.85 to 1.37.

4.1. *Absolute and Relative Velocity Measurements*

In the first test, the different interpolation schemes are used to evaluate the absolute velocity of a single particle and the relative velocity of two particles for a large number of particles, using the two turbulent flows described above. A thousand fluid particles in 125 groups of 8 each are introduced into the turbulent flow at $t = 0$. For each group the position of the first particle $J = 0$, is chosen at random within the box. The position of the Jth particle $J = 1, 2, ..., 7$, within the same group is chosen at random to lie on the surface of a sphere, centered on the first particle, with radius r_J. The values of r_J for these simulations are

$$r_1 = 0.01, \tag{4.6a}$$

$$r_2 = 0.05, \tag{4.6b}$$

$$r_3 = 0.1, \tag{4.6c}$$

$$r_4 = 0.2, \tag{4.6d}$$

$$r_5 = 0.5, \tag{4.6e}$$

$$r_6 = 1.0, \tag{4.6f}$$

$$r_7 = 2.0. \tag{4.6g}$$

The individual particle positions are labelled $\mathbf{X}(I, J; t)$ for the position of the Lagrangian fluid element at time t, where $I = 1, 2, ..., 125$ indicates the group and $J = 0, 1, ..., 7$ indicates the particle number within the group.

The local fluid velocity $\mathbf{u}(\mathbf{x} = \mathbf{X}(I, J; t), t)$, designated by $\mathbf{u}(I, J)$, is evaluated by direct summation (DS). The different interpolation schemes LGI, PHI, LNI, and SFM are used in turn to obtain approximations $\mathbf{v}(\mathbf{x} = \mathbf{X}(I, J; t), t)$, designated by $\mathbf{v}(I, J)$, to the local fluid velocity. The average fractional error in evaluating the absolute velocity of a single particle is then defined for each method as

$$\Delta u = \frac{\{\sum_{I=1}^{125} \sum_{J=0}^{7} |\mathbf{v}(I, J) - \mathbf{u}(I, J)|^2\}^{1/2}}{\{\sum_{I=1}^{125} \sum_{J=0}^{7} |\mathbf{u}(I, J)|^2\}^{1/2}}, \tag{4.7}$$

where the average is formed by summing over all the particles.

Similarly the average fractional error in evaluating the relative velocity of two particles, for each separation distance r_J, is defined as

$$\Delta u = \frac{\{\sum_{I=1}^{125} |(\mathbf{v}(I, J) - \mathbf{v}(I, 0)) - (\mathbf{u}(I, J) - \mathbf{u}(I, 0))|^2\}^{1/2}}{\{\sum_{I=0}^{125} |\mathbf{u}(I, J) - \mathbf{u}(I, 0)|^2\}^{1/2}}, \tag{4.8}$$

where $J = 1, 2, 3, ..., 7$.

Tables I and II summarize the results for the fractional error Δu, where the particles are introduced at $t = 0$ into the two different turbulent flows. The data is also plotted in Figs. 9 and 10 on a logarithmic scale. Table I summarizes the results for the first experiment at $\mathrm{Re}_\lambda = 17$. For all the interpolation schemes the error in evaluating the absolute particle velocity at this lower Reynolds number is quite small, 4% or less. The PHI scheme is the most accurate, the error being two orders of magnitude less than for LNI, while the shape function method (SFM) is the next most accurate. The errors, shown in Fig. 9, in evaluating the relative velocities are significant when the particle separation r_J is small, but for large values of the particle separation the errors in the relative velocities decrease and become comparable to the error involved in evaluating the absolute particle velocity. When the particle separation is smaller than the grid size, which in our case is equal to $2\pi/32 \approx 0.2$, the particle velocities are strongly correlated and the contribution to the relative velocity from high wavenumber components is high. Whereas, for a larger particle

