Computers & Fluids 38 (2009) 1943–1949

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

A spurious evolution of turbulence originated from round-off error in pseudo-spectral simulation

Lian-Ping Wang *, Bogdan Rosa

Department of Mechanical Engineering, 126 Spencer Laboratory, University of Delaware, Newark, Delaware 19716-3140, USA

ARTICLE INFO

Article history:
Received 27 August 2008
Received in revised form 20 April 2009
Accepted 1 June 2009
Available online 9 June 2009

ABSTRACT

We report a slowly-developing, spurious numerical solution in pseudo-spectral direct numerical simulation (DNS) of incompressible fluid turbulence. When the effect of machine round-off on the divergence-free condition is not carefully controlled, a problem can develop slowly (over about 50 large-eddy turnover times) and eventually leads to an unphysical flow field. The problem was found with a previously published, highly-compact algorithm for pseudo-spectral DNS and therefore it is important to document the contamination of this numerical artifact on simulated turbulence structure and statistics. This is a striking example since the problem is not easily noticeable due to its very long development time, and it does not lead to numerical instability but rather a different flow state. A theory is developed to explain the unphysical evolution and predicts the exponential growth of round-off error induced velocity divergence. The theory shows that any correlation of the large-scale forcing with the velocity field at the beginning of the time step could lead to amplification of the velocity divergence. For this reason, the problem is quite reproducible. Several simple remedies are tested and shown to correct the problem. It is shown that all revised algorithms are identical *theoretically* to the original algorithm, with the only difference in the level of control for the divergence-free condition of the simulated flow field. A general recommendation is that the pressure projection operation should be performed at the end of each time step to ensure that the divergence-free condition is not contaminated by machine round-off.

1. Introduction

Pseudo-spectral direct numerical simulation (DNS) of incompressible fluid turbulence has played a central role in our understanding of turbulence dynamics in the inertial and dissipation subranges since its introduction in the early 1970s [1–3]. The exponential accuracy in space of the pseudo-spectral method makes it a method of choice for small-scale turbulence simulation. The pseudo-spectral DNS has also contributed enormously to our understanding of the complex interactions between small suspended particles and flow vortical structures such as preferential concentration and enhanced particle collision rate by turbulence [4–6]. The algorithms for pseudo-spectral DNS are generally thought to be well established.

In this short communication, we wish to demonstrate a spurious numerical solution in pseudo-spectral DNS of incompressible fluid turbulence. The problem was realized with a previously published, highly-compact algorithm [7] for pseudo-spectral DNS and is shown to originate from machine round-off errors. The unique feature here lies in the very long development time of the numerical problem so that it has the potential to contaminate the physical flow statistics without notice. Under most circumstances, round-off errors can affect the instantaneous turbulent flow field due to the nonlinear nature of the Navier-Stokes equation; it is generally believed, however, that they do not affect the statistical features of flow vortical structures or averaged statistics of the simulated flow as long as the flow is well resolved in space and the Courant-Friedrichs-Lewy (CFL) condition is observed [8]. Theoretically, the algorithm to be discussed is precise and accurate. We discovered the problem in a very unusual manner as the problem did not lead to numerical instability. Instead it produces a numerically realizable but unphysical flow field for incompressible fluid. It could be possible that such evolution has some dynamic relevance to weakly compressible turbulence.

The problem to be discussed falls under the general context of spurious numerics in computational fluid dynamics [9,10]. Spurious numerics could come in many forms. They may lead to (1) numerical instability, (2) unphysical flow evolution without numerical instability, or (3) negligible consequence if the magnitude of spurious numerics remains small in magnitude. Case (1) is easily realizable and Case (3) can be tolerated. Case (2) is the most difficult case to deal with, which occurs in our pseudo-spectral DNS due to the amplification of round-off error in the divergence of velocity field after long-time integration. Yee et al. [9] have shown that, due to spurious numerics, a discretized representation of partial differential equation (PDE) can produce a stable solution that is completely different from the physical solution of

* Corresponding author. Tel.: +1 13028318160; fax: +1 3028313619.
E-mail addresses: lwang@udel.edu (L.-P. Wang), bogdan.rosa@imgw.pl (B. Rosa).

0045-7930/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2009.06.001

the original conrinuum PDE. The spurious numerics may originate from errors introduced through initial conditions, boundary conditions, temporal and spatial discretizations, finite time step size, and finite grid spacings. Standard CFL guidelines for stability do not eliminate spurious numerics contained in long time integrations.

In a typical pseudo-spectral DNS, the following incompressible, time-dependent, and three-dimensional Navier Stokes equation governing the fluid velocity $\mathbf{U}$ is solved

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{U} \times \vec{\omega} - \nabla\left(\frac{P}{\rho} + \frac{1}{2}\mathbf{U}^2\right) + \nu\nabla^2\mathbf{U} + \mathbf{F}(\mathbf{x}, t), \tag{1}$$

along with the continuity equation $\nabla \cdot \mathbf{U} = 0$. Here $\vec{\omega} \equiv \nabla \times \mathbf{U}$ is the flow vorticity, P is the pressure. The fluid density ρ is assumed to be a constant. The $\mathbf{F}$ term represents a large-scale, prescribed forcing field. A cubic domain of nominal size of $L_B = 2\pi$ is considered and periodic boundary conditions are assumed in all three spatial directions.