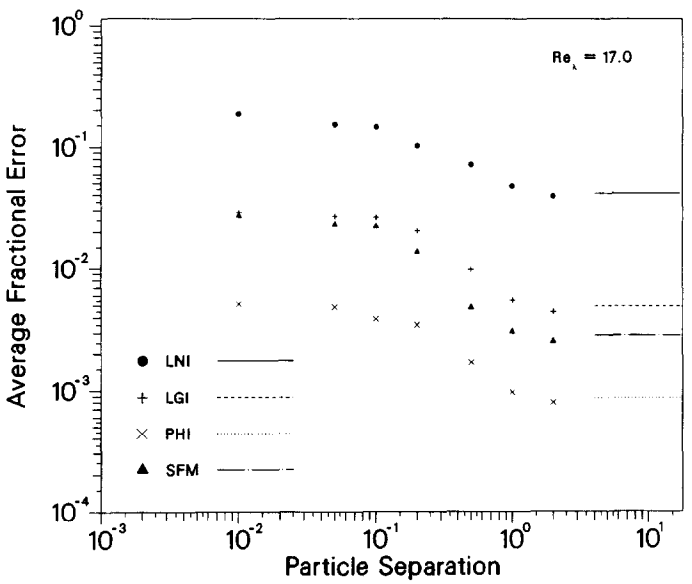

FIG. 9. Average fractional error in evaluating the relative velocity of two particles against particle separation distance for the first simulation with initial Reynolds number $\mathrm{Re}_\lambda = 17$. The lines at the right of the figure show the fractional error in evaluating the absolute velocity of a single particle.

TABLE I

Values of the Average Fractional Error Δu in Evaluating the Velocity for a Single Particle, see (4.7), and the Relative Velocity of Two Particles, see (4.8), for the Different Interpolation Schemes

Distance r_j / Methods	Single particle	0.01	0.05	0.1	0.2	0.5	1.0	2.0
LNI	4.11×10^{-2}	1.86×10^{-1}	1.52×10^{-1}	1.45×10^{-1}	1.02×10^{-1}	7.17×10^{-2}	4.73×10^{-2}	3.9×10^{-2}
LGI	4.91×10^{-3}	2.87×10^{-2}	2.66×10^{-2}	2.61×10^{-2}	2.03×10^{-2}	9.67×10^{-3}	5.48×10^{-3}	4.43×10^{-3}
PHI	8.65×10^{-4}	5.14×10^{-3}	4.85×10^{-3}	3.89×10^{-3}	3.48×10^{-3}	1.70×10^{-3}	9.71×10^{-4}	7.94×10^{-4}
SFM	2.86×10^{-3}	2.69×10^{-2}	2.28×10^{-2}	2.20×10^{-2}	1.37×10^{-2}	4.78×10^{-3}	3.05×10^{-3}	2.56×10^{-3}

Note. Reynolds number Re_λ is 17.

TABLE II

Values of the Average Fractional Error Δu in Evaluating the Velocity for a Single Particle, see (4.7), and the Relative Velocity of Two Particles, see (4.8), for the Different Interpolation Schemes

Distance r_j / Methods	Single particle	0.01	0.05	0.1	0.2	0.5	1.0	2.0
LNI	8.88×10^{-2}	4.12×10^{-1}	3.14×10^{-1}	3.16×10^{-1}	2.78×10^{-1}	1.62×10^{-1}	1.04×10^{-1}	9.43×10^{-2}
LGI	3.85×10^{-2}	1.86×10^{-1}	1.53×10^{-1}	1.64×10^{-1}	1.26×10^{-1}	6.68×10^{-2}	4.39×10^{-2}	4.04×10^{-2}
PHI	6.11×10^{-3}	3.03×10^{-2}	2.74×10^{-2}	2.55×10^{-2}	1.96×10^{-2}	1.06×10^{-2}	7.32×10^{-3}	6.47×10^{-3}
SFM	1.70×10^{-2}	1.02×10^{-1}	8.33×10^{-2}	7.96×10^{-2}	6.08×10^{-2}	2.88×10^{-2}	2.03×10^{-2}	1.82×10^{-2}

Note. Reynolds number Re_λ is 26.5.

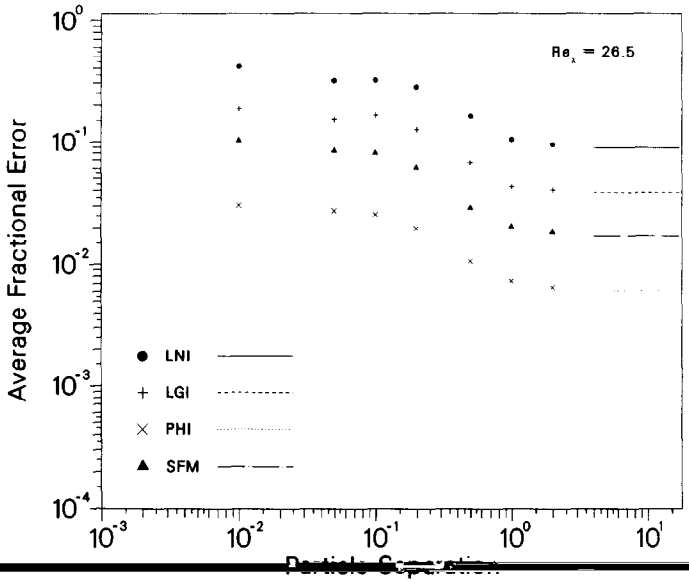

FIG. 10. As in Fig. 9, for second simulation with initial Reynolds number $Re_\lambda = 26.5$.

separation particle velocities are less correlated and the low wavenumber components dominate the relative velocity. For all four interpolation schemes that appear in Table I it can be clearly seen that for particle separations less than 0.2, the errors in the relative velocities are five to ten times larger than the corresponding single particle errors. With increasing particle separations the errors in the relative velocities decrease and for particle separations comparable to the box size the errors are approximately equal to the corresponding single particle errors. For small separations the linear interpolation scheme (LNI) is seriously in error, typically 20%, while the other schemes maintain reasonable accuracy. Again, PHI is the most accurate, with errors of $\frac{1}{2}$% or less, and SFM the next most accurate.

Table II summarizes the results for the second experiment at $Re_\lambda = 26.5$. The high wavenumber content of the energy spectrum in this simulation is much stronger, and the errors in evaluating fluid velocities are more significant. The errors for the absolute velocity range from 8% for LNI to 0.6% for the PHI scheme. Again, the PHI scheme is the most accurate. The error in the relative velocity shown in Fig. 10 for this higher Reynolds number case behaves similarly to the lower Reynolds number case shown above. As before the fractional errors in the relative velocities for small particle separations are much higher than the corresponding error in evaluating the absolute particle velocity and range from 41% for LNI to 3% for the PHI scheme. The sixth-order Lagrange interpolation scheme (LGI) for both experiments is not as accurate as PHI or SFM, but for the absolute velocity of a single particle gives acceptable results with errors of $\frac{1}{2}$ and 4%, respectively.

4.2. *Single Particle Dispersion*

In this second set of tests the different interpolation schemes are used to estimate the turbulent dispersion of individual Lagrangian tracers, for the two different turbulent flow simulations described earlier. At time $t = 0$, 250 particles are introduced into the flow at random locations. The trajectory $\mathbf{X}(t)$ of a fluid particle is determined by (1.1),

$$d\mathbf{X}/dt = \mathbf{u}(\mathbf{x} = \mathbf{X}(t), t), \tag{4.9}$$

subject to the initial condition

$$\mathbf{X}(t = 0) = \mathbf{X}_0. \tag{4.10}$$

Here we use $\mathbf{X}(t)$ to denote the particle trajectory computed by directly summing (DS) the Fourier series (1.1) to evaluate $\mathbf{u}(\mathbf{X}(t), t)$. For the different interpolation schemes we use $\mathbf{Y}(t)$ to denote the trajectory computed from the approximation $\mathbf{v}(\mathbf{x}, t)$ to the flow field $\mathbf{u}$, thus

$$d\mathbf{Y}/dt = \mathbf{v}(\mathbf{x} = \mathbf{Y}(t), t), \tag{4.11}$$

subject to the same initial condition

$$\mathbf{Y}(t = 0) = \mathbf{X}_0. \tag{4.12}$$

Equations (4.9) and (4.11) are solved numerically with a fourth-order, Adams–

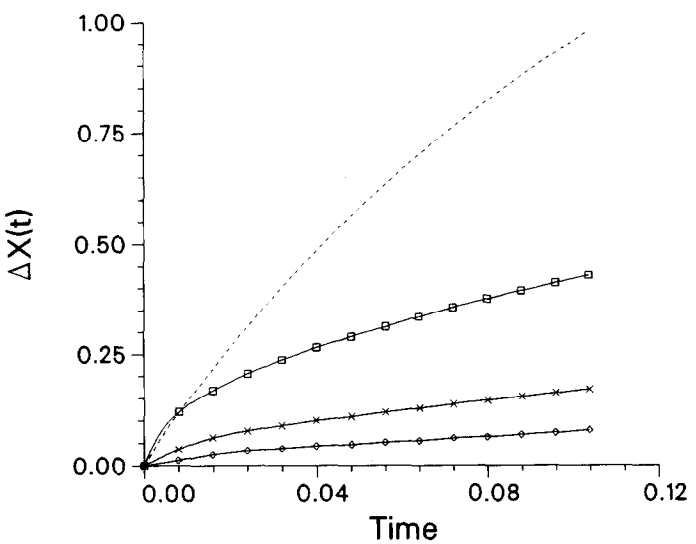

FIG. 11. Growth of average error in particle position against time for different interpolation schemes: □, LNI; ×, LGI; ◇, PHI. First simulation with initial Reynolds number $\mathrm{Re}_\lambda = 17$. The broken line shows the actual root mean-square particle dispersion for the same conditions.

Moulton, predictor-corrector method and a fourth-order Runge–Kutta method is used to initialize the scheme. These fourth-order schemes ensure negligible time stepping error compared to the error involved in the evaluation of $\mathbf{u}(\mathbf{X}(t), t)$.