In the Fourier space, the Navier-Stokes equation is

$$\frac{\partial \mathbf{u}}{\partial t} = \mathcal{N}_1 - \imath\mathbf{k}\mathcal{N}_2 - \nu k^2\mathbf{u} + \mathbf{f}, \tag{2}$$

where $\mathbf{u}$, $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathbf{f}$ are functions of the wave vector k and time t; they represent the Fourier transform of $\mathbf{U}$, $\mathbf{U} \times \vec{\omega}$, $\frac{P}{\rho} + \frac{1}{2}\mathbf{U}^2$, and $\mathbf{F}$, respectively. Note that $\mathcal{N}_1$ is a vector field and $\mathcal{N}_2$ is a scalar. The divergence-free condition $\imath\mathbf{k} \cdot \mathbf{u} = 0$ implies that

$$\mathcal{N}_i \equiv \mathcal{N}_{1,i} - \imath k_i \mathcal{N}_2 = \mathcal{N}_{1,j} P_{ij}, \quad \text{with } P_{ij} \equiv \delta_{ij} - \frac{k_i k_j}{k^2}, \tag{3}$$

where P_{ij} is known as the pressure projection operator. Therefore, $\mathcal{N}_2$ does not need to be explicitly calculated.

The time-integration algorithm proposed in [7] makes use of the Adams-Bashforth scheme for the nonlinear term $\mathcal{N}$ and exact integration for the linear viscous term. There are five basic steps in this algorithm (will be referred to as Algorithm A):

1. Compute the nonlinear term $\mathcal{N}_1$ at the current time t;
2. Compute the full nonlinear term at t by the projection as $\mathcal{N}_i = \mathcal{N}_{1,j} P_{ij}$;
3. Perform the time integration

$$\mathbf{A}(t + \delta t) = \mathbf{u}(t) + \delta t[1.5\mathcal{N}(t) - 0.5\mathcal{N}(t - \delta t)e^{-\nu k^2 \delta t}]; \tag{4}$$

4. Compute the velocity field at $t + \delta t$ by $\mathbf{u}(t + \delta t) = \mathbf{A}(t + \delta t)\exp(-\nu k^2 \delta t)$, and advance time;
5. Replace $\mathcal{N}(t - \delta t)$ with $\mathcal{N}(t)$ calculated in Step 2, in order to prepare for the Adams–Bashforth term in the next time step.

Here δt is the time step size. The transformation shown in Step 4 was used to analytically eliminate the viscous term in Eq. (4), or equivalently the exponential factor $\exp(-\nu k^2 \delta t)$ in Steps 3 and 4 results from the exact integration of the viscous term. The forcing term is left out in the above but can be added as needed, depending on the nature of the forcing scheme.

In theory, the above time integration scheme satisfies the divergence-free condition $\mathcal{D} \equiv \nabla \cdot \mathbf{U} = 0$ or $\imath\mathbf{k} \cdot \mathbf{u}(t + \delta t) = 0$ at the end of the time step if the velocity field $\mathbf{u}(t)$ at the beginning of the time step is divergence-free. In our simulation, the initial flow is an incompressible random flow field with a $k^{-5/3}$ energy spectrum. An MPI code using domain decomposition [11] was developed previously using the above algorithm. The code have always generated acceptable results.

Recently, we use the same turbulence simulation code to study dynamics and collision rate of heavy particles. Due to relatively low particle concentration, we needed to run the simulation for a very long time (say, 50 to 100 large eddy turnover times). When we visualized the particle concentration field, an unexpected large-scale clustering was observed when the relevant parameters of the particles were set to yield clustering at the Kolmogorov scale. This is when we realized that the simulated flow somehow gradually evolved into a flow with completely different vortical structures, although the simulation was supposed to produce a statistically stationary turbulent flow under a large-scale deterministic forcing with a fixed kinetic energy of 0.555440 and 0.159843 for the first two wavenumber shells $0.5 < |\mathbf{k}| < 1.5$ and $1.5 < |\mathbf{k}| < 2.5$, respectively. Fig. 1 shows the isosurfaces of vorticity magnitude at 1.7 times the field mean, at four times $t = 27.8T_e$, $52.8T_e$, $55.6T_e$, $66.7T_e$, respectively, where $T_e \equiv (u')^2/\epsilon = 3.60$ is the large-eddy turnover time, where u' and ϵ are rms fluctuation velocity and the average energy dissipation rate per unit mass (see Table 1). The flow structures appear to be normal for a long time before such an unphysical stage is developed. The flow at $t = 66.7.0T_e$ is dominated by two very strong tornado-like vortices whose induced flow makes other weaker vortex tubes to wrap around them!

It took us a long time to trace the origin of such an unusual flow evolution. In fact, the problem would not have been noticed if the simulation had not heen performed for a very long time. The analysis below demonstrates succinctly that the origin is a result of tiny machine round-off errors combined with a positive feedback mechanism. Several remedies are then offered and are shown to correct the problem.

2. Analysis

A large number of flow statistics were processed in order to trace the origin of the problem. These include rms fluctuating velocity u', the longitudinal integral length scale L_f, the energy dissipation rate ϵ, the Taylor-microscale flow Reynolds number R_λ, spatial resolution parameter $k_{\max}\eta$, the CFL number $|U|_{\max}\delta t/\delta x$, Kolmogorov scales (length η and time τ_k), and the skewness ($\mathcal{S}$) and flatness ($\mathcal{F}$) of velocity gradient, where $k_{\max} = 61.5$ is the maximum wavenumber resolved in the simulation. The skewness and flatness are based on an average over the three longitudinal gradients $\partial U_1/\partial x_1$, $\partial U_2/\partial x_2$, and $\partial U_3/\partial x_3$. We made sure that the flow was well resolved spatially and a small time step was used by requiring CFL < 0.3 and $k_{\max}\eta > 1.4$ at any time. Table 1 summarizes the average values from a corrected, well-behaved numerical algorithm (see below).

It was observed that the skewness $\mathcal{S}$ and flatness $\mathcal{F}$ were well behaved for $t < 50T_e$ (Fig. 2), but then they mysteriously deviated from the expected range.

The origin of the problem became immediately clear when the rms fluctuation $\mathcal{D}'$ of the velocity divergence $\mathcal{D}$ was processed (Fig. 3). According to Eq. (4) and noting that $\mathcal{N}$ is always divergence-free, one can conclude that $\mathcal{D}'$ would be identically zero at all times.

This is not the case as shown in Fig. 3. While $\mathcal{D}'$ is on the order of 10^{-15} initially, it grows monotonically with time exponentially for the time interval of $10T_e < t < 50T_e$. While $\mathcal{D}'$ is still very small after many eddy turnover times, it eventually grows to a level that renders the flow field *locally compressible* (note that the mean of $\mathcal{D}$ is always zero). So there are regions of positive and negative D in the flow field. While nothing strange is noticeable for $t < 50T_e$ when other statistics are plotted as a function of time, the exponential growth of $\mathcal{D}'$ is established long before $t = 50T_e$. This indicates that there is a positive dynamic feedback mechanism that drives the system away from $\mathcal{D} = 0$ once a small error is introduced to the $\mathcal{D}$ field. The value of $\mathcal{D}'$ reached a maximum at around $t = 53.6T_e$ and then decreases slightly with time thereafter. The maximum $\mathcal{D}'$ is around $0.118/\tau_k$ or $3.041/T_e$, therefore, it is a relative weak compressibility when scaled by the Kolmogorov time but strong relative to the large scale at the moment of saturation.

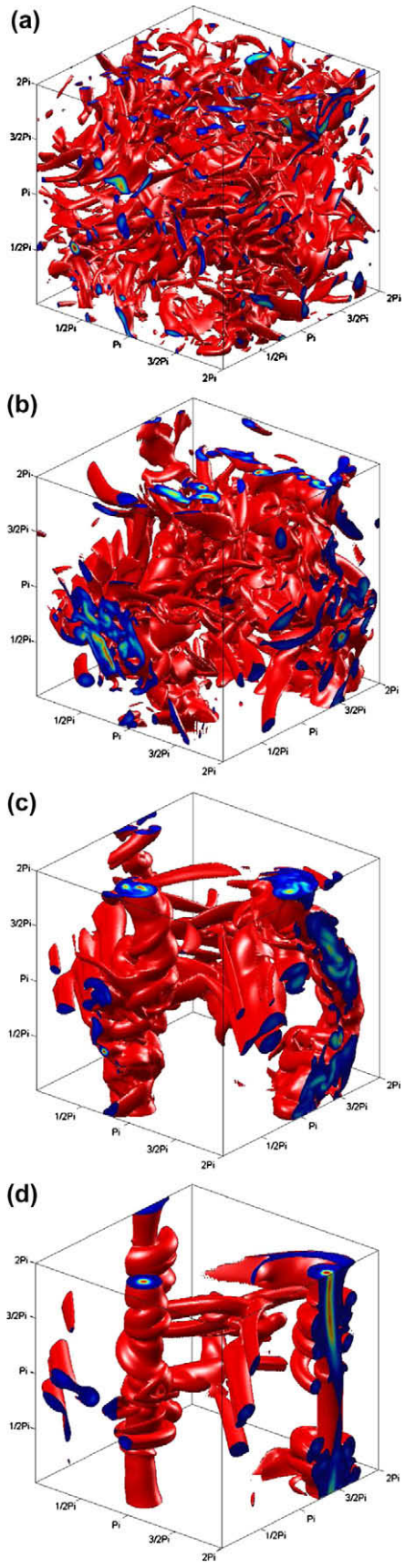

Fig. 1. Isosurfaces of vorticity magnitude at 1.7 times the local-in-time field mean at $t = 27.8T_e$, $52.8T_e$, $55.6T_e$, and $66.7T_e$ from top to bottom, respectively.

What originates the deviation from the divergence-free condition? We first note that, since $\mathcal{D}(\mathbf{x}, t = 0) = 0$, the only possibility for the deviation is the machine round-off in the code. Namely, summing three divergence-free components in Eq. (4) could introduce a small round-off error that leads to a very tiny non-zero $\mathcal{D}$ to the $\mathbf{u}(t + \delta t)$ field. Second, if $\mathbf{u}(t)$ has a non-zero $\mathcal{D}$, it will be passed on to $\mathbf{u}(t + \delta t)$.