The first test made is to compare the approximate trajectory $\mathbf{Y}(t)$ with the true trajectory $\mathbf{X}(t)$, and to see how rapidly the two diverge. The separation between the two trajectories is $(\mathbf{Y}^{(m)}(t) - \mathbf{X}^{(m)}(t))$ at any instant, for the mth particle, where $m = 1, 2, ..., M$. The total number of particles, M is here equal to 250. The separation distance is averaged over all the particles and we define

$$\Delta X(t) = \left\{ \frac{1}{M} \sum_{M=1}^{M} |\mathbf{Y}^{(m)}(t) - \mathbf{X}^{(m)}(t)|^2 \right\}^{1/2}. \tag{4.13}$$

Figures 11 and 12 show the growth of this separation distance Δx with time for the different interpolation schemes and for the two different Reynolds numbers. For comparison the corresponding root mean-square (rms) actual particle displacements for single particle dispersion are also plotted as broken lines in these figures. The divergence of the approximated trajectory from the true trajectory grows most rapidly for the linear interpolation scheme (LNI), and least rapidly for the PHI scheme. In Fig. 11 at time equal to 0.1, the root mean-square error in particle displacement is 44% of the actual rms displacement for LNI and the corresponding percentage errors for LGI and PHI are 17% and 8%. For the higher Reynolds number case the corresponding percentage errors in particle displacement are higher and are equal to 65, 43, and 16% respectively for LNI, LGI, and PHI. The PHI method is reasonably accurate for the full duration of the simulation.

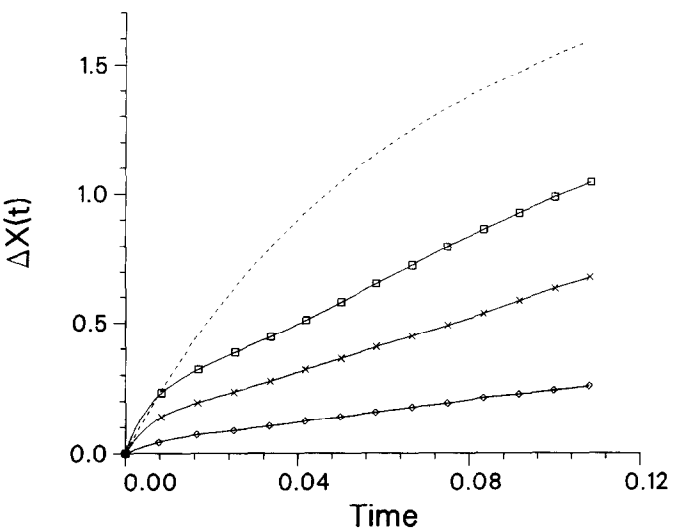

FIG. 12. As in Fig. 11, for second simulation with initial Reynolds number $\mathrm{Re}_\lambda = 26.5$.

The second test is to compare the Lagrangian statistics associated with turbulent dispersion [10]. First, the averaged Lagrangian velocity correlation is evaluated

$$R_L(t) = \frac{1}{3M} \sum_{m=1}^{M} [\mathbf{v}(\mathbf{x} = \mathbf{Y}^{(m)}(0), 0) \cdot \mathbf{v}(\mathbf{x} = \mathbf{Y}^{(m)}(t), t)], \tag{4.14}$$

where the average is over all the particles and over the three coordinate directions. The correlation $R_L(t)$ is determined for direct summation (DS) and the three approximate methods LGI, PHI, and LNI. Second, the effective eddy diffusivity $D(t)$ is determined as

$$D(t) = \frac{1}{3M} \sum_{m=1}^{M} [(\mathbf{Y}^{(m)}(t) - \mathbf{Y}^{(m)}(0)) \cdot \mathbf{v}(\mathbf{x} = \mathbf{Y}^{(m)}(t), t)], \tag{4.15}$$

where the average is over all the particles and the three coordinate directions. Third, the mean square particle dispersion is determined by

$$S(t)^2 = \frac{1}{3M} \sum_{m=1}^{M} |\mathbf{Y}^{(m)}(t) - \mathbf{Y}^{(m)}(0)|^2, \tag{4.16}$$

again averaging over all particles and coordinate orientations. These statistics of turbulent dispersion are related to each other, since

$$d/dt\,[S(t)^2] = 2D(t). \tag{4.17}$$

The results for these different Lagrangian statistics are shown in Fig. 13 for the first turbulence simulation at the lower Reynolds number, and in Fig. 14 for the second simulation at the higher Reynolds number. The results show that there is not a large difference between the results for the different methods, with the possible exception of the linear interpolation scheme, which shows the greatest difference at small time separations when the high wavenumber content of the spectrum is strongest. This is slightly more pronounced for the higher Reynolds number simulation. These results indicate that the single-particle, statistical averages are not sensitive to the approximation scheme used. This is a consistent with the results of Section 4.1, where the absolute velocity of a single particle was approximated reasonably well by most methods. Also the particle dispersion is determined by the Lagrangian velocity correlation $R_L(t)$, which at large time separations is dominated by the contribution of large scale, well correlated motions. This further reduces the impact of the high wavenumber content of the energy spectrum.