With the above observations, we shall now attempt to develop a theory to predict the exponential growth of $\mathcal{D}'$ in the intermediate time interval $5T_e < t < 55T_e$. Let d be the Fourier transform of $\mathcal{D}$, then $d = \imath\mathbf{k} \cdot \mathbf{u}$. By multiplying Eq. (2) with $\imath\mathbf{k}\cdot$, we have

$$\frac{\partial d}{\partial t} = \imath\mathbf{k} \cdot \mathcal{N} - \nu k^2 d + \imath\mathbf{k} \cdot \mathbf{f}. \tag{5}$$

Since the algorithm computes $\mathcal{N}$ by Eq. (3), it follows that

$$\imath\mathbf{k} \cdot \mathcal{N} = 0. \tag{6}$$

Therefore, Eq. (5) becomes

$$\frac{\partial d}{\partial t} = -\nu k^2 d + \imath\mathbf{k} \cdot \mathbf{f}. \tag{7}$$

For unforced modes, we have

$$\frac{\partial d}{\partial t} = -\nu k^2 d, \quad \text{for} \quad \mathbf{k} > 2.5. \tag{8}$$

This implies that the divergence field would not have small-scale structures since the viscous term quickly removes the round-off errors. For the forced modes (i.e., $\mathbf{k} < 2.5$), we need to examine the term $\imath\mathbf{k} \cdot \mathbf{f}$. In the particular deterministic forcing scheme used in [7], the value of $\mathbf{u}(t + \delta t)$ in each of the two forced wavenumber shells (i.e., $0.5 < k < 1.5$ and $1.5 < k < 2.5$) was multiplied, at the end of each time step, by a factor α such that the kinetic energy in the first two shells is restored to the prescribed value of 0.555440 and 0.159843, respectively. The value α could be different for the two shells and could vary with time. Now we shall make an approximation by assuming that all forced modes receive a same scaling factor α. We proceed to estimate α. Since all dissipated energy is made up by the forcing in the range $\mathbf{k} < 2.5$, a reasonable estimate for α is then

$$\alpha = \sqrt{\frac{0.555440 + 0.159843}{0.555440 + 0.159843 - \epsilon\delta t}} \approx 1 + 0.14386\delta t, \tag{9}$$

where $\epsilon = 0.2058$ is the average dissipation rate (See Table 1). At this point, according to Eq. (2), we can approximate the forcing term as

$$\delta t\mathbf{f} \approx (\alpha - 1)\{\mathbf{u}(t) + \delta t[\mathcal{N} - \nu k^2 \mathbf{u}(t)]\}. \tag{10}$$

It follows that

$$\imath\mathbf{k} \cdot \mathbf{f}\delta t = (\alpha - 1)(1 - \nu k^2 \delta t)d \approx (\alpha - 1)d \approx 0.14386 d\delta t. \tag{11}$$

Since $\nu k^2 \delta t << 1$ for $\mathbf{k} < 2.5$, then the divergence for the forced modes grows exponentially as

$$d \propto e^{0.14386t} \quad \text{for} \quad \mathbf{k} < 2.5. \tag{12}$$

Therefore, a prediction for the slope of $\log_{10}(\mathcal{D}'\tau_k)$ versus t/T_e is

$$0.14386 \times \log_{10} e \times T_e = 0.14386 \times 0.4343 \times 3.60 = 0.2249. \tag{13}$$

This theoretical slope is plotted in Fig. 3 (a) and compares well with the simulation result from Algorithm A. The positive feedback mechanism is now clear: it is the specific deterministic scheme that amplifies the velocity divergence on large-scale forced modes, as shown by Eqs. (10) and (11). In other words, the forcing field is correlated with the $\mathbf{u}(t)$ field. If a stochastic forcing scheme is used,

Table 1
Parameter setting and average flow statistics obtained from the time interval $13.9T_e < t < 66.7T_e$ using Algorithm B.

N	ν	δt	u'	L_f	ϵ	R_λ	$k_{\max}\eta$	CFL	η	τ_k	$\mathcal{S}$	$\mathcal{F}$
128	0.004	0.002	0.8608	1.519	0.2058	100.2	1.478	0.2149	0.02365	0.1399	−0.4715	5.084

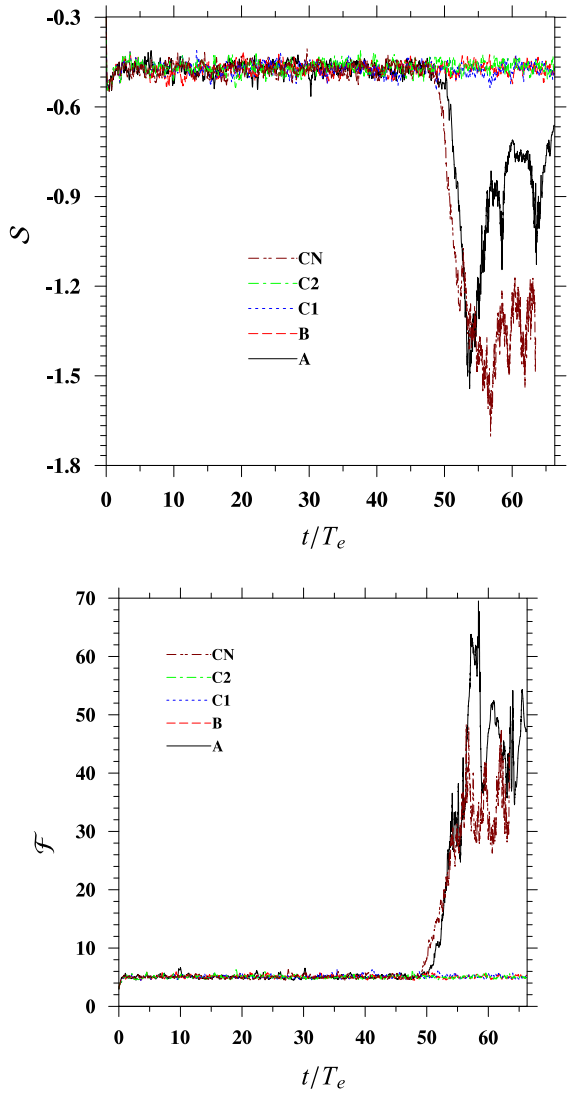

Fig. 2. The skewness and flatness as a function of time. The legends indicate the algorithms: A for the original algorithm with the projection at the beginning of a time step, B for the revised algorithm with the projection done at the end of a time step, C1 applies a second projection to $\mathbf{u}(t)$ once every 1000 time steps, C2 applies a second projection to $\mathbf{u}(t+\delta t)$ once every 1000 time steps, and CN for using the Crank-Nicholson scheme for the viscous term. Algorithm A is described in Section 1, Algorithm CN in Section 2, and Algorithms B, C1, and C2 in Section 3.

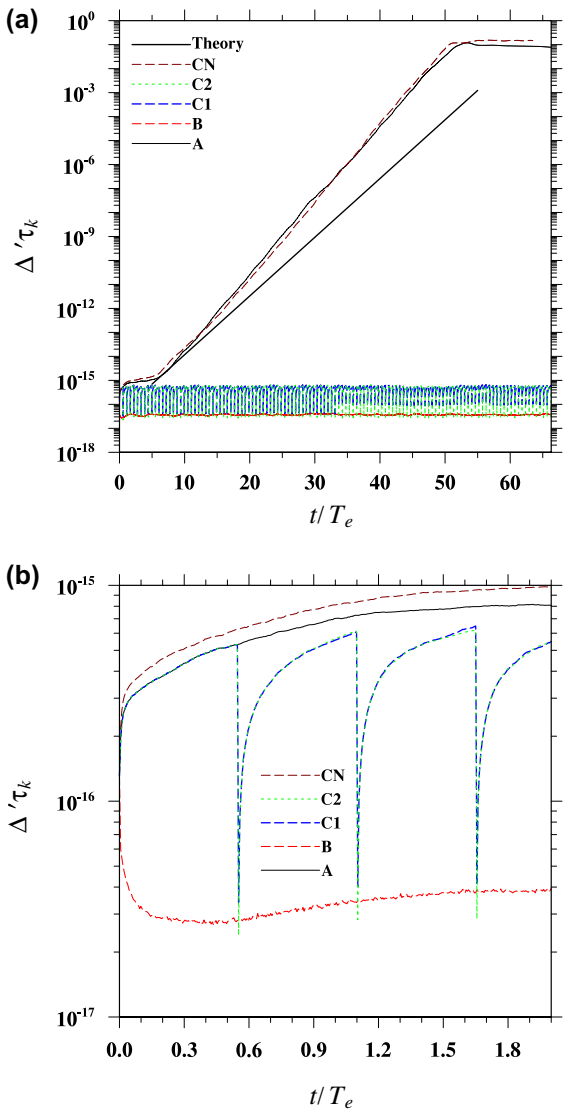

Fig. 3. The rms fluctuation of the velocity divergence as a function of time. The legends indicate the type of algorithm: (a) the whole time interval; (b) zoom in near $t = 0$. In Algorithms C1 and C2, an additional projection was done every 1000 time steps or every $\Delta t = 1000\delta t = 2.0 = 0.56T_e$ time interval, so there is a periodic restoration of $\mathcal{D}'$ back to the machine representation of zero divergence. Algorithm B always gives the machine zero-divergence.

such a correlation may be weaker or non-exist; the exponential amplification would also be either weaker or non-exist.

Fig. 4 shows color contours of vorticity and velocity divergence in a same 2D slice for the flow shown in Fig. 1(d). Several interesting observations can be made. First, the divergence exhibits large-scale structure only, consistent with what are described by Eqs. (8) and (12); while the vorticity shows small-scale structure inside the large-scale bundles. Second the large-scale vorticity bundles tend to overlap with regions of negative divergence, although the correlation of bundles and the convergence zones is not strong. Therefore, large-scale convergence zones cause bundling of the vortices. At the same time, the bundling of the vortices can effectively induce large-scale flow field. This positive dynamic feedback can drive the flow from initially randomly distributed small-scale vortical structures to clustered large-scale stuctures, as seen in Fig. 1. The rapid growth of $\mathcal{S}$ and $\mathcal{F}$ at the late stage also signifies that a dynamic bias is introduced by this positive feedback mechanism. The positive feedback mechanism shown here and the mechanism for exponential growth of $\mathcal{D}'$ in Eq. (12) make the spurious flow evolution very reproducible.

To demonstrate this point, we consider the use of Crank-Nicholson scheme for the viscous term (Algorithm CN) as:

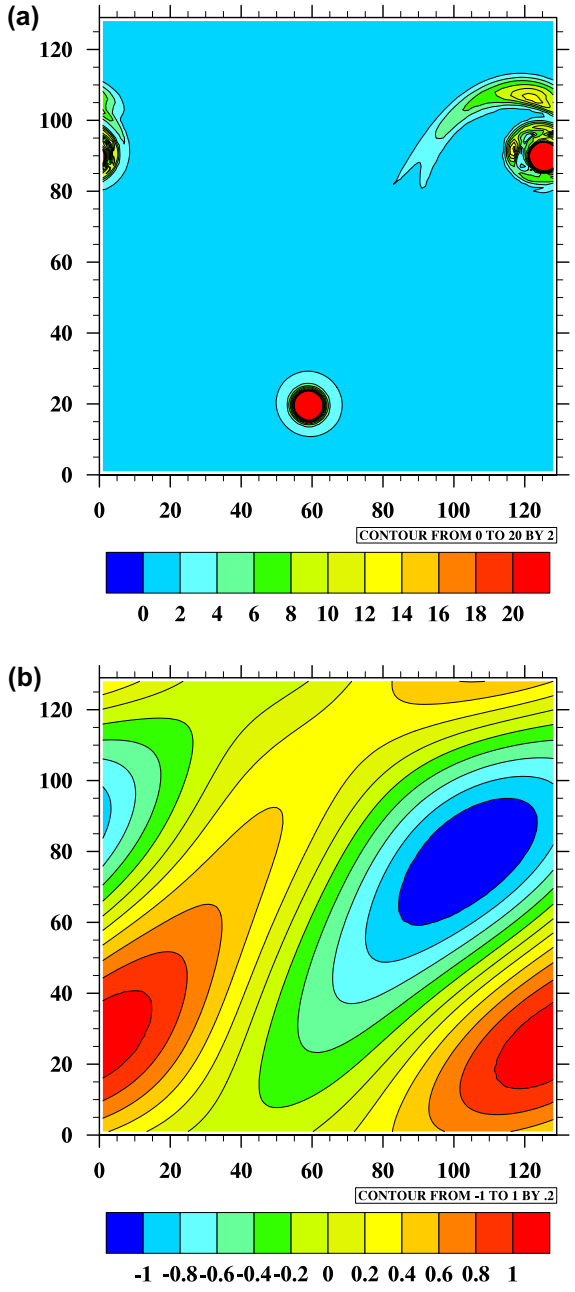

Fig. 4. Color contours of (a) vorticity magnitude and (b) velocity divergence in the same 2D cross-section for the flow shown in Fig. 1(d). The correlation coefficient between the vorticity magnitude and velocity divergence is −0.55.

1. Compute the nonlinear term $\mathcal{N}_1$ at the current time t;
2. Compute the full nonlinear term at t by the projection as $\mathcal{N}_i = \mathcal{N}_{1,j}P_{ij}$;
3. Perform the time integration as

$$\mathbf{u}(t+\delta t) = \mathbf{u}(t) + \delta t[1.5\mathcal{N}(t) - 0.5\mathcal{N}(t-\delta t)] - \delta t\nu k^2[0.5\mathbf{u}(t) + 0.5\mathbf{u}(t+\delta t)] \quad (14)$$

and advance time;
4. Replace $\mathcal{N}(t-\delta t)$ with $\mathcal{N}(t)$ calculated in Step 2, in order to prepare for the Adams-Bashforth term in the next time step.

The results from this CN algorithm are also plotted in Figs. 2 and 3.

The average exponential growth rates are similar, as implied by the theory, Eq. (12). The skewness and flatness deviated from the normal range somewhat earlier (Fig. 2).

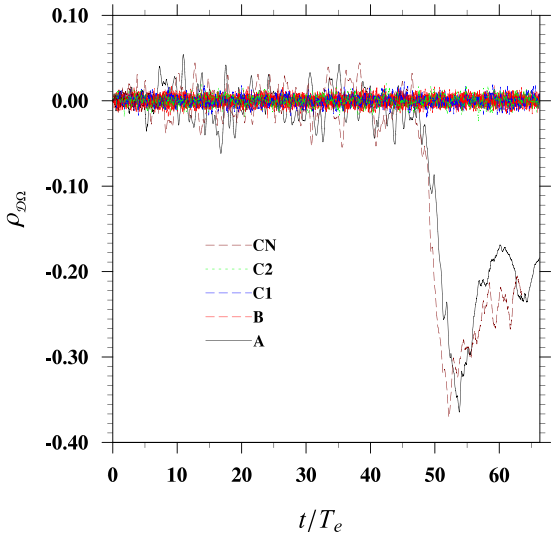

Fig. 5. Correlation coefficient between the divergence field and the vorticity magnitude field.

If double-precision representation were used in the code, the problem may be delayed since the relative magnitude of D' at the initial time due to the machine round-off will be smaller. The positive feedback mechanism, Eq. (12), implies nevertheless that the same problem will eventually occur.

Fig. 3 shows that the growth of $\mathcal{D}'$ is very slow initially, but it is monotonic. After about $t = 8T_e$, an exponential growth is quickly established. But even at this stage, the flow dynamics is not yet affected. Only when $\mathcal{D}'$ reaches a level of $\mathcal{O}(1/T_e)$, the skewness $\mathcal{S}$ and flatness $\mathcal{F}$ become unphysical. The true danger here is that one could use the unphysical flow without knowledge, due to the facts that the development time is very long (about $50T_e$) and most of the flow statistics appear to be acceptable for a long time.

Fig. 5 shows the correlation coefficient between the $\mathcal{D}$ field and the vorticity magnitude field $\Omega \equiv |\vec{\omega}|$, namely, $\rho_{\mathcal{D}\Omega} \equiv \langle (D - \langle D\rangle)(\Omega - \langle\Omega\rangle)\rangle/(\mathcal{D}'\Omega')$, where Ω' is rms fluctuation of Ω. A significant negative correlation is observed when the flow deviates from its normal evolution. This indicates that the high-vorticity region tends to be found in convergence zones, consistent with the visualizations in Fig. 4. Another observation is that the fluctuation level of $\rho_{\mathcal{D}\Omega}$ is higher, even at the earlier times, for the two algorithms (A and CN) contaminated by the round-off.