4.3. *Computational Speed*

So far the emphasis in comparing the different approximation methods has been on accuracy. In any practical situation the choice of a numerical interpolation scheme will be made on the basis of both accuracy and computational speed. To check the computational speed of the various methods a series of tests were made

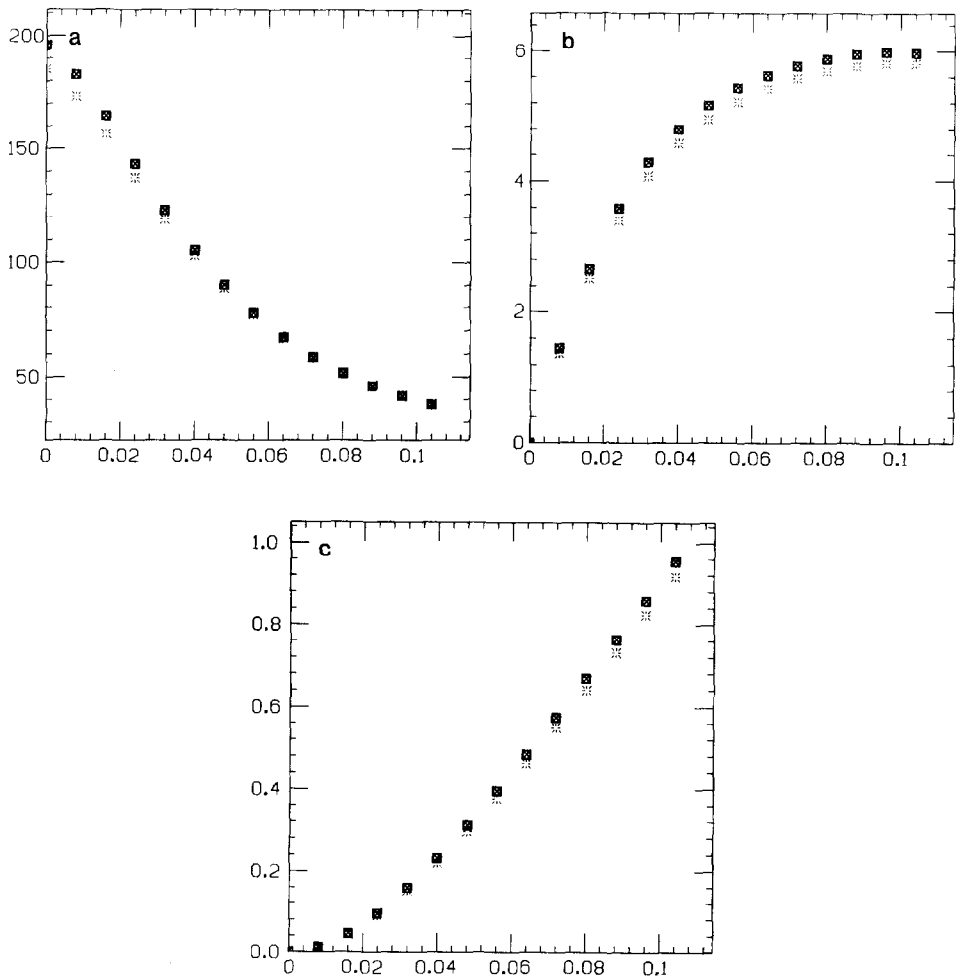

FIG. 13. (a) Lagrangian velocity correlation against time for the first turbulence simulation at the lower initial Reynolds number. ×: DS; ◇: LGI; □: PHI; *: LNI. (b) Effective eddy diffusivity against time for the first turbulence simulation at the lower initial Reynolds number. ×: DS; ◇: LGI; □: PHI; *: LNI. (c) Mean square particle dispersion against time for the first turbulence simulation at the lower initial Reynolds number. ×: DS; ◇: LGI; □: PHI; *: LNI.

to find the time taken, including any initial overhead for additional FFTs, to evaluate the fluid velocities for M particles, where M was taken to be 1, 10, 100, and 1000. The results of these tests are summarized in Table III. In a typical run, the computation for the time evolution of a $(32)^3$ homogeneous turbulent flow field requires 0.26 s of the computer time for each time step. From Table III, it is clear that for large values of M the computational time required for evaluating the fluid velocities can exceed the time required for the computation of the flow field.