3. Remedies

Once the origin of the problem is found, we can offer several remedies that will correct the problem. The most logical modification is to switch the projection operation to the end of the time step by the following modified algorithm (Algorithm B):

1. Compute the nonlinear term $\mathcal{N}_1$ at the current time t;
2. Perform the following time integration

$$\widetilde{\mathbf{A}}(t+\delta t) = \mathbf{u}(t) + \delta t\left[1.5\mathcal{N}_1(t) - 0.5\mathcal{N}_1(t-\delta t)e^{-\nu k^2\delta t}\right]; \quad (15)$$

3. Compute $\widetilde{\mathbf{u}}(t+\delta t) = \widetilde{\mathbf{A}}(t+\delta t)\exp(-\nu k^2\delta t)$;
4. Project $\widetilde{\mathbf{u}}$ to yield $u_i(t+\delta t) = \widetilde{u}_j P_{ij}$ and advance time;
5. Replace $\mathcal{N}_1(t-\delta t)$ with $\mathcal{N}_1(t)$ calculated in Step 1, in order to prepare for the Adams–Bashforth term in the next time step.

Delaying the projection operation and applying it directly to the velocity field ensure that the $\mathbf{u}(t+\delta t)$ field is divergence-free or any round-off error on the $\mathcal{D}$ field at the beginning of the time step

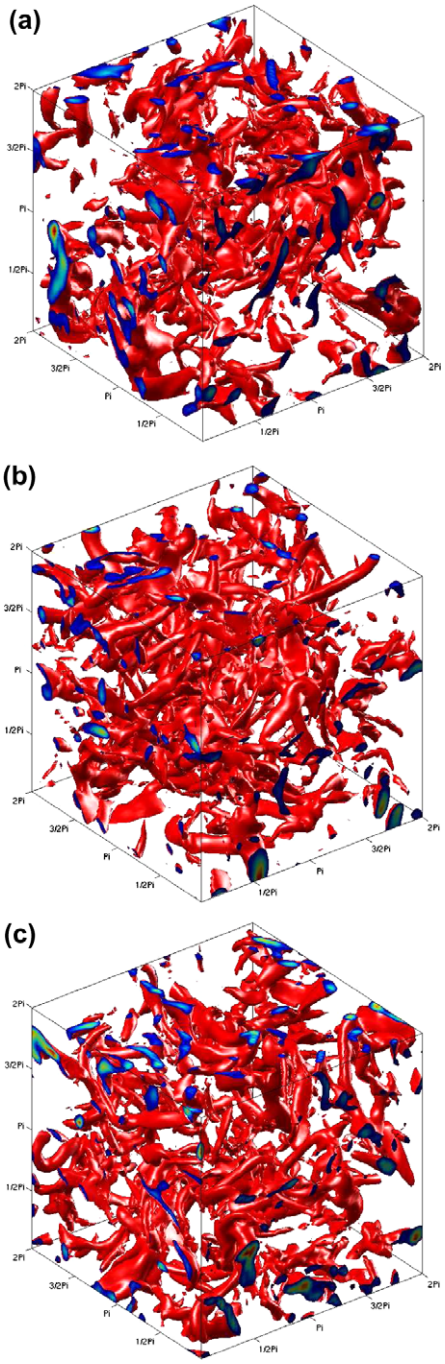

Fig. 6. Isosurfaces of vorticity magnitude at 1.7 times the local-in-time field mean at the end of the simulation ($t = 66.7T_e$) using Algorithm (a) B, (b) C1, and (c) C2.

will not be carried on to the $\mathbf{u}(t+\delta t)$ field. The main change in Algorithm B is that the first part of the nonlinear term $\mathcal{N}_1$ is saved for the purpose of the Adams-Bashforth step, instead of the full nonlinear term $\mathcal{N}$ in Algorithm A. It is straightforward to show that the net time evolution in Algorithm B is identical to that in Algorithm A, in view of the relationship given by Eq. (3). The computational requirement for Algorithm B is identical to that of Algorithm A.

Another alternative is to *periodically* flush out any contamination to the $\mathcal{D} = 0$ condition in Algorithm A by adding a second projection: either perform a second projection operation on $\mathbf{u}(t)$ (Algorithm C1) or perform a second projection operation on $\mathbf{u}(t+\delta t)$ (Algorithm C2). This flushing or second-projection step adds some additional computation, however, it only needs to be performed infrequently, say once every 1000 time steps, in view of the long development time for the contamination by round-off to become physically relevant.

Since $P_{ij}P_{jm} = P_{im}$ and thus $\mathcal{N}_j P_{ij} = \mathcal{N}_i$, we obtain

$$\begin{aligned} A_j(t+\delta t)P_{ij} &= u_j(t)P_{ij} + \delta t\left[1.5\mathcal{N}_i(t) - 0.5\mathcal{N}_i(t-\delta t)e^{-\nu k^2 dt}\right] \\ &= P_{ij}\left\{u_j(t) + \delta t\left[1.5\mathcal{N}_{1j}(t) - 0.5\mathcal{N}_{1j}(t-\delta t)e^{-\nu k^2 dt}\right]\right\} \\ &= P_{ij}\widetilde{A}_j(t+\delta t). \end{aligned} \tag{16}$$

It follows

$$\begin{aligned} u_i^{(C2)}(t+\delta t) &= u_j^{(A)}(t+\delta t)P_{ij} \\ &= A_j(t+\delta t)P_{ij}e^{-\nu k^2 \delta t} \\ &= \tilde{u}_j(t+\delta t)P_{ij} \\ &= u_i^{(B)}(t+\delta t), \end{aligned} \tag{17}$$

where the superscript above $\mathbf{u}(t+\delta t)$ corresponds to the Algorithm used. Therefore the net evolution for Algorithm C2 is identical to that of Algorithm B. Also Algorithm C1 is essentially the same as Algorithm C2. Algorithm B is clearly the optimal one since it only requires one projection operation per time step as in Algorithm B, and at the same time ensures zero-divergence at the end of every time step.