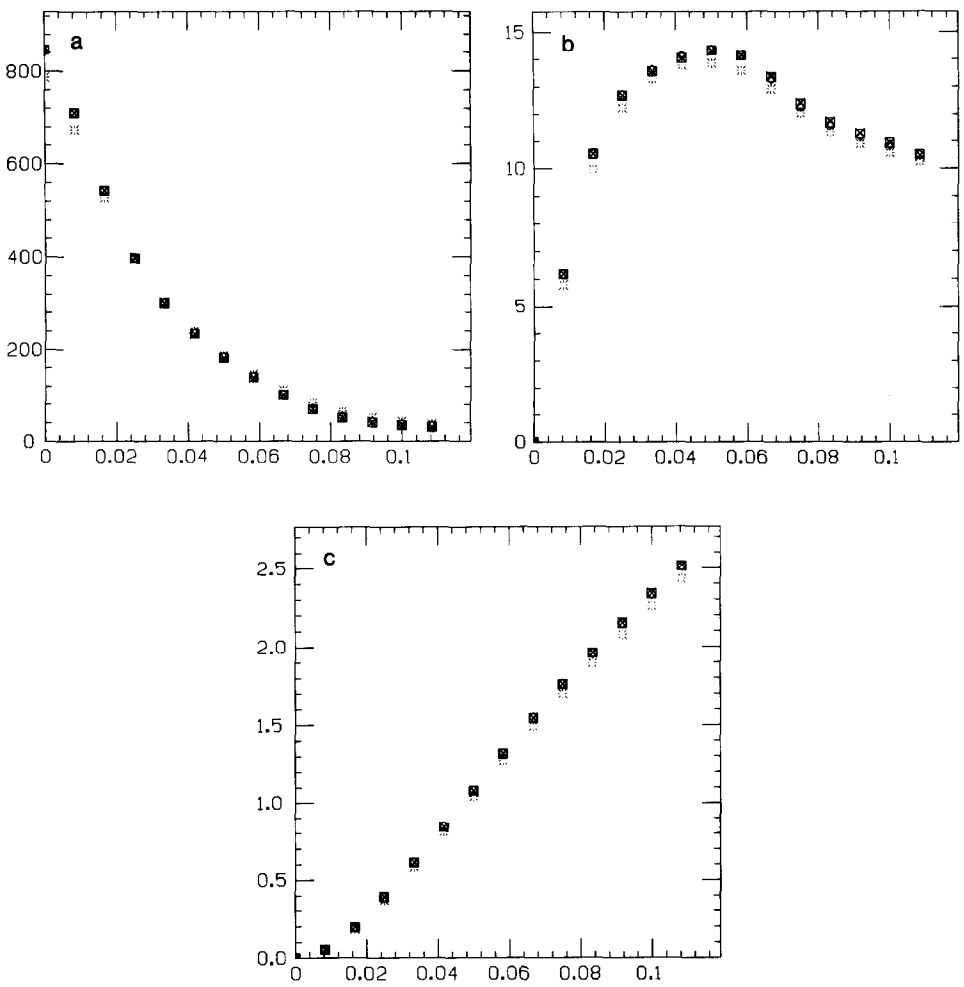

FIG. 14. (a) The same as Fig. 13a but for the second turbulence simulation at the higher initial Reynolds number. (b) The same as Fig. 13b but for the second turbulence simulation at the higher initial Reynolds number. (c) The same as Fig. 13c but for the second turbulence simulation at the higher initial Reynolds number.

Direct summation (DS) involves no overhead in computation, and where vectorization is over a wavenumber component (DSB), giving vectors of length N, the computation time increases linearly with the numbers of particles. Vectorization over the number of particles (DSA), giving vectors of length M, is more effective

overhead to perform the additional FFTs, while linear interpolation and the LGI scheme require no fixed overhead. The PHI scheme in particular involves a sub-

TABLE III

The Time Taken in Seconds to Compute the Particle Velocities by the Various Interpolation Schemes for a Grid of 32^3 Mesh Points

	DSB	DSA	LNI	LGI	PHI	SFM
$M = 1$	0.022	0.412	0.002	0.011	0.188	0.361
$M = 10$	0.220	0.443	0.004	0.009	0.197	0.386
$M = 100$	2.200	0.651	0.005	0.034	0.204	0.453
$M = 1000$	22.30	2.650	0.010	0.130	0.212	0.677

Note. The total number of particles, M varies between 1 and 1000.

stantial overhead in initial computations, but requires little additional time as the number of particles is increased. This scheme is quite effective for larger number of particles.