In Figs. 2, 3, and 5, we also show the results from Algorithms B, C1, and C2. All simulations were started with an identical initial flow field and were driven by a same large-scale forcing. Clearly, the problem in Algorithm A has been corrected by all of them. The additional projection in Algorithms C1 and C2 were done once every 1000 time steps, the desired effect of this additional projection is shown clearly in the zoom-in plot, Fig. 3(b). Also the results of Algorithms C1 and C2 are indistinguishable in Fig. 3(b). Finally, we visualize the vortical structures at the end of the simulation at $t = 66.7T_e$, in Fig. 6, to confirm that the overall flow structures are normal.

4. Conclusions and summary

In this paper, we have documented a spurious numerical solution in pseudo-spectral DNS of incompressible turbulence in a periodic cube, using a previously published, popular algorithm. This falls under the general context of spurious numerics, and in some cases, such spurious numerics can lead to a stable but unphysical flow evolution [9,10]. The artifact is shown to originate from machine round-off errors, but is driven by a dynamic positive feedback mechanism in the large-scale forcing term. The artifact does not affect the dynamics of the turbulent flow until about 50 large-eddy turnover times, making it rather hard to be detected. The artifact does not lead to numerical instability, but rather creates completely different flow structures and turbulence statistics. At the same time, it is reproducible due to the positive dynamic feedback. The striking tornado-like flow structures developed from the numerical artifact was found by surprise.

The problem was examined theoretically by the velocity-divergence equation. It was shown that unforced modes are able to remove the round-off errors by the viscous effect. However, the deterministic forcing provides a positive feedback on the large-scale forced modes. The exponential rate of growth at intermediate times is reasonably predicted.

Several remedies that do not require much additional computation have been offered and are shown to solve the problem. The general recommendation is that the pressure projection operation should be performed at the end of each time step to ensure that the

divergence-free condition is not contaminated by machine round-off. Fortunately, this is typically the case in non-spectral flow simulation codes.

Another general recommendation is that the divergence field be monitored, even for an algorithm that is designed to ensure zero-divergence. In the algorithms shown in this paper, the incompressibility is used to construct each of the algorithms, but the machine round-off combined with a dynamic feedback slowly moves the system away from the designed state of the turbulent flow.

Acknowledgements

This work was supported by the National Science Foundation (NSF) under grants ATM-0114100, ATM-0527140 and ATM-0730766; and by the National Center for Atmospheric Research (NCAR). The supercomputing support from NCAR's Computational and Information Systems Laboratory (Project numbers 35751010 and 35751012) is gratefully acknowledged. LPW also acknowledges support by National Natural Science Foundation of China (Project No. 10628206). Comments on the manuscript by Dr. Wojciech W. Grabowski, Mr. Hossein Parishani and Mr. Hui Gao are acknowledged. The National Center for Atmospheric Research is sponsored by the National Science Foundation.

References

[1] Orszag SA, Patterson GS. Numerical simulation of three-dimensional homogeneous isotropic turbulence. Phys Rev Lett 1972;28(2):76–9.
[2] Wang L-P, Chen S, Brasseur JG, Wyngaard JC. Examination of hypotheses in the Kolmogorov refined turbulence theory through high-resolution simulations. Part 1. Velocity field. J Fluid Mech 1996;309:113–56.
[3] Moin P, Mahesh K. Direct numerical simulation: a tool in turbulence research. Annu Rev Fluid Mech 1998;30:539–78.
[4] Wang L-P, Maxey MR. Settling velocity and concentration distribution of heavy particles in homogeneous isotropic turbulence. J Fluid Mech 1993;256:27–68.
[5] Sundaram S, Collins LR. Collision statistics in an isotropic particle-laden turbulent suspension. Part 1. Direct numerical simulations. J Fluid Mech 1997;335:75–109.
[6] Wang L-P, Ayala O, Kasprzak SE, Grabowski WW. Theoretical formulation of collision rate and collision efficiency of hydrodynamically interacting cloud droplets in turbulent atmosphere. J Atmos Sci 2005;62:2433–50.
[7] Chen S, Shan X. High-resolution turbulence simulations using the connection machine-2. Comput Phys 1992;6(6):643–6.
[8] Eswaran E, Pope SB. An examination of forcing in direct numerical simulations of turbulence. Comput Fluids 1988;16:257–78.
[9] Yee HC, Sweby PK, Griffiths DF. Dynamic approach study of spurious steady-state numerical-solutions of nonlinear differential-equations. 1. The dynamics of time discretization and its implications for algorithm development in computational fluid-dynamics. J Comp Phys 1991;97:249–310.
[10] Yee HC, Torczynski JR, Morton SA, Visbal MR, Sweby PK. On spurious behavior of CFD simulations. Int J Numer Meth Fluids 1999;30:675–711.
[11] Dmitruk P, Wang L-P, Matthaeus WH, Zhang R, Seckel D. Scalable parallel FFT for spectral simulations on a Beowulf cluster. Parallel Comput 2001;27:1921–36.