5. Conclusions

Overall accuracy and computational speed are the two most important factors in choosing a numerical interpolation scheme for evaluation of fluid velocities. Direct summation offers high accuracy but at low speeds while linear interpolation gives low accuracy at high speeds. The alternative schemes considered here, LGI, PHI, SHM, and the scheme TS13 due to Yeung and Pope [11] lie between these two extremes in both accuracy and speed. From the results of Section 4.2 on the Lagrangian statistics of turbulent diffusion it is clear that the results are very similar for all the interpolation schemes, except perhaps for LNI. This is partly due to the dominant contribution of large scale, low wavenumber motions to the dispersion process. So if one is only interested in one-particle Lagrangian statistics it is sufficient to settle for a less accurate but computationally faster scheme such as LGI. On the other hand, for experiments such as the simulation of particle coagulation, where close interaction of particles play a crucial role, the choice of the interpolation scheme is determined more by the need for accuracy.

For a turbulence simulation with only $(32)^3$ mesh points, the maximum Reynolds number Re_λ for which the flow is adequately resolved is generally accepted to be about 24 [3]. At this limiting value the energy in the high wavenumber components increases and so do the errors in the interpolation schemes. The results of Table II thus indicate "worst case" results. As noted in Section 4.1 the LGI scheme gives acceptable results if the Reynolds number is not too high, but at higher Reynolds numbers the slower, but more accurate PHI scheme is required. An alternative approach that may be adopted is to perform the interpolation on a finer mesh, say $(48)^3$ as opposed to $(32)^3$, even if the rest of the turbulence simulation is performed on the coarser mesh. The Fourier coefficients for the velocity field may be expanded from $(32)^3$ to $(48)^3$ by setting the additional Fourier coefficients to

zero and then using FFTs to give the velocities on the finer mesh of grid points. As a comparison for example, we may see from Fig. 5 that the normalized error for the LGI scheme at $N=49$ is the same as the normalized error for PHI when $N=32$. The final choice between these approaches will depend on the additional computational time required.

The choice of an interpolation scheme depends on what is being computed in the rest of the simulation. If the simulation includes a dealiassing scheme then an interpolation scheme such as TS13 will not involve any additional overhead, since the velocity values on the shifted grid will already be available. Similarly if the simulation requires the evaluation of velocity derivatives then schemes such as PHI or SHM involve little or no additional overhead in computation and become relatively fast. Factors such as storage requirements, available memory, and paging input/output will also affect the choice of scheme.

The preceding discussions have focused on methods of evaluating fluid velocities in homogeneous turbulence, where periodic boundary conditions are applied. This allows the flow field to be extended to points beyond the computational domain. In particular, for the higher order Lagrangian interpolation schemes such as LGI this allows fluid velocities to be evaluated at points close to the boundary while still keeping the interpolation grid centered on the particle position. In other problems where spectral simultations are used with no-slip or inhomogeneous boundary conditions this is not appropriate, and Lagrangian interpolation schemes are less accurate if the interpolation grid is not centered. The other schemes such as PHI, SFM, and direct summation (DS) are not restricted in this way. Hermite interpolation for example requires only the data locally at the nearest neighboring grid points. Although the topic is not pursued in this paper these other schemes should perform equally well in bounded regions or for inhomogeneous turbulence.

APPENDIX: Basis Functions

Lagrangian Interpolation

Following the notation used in Section 2, Lagrangian basis function can be written as

$$L_i(x)=0; \qquad x<x_{i-3}$$

$$=\frac{1}{120}\,[4\xi-5\xi^3+\xi^5], \qquad \xi=\frac{(x-x_{i-3})}{h}; \qquad x_{i-3}\leqslant x\leqslant x_{i-2}$$

$$=\frac{1}{24}\,[-6\xi-\xi^2+7\xi^3+\xi^4-\xi^5], \qquad \xi=\frac{(x-x_{i-2})}{h}; \qquad x_{i-2}\leqslant x\leqslant x_{i-1}$$

$$
\begin{aligned}
&= \frac{1}{12}\,[12\xi + 8\xi^2 - 7\xi^3 - 2\xi^4 + 2\xi^5], && \xi = \frac{(x - x_{i-1})}{h}; && x_{i-1} \leqslant x \leqslant x_i \\
&= \frac{1}{12}\,[12 - 4\xi - 5\xi^2 + 5\xi^3 + 3\xi^4 - \xi^5], && \xi = \frac{(x - x_i)}{h}; && x_i \leqslant x \leqslant x_{i+1} \\
&= \frac{1}{24}\,[-6\xi + 16\xi^2 - \xi^3 - 4\xi^4 + \xi^5], && \xi = \frac{(x - x_{i+1})}{h}; && x_{i+1} \leqslant x \leqslant x_{i+2} \\
&= \frac{1}{120}\,[6\xi - 5\xi^2 - 5\xi^3 + 5\xi^4 - \xi^5], && \xi = \frac{(x - x_{i+2})}{h}; && x_{i+2} \leqslant x \leqslant x_{i+3} \\
&= 0; && && x_{i+3} \leqslant x, \qquad \text{(A.1)}
\end{aligned}
$$

where h is the grid spacing and because of periodicity

$$
\begin{aligned}
x_p &= x_{p+N} && \text{if } p < 0 \\
&= x_{p-N} && \text{if } p > N.
\end{aligned}
\qquad \text{(A.2)}
$$

Hermite Interpolation

The basis functions for Hermite interpolation can be written as

$$
\begin{aligned}
H_i(x) &= 0; && && x < x_{i-1} \\
&= \xi^2(3 - 2\xi), && \xi = \frac{(x - x_{i-1})}{h}; && x_{i-1} \leqslant x \leqslant x_i \\
&= (1 - \xi)^2\,(1 + 2\xi), && \xi = \frac{(x - x_i)}{h}; && x_i \leqslant x \leqslant x_{i+1} \\
&= 0; && && x_{i+1} < x
\end{aligned}
$$

and

$$
\begin{aligned}
G_i(x) &= 0; && && x < x_{i-1} \\
&= h\xi^2(\xi - 1), && \xi = \frac{(x - x_{i-1})}{h}; && x_{i-1} \leqslant x \leqslant x_i \\
&= h(1 - \xi)^2\,\xi, && \xi = \frac{(x - x_i)}{h}; && x_i \leqslant x \leqslant x_{i+1} \\
&= 0; && && x_{i+1} < x, \qquad \text{(A.3)}
\end{aligned}
$$

where h is the grid spacing and periodicity implies (A.2).

Linear Interpolation

The basis function for Linear interpolation is

$$\begin{aligned} P_i(x) &= 0; && x < x_{i-1} \\ &= (1-\xi), \quad \xi = \frac{(x-x_{i-1})}{h}; && x_{i-1} \leqslant x \leqslant x_i \\ &= \xi, \quad \xi = \left(\frac{x-x_i}{h}\right); && x_i \leqslant x \leqslant x_{i+1} \\ &= 0; && x_{i+1} < x, \end{aligned} \tag{A.4}$$

where h, as in the previous cases, is the grid spacing and periodicity implies Eq. (A.2).

ACKNOWLEDGMENTS

The authors gratefully acknowledge support by DARPA, under URI Contract N00014-86-K0754, and the National Science Foundation, under Grant ATM-8310136. The computations reported here were performed at the John von Neumann Center, Princeton.

REFERENCES

1. S. BALACHANDAR AND M. R. MAXEY, "On Aerosol Particle Coagulation in Homogeneous Turbulence," Sixth Symposium on Turbulent Shear Flows, Toulouse, France, September 7–9, 1987.
2. D. B. HAIDVOGEL, in *Symposium on Spectral Methods for Partial Differential Equations, NASA. Langley*, 1982, edited by R. G. Voigt *et al.* (Siam, Philadelphia, 1984), p. 97.
3. R. M. KERR, *J. Fluid Mech.* **153**, 31 (1985).
4. R. H. KRAICHNAN, *Phys. Fluids* **13**, 22 (1970).
5. S. A. ORSZAG, *Stud. Appl. Math.* **50**, 293 (1972).
6. Y. H. PAO, *Phys. Fluids* **8**, 1063 (1965).
7. G. S. PATTERSON AND S. A. ORSZAG, *Phys. Fluids* **14**, 2538 (1971).
8. J. R. RICE, *Numerical Methods, Software, and Analysis* (McGraw–Hill, New York, 1983), p. 79.
9. J. J. RILEY AND G. S. PATTERSON, *Phys. Fluids* **17**, 292 (1974).
10. G. I. TAYLOR, *Proc. London Math. Soc.* **20**, 196 (1921).
11. P. K. YEUNG AND S. B. POPE, Sibley School of Mechanical and Aerospace Engineering, Technical Report No. FDA-87-6, Cornell University, Ithaca, NY, 1987 (unpublished).
12. P. K. YEUNG AND S. B. POPE, *J. Comput. Phys.* **79**, 373 (1988).
13. O. C. ZIENKIEWICZ, *The Finite Element Method* (McGraw–Hill, New York, 1979), p. 737.